CONTENT ASSESSMENT IN
INTELLIGENT COMPUTER-AIDED LANGUAGE LEARNING:
MEANING ERROR DIAGNOSIS FOR ENGLISH AS A SECOND LANGUAGE

DISSERTATION

Presented in Partial Fulfillment of the Requirements for

the Degree Doctor of Philosophy in the

Graduate School of The Ohio State University

By

Stacey M. Bailey, B.S., M.A., M.A.

*****

The Ohio State University
2008

Dissertation Committee: Approved by

Professor W. Detmar Meurers, Advisor

Professor Christopher H. Brew

Professor Donna R. Long Advisor

Graduate Program in Linguistics
Language practice that includes meaningful interaction is a critical component of many current language teaching theories. At the same time, existing research on intelligent computer-aided language learning (ICALL) systems has focused primarily on providing practice with grammatical forms. For most ICALL systems, content assessment of a learner response is limited to comparing it with a target response through string matching. But this severely restricts exercises that can be offered because expected responses must be tightly controlled. Yet the task-based activities that language instructors typically use in real language-learning settings require both form and content assessment in answer evaluation. Thus, there is a real need for ICALL systems that provide accurate content assessment beyond string matching.

This thesis addresses that need by taking an empirically driven approach to the exploration of content assessment. This work argues that while some meaningful activities are too unrestricted for ICALL systems to provide effective content assessment, where the line should be drawn on a spectrum of language exercises is an open question. At one extreme of the spectrum, there are tightly restricted exercises requiring minimal analysis in order to assess content. At the other extreme are unrestricted exercises requiring extensive form and content analysis to assess content. While full natural language understanding is beyond the scope of current technology, this thesis explores activities in the space between spectrum extremes.
The primary source of material for this exploration into ICALL content assessment is a corpus of language learner data collected for that purpose. The corpus is comprised exclusively of responses to short-answer reading comprehension questions by intermediate English language learners. Responses to these questions are ideal for developing and testing an approach to content error diagnosis because they exhibit linguistic variation on lexical, morphological, syntactic and semantic levels, but they have definable target responses that capture what it means to be correct.

The corpus is one of the first known to be annotated with diagnoses of meaning errors. Diagnoses were developed from analyzing the learner data and adopting an annotation scheme based on target modification. This corpus provided invaluable insight into the considerations necessary for developing an approach to diagnosing meaning errors.

Because variation is possible across learner responses in activities in the middle ground of the spectrum, any degree of content assessment must be flexible and support the comparison of target and learner responses on several levels including token, chunk and relation. This thesis presents an architecture for a content assessment module (CAM) which provides this flexibility using multiple surface-based matching strategies and existing language processing tools. This thesis shows that content assessment for middle ground language activities is feasible using shallow NLP strategies. Detection of meaning errors approaches 90% in the test set. It also shows that diagnosis of meaning errors is feasible using an approach that relies on machine learning, though additional testing with a larger corpus is needed. By developing and testing this model, as well as exploring the middle ground of activities, this work begins to bridge the gap between what is practical and feasible from a processing perspective and what is desirable from the perspective of current theories of language instruction.
To my mother, Dr. Sandra S. Bailey:

Mom, the best in me is merely a reflection of you.

And to the memory of my father, Dan Bailey:

Daddy, you’re missin’ all the good stuff.
ACKNOWLEDGMENTS

Just because there are no words to express my gratitude for those who helped me see this thesis through to the end doesn’t mean I shouldn’t say thanks. So, …

… Thank you to the members of my committee, Dr. Detmar Meurers, Dr. Chris Brew, and Dr. Donna Long.

First, to my advisor, Detmar. He helped to shape this fuzzy idea into a clear project. I thank him sincerely for his guidance through this process and for his insights into the interface between computational linguistics and ICALL. I also will forever be grateful for his encouragement when I needed it, not to mention his often clever use of metaphor.

And my thanks go to Chris, who gave me the one piece of advice I returned to again and again. And that was to keep it simple. So, from data collection to programming to writing, I tried to start from the simplest solution and work forward. I’d also like to thank Chris for asking good questions, even when I didn’t have the answers.

I would also like to express my gratitude for Donna’s advice and support. Her expertise on matters of second language teaching wasn’t merely useful. It was essential. She helped ground this work in the real world of language teaching. The importance of her willingness to help with comments or commitment to my progress cannot be overstated.
Thank you to the instructors and administrators of the American Language Program – Dr. Kathy Romstedt, Jeannette Bolivar, Laura Thomas, and Sonja Gassett. Their cooperation and participation made this work possible. And thank you to the students of those ALP classes. Without their words, this thesis would be wouldn’t be filled with mine.

Thank you to the ICALL and Clippers discussion groups. These groups were everything discussion groups should be – supportive, engaging and constructive. They provided a comfortable atmosphere for exchanging ideas, not to mention some practical advice.

Thank you to my former office mates and friends, Luiz Amaral and Ila Nagar. Luiz’s insights into Intelligent Computer-Aided Language Learning were much needed and his enthusiasm for my work much appreciated. And I give special thanks to Ila, for her limitless supply of moral support, her company, and her futon.

Thank you to my family. Stephanie, Joe and Alex deserve thanks for putting up with me. I’m sure it’s never easy in the best of times, and these were not always the best of times. I needed a little understanding, and (as always) they came through. I give thanks and eternal gratitude to my mother. I edited her dissertation, and now she has edited mine. I could say we’re even, but I know that’s just not true. I owe her more than I can ever repay.

And, finally, thank you to those who are as much family as they are friends. Thanks to Wilbert Munfus Jr. and Oni Guha for their friendship and loyalty in general and for their faith I could finish in particular. And my thanks go to Vanessa Metcalf. I’m not sure I could have finished without her, and not just because she lent me a desk. Her encouragement in the final few months meant more than I can say.

And thanks to everyone I inadvertently left out: No really. Thank you.
VITA

March 13, 1977 .......................... Born – Anniston, Alabama

1998 ................................. B.S., Journalism,
                             University of Florida

2001 ................................. M.A., Communications,
                             Culture and Technology,
                             Georgetown University

2006 ................................. M.A., Linguistics,
                             The Ohio State University

2003 - 2006 .......................... Teaching Assistant,
                             The Ohio State University

FIELDS OF STUDY

Major Field: Linguistics
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>ii</td>
</tr>
<tr>
<td>Dedication</td>
<td>iv</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>v</td>
</tr>
<tr>
<td>Vita</td>
<td>vii</td>
</tr>
<tr>
<td>List of Figures</td>
<td>xii</td>
</tr>
<tr>
<td>List of Tables</td>
<td>xiii</td>
</tr>
</tbody>
</table>

## 1 Introduction

1.1 Motivation ........................................... 2
   1.1.1 From an ICALL Perspective ......................... 2
   1.1.2 From a Language Teaching Perspective .......... 3
   1.1.3 From a Language Processing Perspective ....... 6
1.2 The Current Research ................................ 7
1.3 Outline .............................................. 9

## 2 Previous Research in Content Assessment

2.1 String Matching ....................................... 12
   2.1.1 Controlling Variation in Student Responses for Effective String Matching .......... 15
2.2 Pattern Matching ..................................... 16
2.3 Lexical Analysis and Matching ....................... 19
2.4 Syntactic Representation Analysis and Matching .... 24
   2.4.1 A Note on Indirect Content Analysis .......... 27
   2.4.2 Syntactic Representation Matching .......... 28
   2.4.3 Syntactic Analysis and Unexpected Input .... 29
2.5 Semantic Representation Analysis and Matching .... 30
2.6 Content Analysis and Assessment in Automatic Grading Systems .................. 38
   2.6.1 Short-Answer Automatic Grading ............... 42
3 Language Learning Exercises ............................................. 50
  3.1 Language-learning Exercises ......................................... 51
    3.1.1 The Inadequacy of Categorizing by Common Exercise Types 53
    3.1.2 A Sample of Language Learning Exercises .................... 57
    3.1.3 Characterization of Exercise Properties ...................... 64
      3.1.3.1 Explicit Knowledge in the Activity Model .............. 64
      3.1.3.2 The Nature and Availability of Target Responses .... 67
      3.1.3.3 External Criteria for Activity Evaluation ............. 71
  3.2 The Spectrum of Language-learning Exercises ..................... 74
    3.2.1 Placing Exercises on the Spectrum ............................ 78
    3.2.2 Focusing on Exercises in the Middle Ground ................ 82
  3.3 A Case Study: Reading Comprehension Questions ................... 85
  3.4 Summary and Conclusion ............................................. 88

4 Influences ................................................................. 92
  4.1 Natural Language Question Answering ............................... 94
    4.1.1 QA-IR Systems ............................................... 95
    4.1.2 Essay-based Question Answering ............................... 100
    4.1.3 More on Answer Types ....................................... 102
  4.2 Machine Translation Evaluation ..................................... 104
    4.2.1 Basic Strategies for Comparing Translations ............... 105
    4.2.2 Applicability to an ICALL Task ............................. 112
    4.2.3 An MT Evaluation Approach to Automatic Grading ......... 115
  4.3 Summarization and its Evaluation .................................. 117
    4.3.1 A Basic Summarization Strategy ............................. 118
    4.3.2 Applicability to an ICALL Task ............................. 121
    4.3.3 Summarization Evaluation .................................. 124
  4.4 Paraphrase Recognition .............................................. 127
    4.4.1 Techniques in Paraphrase Recognition ....................... 129
    4.4.2 Applicability to an ICALL task ............................. 131
  4.5 Summary of Design Insights ......................................... 131

5 CAM Design ............................................................... 135
  5.1 Architecture Overview .............................................. 136
  5.2 Analysis Filter ..................................................... 138
  5.3 Linguistic Annotation ............................................... 141
    5.3.1 Sentence Detection and Tokenization ......................... 142
    5.3.2 Morphological Analysis ...................................... 144
    5.3.3 POS Tagging .................................................. 145
    5.3.4 Spelling Correction .......................................... 147
    5.3.5 Synonym Identification ...................................... 149
6.3.2.4 Reducing Diagnosis to Binary Decisions .......................... 244
6.3.2.5 Summary of Classifier Development ................................. 244
6.4 Evaluation Against the Test Set ........................................... 245
6.4.1 Correct Diagnosis Examples ............................................. 247
6.4.2 Incorrect Diagnosis Examples ........................................... 249
6.4.3 Form Errors and Semantic Error Diagnosis ............................ 253
6.4.4 Alignment Properties and Diagnosis .................................... 257
6.5 Summary ............................................................................. 258

7 Discussion .............................................................................. 260
7.1 Corpus Considerations ........................................................ 264
7.2 On Representative Target Responses ....................................... 265
7.3 Improving the Comparisons .................................................. 266
7.3.1 Question Complexity and Classification ................................. 266
7.3.1.1 Classifying by Learning Goals .................................. 267
7.3.1.2 Classifying by Knowledge Sources ............................... 269
7.3.1.3 Classifying by Text Type ............................................. 270
7.3.1.4 Classifying by Question Format .................................. 271
7.3.1.5 Classifying by Answer Type ........................................ 273
7.3.1.6 A Final Note on RC Question Taxonomies ..................... 273
7.3.1.7 RC Question Taxonomy .............................................. 274
7.3.1.8 Question Classification and Performance ......................... 277
7.3.2 Improvement of Alignment Between Responses ...................... 278
7.3.2.1 Improving Target Responses ...................................... 278
7.3.2.2 Recognizing Answer Types .......................................... 279
7.3.2.3 Improving Alignments ................................................. 282
7.3.3 Developing Machine Learning for Diagnosis ......................... 283
7.4 Summary ............................................................................. 285

Appendices ................................................................................ 288
A Glossary of Terms ................................................................. 288
B The BLEU Metric ................................................................. 293
C NLP Tags and Labels ............................................................. 296
D Traditional Marriage Algorithm Example .................................. 300

Bibliography ............................................................................. 302
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 Screenshot of a Sentence Builder Exercise in German Tutor.</td>
<td>15</td>
</tr>
<tr>
<td>2.2 Example of a Pseudo-semantic Structure (PSS)</td>
<td>32</td>
</tr>
<tr>
<td>3.1 The Language Exercise Spectrum</td>
<td>74</td>
</tr>
<tr>
<td>4.1 A Screenshot of a Graded Response in Willow</td>
<td>116</td>
</tr>
<tr>
<td>5.1 CAM Architecture Overview</td>
<td>137</td>
</tr>
<tr>
<td>5.2 Dependency Triples for Response Pair</td>
<td>158</td>
</tr>
<tr>
<td>5.3 Proportion of Unmapped Concepts and Relevant Meaning Errors</td>
<td>210</td>
</tr>
<tr>
<td>6.1 Example Corpus Data</td>
<td>220</td>
</tr>
<tr>
<td>6.2 Target Response Construction Guidelines</td>
<td>224</td>
</tr>
<tr>
<td>C.1 Semantic Types Adapted from Li and Roth (2002)</td>
<td>298</td>
</tr>
</tbody>
</table>
LIST OF TABLES

3.1 Example Exercises and their Properties ............................. 79
3.2 Example Comprehension Questions from Brown and Hood (2002) ...... 85

4.1 CL Tasks Related to Content Assessment ......................... 93
4.2 ROUGE-L Scores for Hypothetical Candidate Summaries ............... 125

5.1 NLP Tools in CAM ............................................. 142
5.2 Lemmatizer Examples ............................................ 145
5.3 Similarity Module Results ........................................ 150
5.4 NP Chunker Comparison ......................................... 154
5.5 Answer Types in the Development Data ............................ 156
5.6 Pronoun Resolution Scoring Rubric ................................. 164
5.7 Lemma Overlap Comparison Data: Average Recall .................... 170
5.8 Lemma Overlap Comparison Data: Recall Below Threshold .......... 170
5.9 Match Failure Types and Counts ................................ 172
5.10 Suggestions for Improving Concept Matches ........................ 173
5.11 CAM Mapping Modules .......................................... 174
5.12 Parallels Between Concept Matching and Marriage Problems .......... 181
5.13 Mapping Types and Examples .................................... 182
5.14 Meaning Errors Based on Target Modification ...................... 206
5.15 Features for Machine Learning of Diagnosis ....................... 214

6.1 Targeted Reading Skills for Level 3 ALP Course .................... 219
6.2 Discarded Questions from Development and Test Sets ................ 221
6.3 Student Submissions for Development Data Questions ............... 221
6.4 Student Submissions for Test Data Questions ........................ 222
6.5 Learner Native Languages ....................................... 223
6.6 Diagnosis Codes and Descriptions ................................ 225
6.7 Inter-rater Agreement and Reliability on the Development Set ........ 230
6.8 Inter-grader Agreement and Reliability in the Test Set ............. 231
6.9 Breakdown of Data Sets by Judgment Labels ........................ 233
6.10 Hypothetical Performance Metric Example Data .................... 234
6.11 TiMBL Distance Metrics ........................................ 241
6.12 Development Set Performance with(out) AA Pairs .................... 242
6.13 Development Set Diagnosis Performance with(out) Detection Feature . 243
6.14 Detection Performance ......................................... 246
CHAPTER 1

INTRODUCTION

In the domain of second-language learning, language exercises evaluate learner knowledge on at least two linguistic dimensions—meaning and grammatical form. To be clear, meaning here refers to the semantic information, or content, conveyed by the answer given to an activity, henceforth referred to as a learner response. Depending on the particular teaching goals or learning situation, human instructors may emphasize the form, content or both in grading and providing feedback to the language learner.\(^1\)

This distinction between form and content is a useful one for conceptualizing the current work and will be referred to throughout this thesis. The term form assessment will be used to refer to the analysis and diagnosis with respect to the grammatical structure of the learner response. The analysis and diagnosis with respect to the appropriateness of the meaning conveyed by a learner response will be referred to as content assessment.

Still, as with most language-related issues, the actual realization of a conceptual idea is not so simple. As Garrett (1995, p. 351) points out, “...meaning cannot be communicated in language without taking linguistic form.” Thus, the form-content

\(^1\)Of course, other dimensions of evaluation might include cultural awareness, often taught as part of language-learning curricula. Such dimensions are orthogonal to this thesis.
dichotomy in response evaluation is not really a dichotomy. There is no way to evaluate the content of a response except through the form used to express that meaning. But regardless of the sometimes blurry line between form and content assessment, for intelligent computer-aided language learning (ICALL) applications, adequate feedback to learner responses to language activities must address both form and content, as needed. This thesis explores to what extent ICALL content assessment is feasible, without extensive linguistic and world knowledge representation.

1.1 Motivation

The motivation for the current study of ICALL content assessment comes from three directions – ICALL research itself, the needs of language teaching, and language processing technology.

1.1.1 From an ICALL Perspective

First, as will be discussed in depth in Chapter 2, research on ICALL systems has focused largely on grammatical form assessment (Holland et al., 1993). Existing ICALL systems (Heift, 2001; Schneider and McCoy, 1998, among others) detect and provide feedback for incorrect grammatical forms in a variety of exercises. Knowledge of the expected forms of responses may be computed based on the linguistic knowledge encoded in the system or (partially) stored in a target response in an activity model for an exercise. But how do such systems determine whether the content of a learner response is compatible with the expected content of a target response (i.e., the correct answer)?

For most CALL and ICALL systems, the compatibility of content in learner and target responses is simply determined by string matching: if the learner response
string is identical to that of a target response stored in the activity model of an exercise, then the content of the learner response is interpreted as correct (i.e., compatible with the expected content of the target response). This approach has the distinct advantage that it is efficiently and reliably achievable given current technology. But this type of content assessment, which involves matching meaning through matching surface form, severely restricts the types of exercises that can be offered to learners given that the expected responses must be tightly controlled. And tightly controlled responses do not conform to notions of communicative language teaching, as will be discussed in the next section.

A simple, string-matching approach to content assessment is too limited for an ICALL system implementing exercises where the learner response is unconstrained. In these cases, content assessment requires more sophisticated linguistic processing in order to determine whether the content conveyed by a learner response is compatible with the expected content of a target response. Any exercise for which it is not feasible to list all possible answers in the activity model is such an activity. But while there has been substantial research into sophisticated content assessment involving full semantic processing, there is no ICALL system used in real-life language teaching that successfully employs such an approach. And deep syntactic and semantic analysis of all possible unrestricted, real-world responses, which may contain learner errors, is beyond the scope of current research and technology. Thus, there is a need for research into ICALL content assessment that provides shallow semantic processing.

1.1.2 From a Language Teaching Perspective

A general trend in language instruction also drives the need for more sophisticated ICALL content assessment. Holland et al. (1993) argue that the existing ICALL emphasis on form-based feedback limits the usefulness of ICALL systems to instructors
who would rather use activities that highlight aspects of meaning. In other words, an
emphasis on form assessment runs counter to the communicative focus of language
teaching since the 1970s and its current generalization in task-based instruction.

At the core of communicative approaches to language teaching is the idea that
because language is used to communicate, this communicative function should be the
focus of language instruction. That is, because the ability to effectively communi-
cate goes beyond knowledge of vocabulary and linguistic structures, language-learning
activities should not only take into consideration linguistic competence, but also com-
municative competence (Hymes, 1972). This communicative competence encompasses
knowledge of when and how to use language to achieve a particular goal. Models of
communicative language teaching (e.g., Canale and Swain, 1980; Bachman, 1990)
build on this idea of different types of competency contributing to second language
proficiency. Language activities based on such models place a greater emphasis on
the exchange of meaningful language than on linguistic drills. Simulation of real-life
communication situations is preferred (Cheng, 1980).

Task-based instruction (Nunan, 1989), which developed from the communicative
teaching tradition, organizes language teaching around natural tasks such as
reading newspapers, writing and editing essays and other communicative activities
that emphasize meaning. As tasks rooted in real-life activities, such language exer-
cises tend to involve only loosely restricted responses from learners. In other words,
such responses may be of any length or form, with few (if any) constraints imposed
by the activity itself. Moreover, such responses often require both form and content
assessment in answer evaluation and error diagnosis. Thus, to meet the pedagogical
needs of many language instructors, there is a need for ICALL systems that provide
accurate content assessment that allows for, if not unconstrained responses, at least
less-restricted responses from students.
Of course, as implied by this last statement, there is a delicate balance between what language instructors might want in automatic content assessment and what the technology will support. In one study of learner response variation, Cox (2005) explores whether it is possible to predict learner responses when the tasks are open-ended. An example of one of the open-ended tasks in Cox’s study is shown in (1).

(1) Discuss what you think life will be like in fifty years. List three aspects you agree on.

After surveying instructors about the likely variation in learner responses and analyzing actual learner responses to five such open-ended tasks, Cox concluded that instructors are *not* particularly good at predicting either the form or the content of the responses that learners produce for open-ended tasks.

This lack of predictability in responses has important implications for content assessment in ICALL. If human instructors cannot predict how the language learner will respond, then they cannot specify in advance what constitutes an appropriate target response. If there is no pre-defined target response, then the instructor must rely on extensive knowledge of the student, the language and the world to evaluate content appropriateness. Encoding such knowledge in an ICALL system is not feasible on a large scale. Thus, the scope of the current work excludes open-ended activities with no clearly defined target response.

It is worth noting that this exclusion is in concert with the view adopted here that the role of ICALL in language instruction is to supplement, not replace, human instructors. As supplementary tools, ICALL systems should ideally provide instructors with exercise materials that can be integrated into instruction without conflicting with the pedagogical approach underlying that instruction. But ICALL applications do not have to provide all types of activities in order to be useful.
As will be discussed in more detail in Chapter 3, this still leaves a large range of language activities to consider for ICALL content assessment that emphasize meaning to a degree that is inline with current language instruction pedagogy and that might be successfully processed automatically.

1.1.3 From a Language Processing Perspective

Zock (1996) suggests that until the mid-1990s, the natural language processing (NLP) community largely ignored CALL research, and the CALL community largely ignored NLP research. While ICALL systems (i.e., CALL systems that use NLP technology) are somewhat more available now than a decade ago, this suggests there is still a lot of opportunity to contribute to the development of CALL through the application of NLP techniques to ICALL systems.

Until recently, research into morphological and syntactic processing has dominated the development of NLP technology. This is only natural considering that in order to obtain more abstract linguistic notions such as a propositional representation of meaning, one must first analyze the form and combination of directly observable linguistic units, such as words. In order to process these directly observable units, technology such as tokenizers, lemmatizers, part-of-speech taggers, etc. must be available first. Partly due to the fact that NLP tools for morphological and structural processing are readily available, whereas tools for accessing or deriving meaning generally are not, most existing ICALL systems have understandably addressed form assessment to a much greater extent than content assessment.

These ICALL systems use parsers to analyze and evaluate language forms produced by learners. While the technology supports parser-driven form analysis, current pedagogy calls for content analysis as well. The current work seeks to lessen this tension by exploring the extent to which effective content analysis is possible
in ICALL systems. To be clear, the term form analysis is used here to refer to the processing of the surface form in order to assess the morphological and syntactic properties or the semantic content of a learner response. The latter naturally is rather restricted given that a difference in form does not necessarily indicate a difference in meaning; in other words, the same meaning can often be expressed in different ways. Form identity can indicate content identity, but even this is not foolproof and requires sufficient context for disambiguation given lexical, morphological, and syntactic ambiguity. Content analysis in the present context refers to computing or deriving a representation of meaning in order to assess the semantic content of a learner response. The type of form or content analysis performed clearly has a direct impact on what degree of reliable content assessment is possible.

At the same time, Garrett (1995) points out that the state of the art in NLP technology places necessary limits on the types of processing ICALL can expect to tackle. However, what those limits are has not been fully explored and can be tied directly to the variation expected in learner responses. Note that expected variation may be variation in content or form of a learner response. A learner response may contain the same content as a target response, but be expressed with a different grammatical form. Or both content and form can vary to express a meaning that is compatible with, though not identical to, the target response. Thus, both form and content variation must be considered for effective content assessment. More will be said about variation in learner responses and its connection to language exercises and ICALL content assessment in Chapter 3.

1.2 The Current Research

The discussion to this point suggests that the state of the art in approaches to content assessment is insufficient for any ICALL system intended to offer learners a range of
activities, including task-based exercises. This thesis focuses on the uncharted space between the extremes of approaches to content assessment and argues for an approach that goes beyond simple string matching, while stopping short of deep semantic processing. Given the need for more flexible, sophisticated content assessment, exploration of the space between identity/string matching and deep semantic processing is warranted to determine what degree of reliable content assessment is possible within the boundaries of current technology, which level of content assessment pushes those boundaries in interesting ways, and which lie beyond the capabilities of practical ICALL systems.

The overarching goals of this thesis are

1. to conduct a comprehensive study of content assessment in ICALL,
2. to develop criteria for determining the type and degree of analysis required for successful ICALL content assessment given the properties of language-learning exercises, and
3. to evaluate the effectiveness of shallow strategies in linguistic processing for ICALL content assessment.

Through the analysis of language-learning exercises, approaches to content assessment in ICALL and related tasks, as well as the evaluation of a content assessment module (CAM), this work will argue that effective content assessment is possible for certain types of language-learning exercises without full natural language understanding. An activity model that includes a target response and clearly defined parameters for assessment (such as the level of variation allowed) makes effective processing of learner input possible.²

²To avoid confusion, an additional explanation of terminology is required. In the language-teaching literature, terms such as input and output have different meanings than they do here. There, input often refers to stimulus that the learner receives and output is the learner response.
As a result, this thesis contributes to the understanding of the relationship between language-learning exercises, the linguistic modeling they require or permit, and the automatic methods of evaluation in ICALL based on such modeling, as well as the relationship between language processing strategies and technological feasibility in practical ICALL settings.

1.3 Outline

This thesis is organized as follows:

Chapter 2 provides background in related work in the fields of ICALL and automatic grading. Abilities and limitations of the related systems are discussed, and this provides a context to situate the present study.

The focus of Chapter 3 is on a spectrum of language exercises and their properties. Within this spectrum, loosely restricted reading comprehension questions, situated in the middle ground of the spectrum, are targeted as a test case. The underlying assumption is that if content assessment is possible for these questions, it will be possible for more tightly restricted exercises too.

As part of the investigation into the effectiveness of content assessment in ICALL using shallow NLP techniques, the current work presents and evaluates a Content Assessment Module (CAM) based on these techniques. Essentially, CAM compares learner and target responses and diagnoses errors by classifying unsuccessful concept matches. Chapters 4 and 5 describe and provide motivation for the CAM design chosen. The evaluation of CAM against a learner corpus is described in Chapter 6.

Meaning is what is processed inside the learner’s head and content is what is presented to the learner in language-learning materials.

For the purposes of this thesis, input refers to the input to the ICALL system. As such, the input is the learner response. Output is the analysis the system produces. Content is the meaning expressed by the learner response which the system attempts to analyze.
And finally, Chapter 7 includes a discussion of the system evaluation with respect to implications for content assessment in ICALL and future research in the area of meaning error diagnosis.

Since interdisciplinary research such as this thesis, which is situated between language teaching and natural language processing, naturally (or hopefully) appeals to researchers from often diverse backgrounds, not all terms used here may be familiar to all readers. In order to reach the largest possible audience with the smallest amount of confusion, a glossary of terms used in this thesis is provided in Appendix A.
CHAPTER 2

PREVIOUS RESEARCH IN CONTENT ASSESSMENT

In choosing language exercises, processing techniques and feedback strategies, all CALL or ICALL system designers make decisions with respect to the level of content assessment their systems will provide. This chapter discusses various approaches to content assessment in CALL and ICALL, as well as in the closely related domain of automatic grading. The discussion is organized by the type of analysis performed for content assessment. Tools developed for automatic grading (Pulman and Sukkarieh, 2005; Wiemer-Hastings et al., 1999, for instance) are included here because, like CALL systems, automatic grading systems evaluate answer correctness. Thus, even though such systems do not typically take the next step of diagnosing errors in answers, they have direct extensions to a CALL domain.

Before discussing previous approaches to content assessment, it is worth noting that the collection of system descriptions to follow is not intended to be exhaustive. Rather, it is intended to be representative of the types of existing approaches to content assessment. However, it excludes CALL systems that use language exercises that are language-unaware in terms of processing. Such systems, which use multiple choice, matching, drag-and-drop, point-and-click or other similar activities, will not be discussed further because the automatic analysis of language in order to provide diagnosis and feedback is minimal, or nonexistent.
2.1 String Matching

One of the simplest and most common approaches to form analysis for content assessment is string matching – either character- or token-based. If the form matches in comparing the learner and target response strings, the meaning is assumed to be correct.

Character-based string matching, i.e., a matching procedure that compares lists of individual characters, only succeeds if each character in the learner response string is identical to the corresponding character in the target response string. If the target response is *Ohio State University*, then character-based matching will only accept exactly those characters in that order in a learner response. Given that character-based string matching integrates no knowledge about language at all, it fails on responses with variation that is as simple as differences in spacing, capitalization, hyphenation, or spelling. Related, a single difference can cause the entire match to be misaligned. For example, a character-based comparison of *Ohio State University* with *The Ohio State University* would not match a single character since the first string begins with *O* and the second begins with *T*.

Token-based string matching is more flexible, but still limited. Rather than compare individual characters in a string, token-based matching first splits the string into a list of tokens, where each token typically corresponds to one word. Tokenization integrates basic knowledge about what constitutes a token in the writing system of a given language and thereby eliminates misalignments following extra spaces or different length words that would cause character-based matching to fail. Additionally, it can include basic normalization of capitalization and spelling. The matching process compares tokens from the learner response to tokens in the target response. Basic token-based matching requires identity of the two lists of tokens. If that fails, the
algorithm can back off to searching for all or some of the target tokens in a sublist or subset of tokens in the learner response. This makes searching more flexible in terms of where in the string a match is attempted, but token-based string matching will fail whenever there is variation in lexical material or the structure used.

In spite of their limitations, character- or token-based string matching has frequently been employed in CALL systems. For instance, the Wida Gapmaster tool performs an identity check using strict, character-based string matching. However, a token-based string matching approach is likely more common. For instance, in an early guide to building CALL systems, Kenning and Kenning (1983) offer practical instruction in programming a CALL system in BASIC. They specifically discuss strategies for dealing with unexpected input but limit their solution to a back-off, token-based string matching approach. Ahmad et al. (1985) characterize these early systems as “linguistically primitive” in comparing learner responses to target responses.

In another token-based string matching approach, Ahmad et al. (1985) identify three basic strategies to evaluating a learner response:

1. ignoring capitalization,
2. matching keywords, and
3. allowing specific multiple correct answers.

The first strategy is obvious and the last simply involves listing every possible target response that one can think of. But their keyword strategy involves token-based string matching. In addition, if the learner response does not exactly match one of the stored target responses for an exercise (even after ignoring capitalization), then the system will search for a substring match of token sequences before rejecting the learner response as incorrect.
Though limited, such approaches can be highly effective for some exercises. For instance, exercises that have only one possible answer (e.g., word conjugation or vocabulary labeling) are easily handled by string matching. However, many exercises require a more complicated analysis. For example, a common reading comprehension activity is to have students read a passage and answer fact-based questions about the passage. Laubach et al. (1991) offer many such exercises in a workbook for practicing reading in English. For one exercise, students read a passage about a bus driver in Kansas City. A subsequent question to the learner is *In what city did the story take place?* The target response is *Kansas City.* But a learner response might be

- *Kansas City, Missouri*
- *in Kansas City*
- *It happens in Kansas City.*
- *in that town near the border of Kansas and Missouri*

For short-answer questions such as this, it is not typically possible to predict all the ways a learner might correctly respond. All the hypothetical answers above are correct, but a basic string match would reject all four. A back-off string-matching procedure that only requires *Kansas City* to be in the learner response would accept the first three responses and fail on the fourth. Back-off string matching relies on a finite approximation of a potentially much larger set of more complex expressions that need to be recognized as correct learner answers. One problem of such an approach is that the method ignores the contribution of substrings in the learner response that are not part of the matched target string. Suppose the learner response is *in a town near Kansas City.* This is not a correct response, but the back-off string-matching algorithm would fail to detect the mismatch. This example illustrates that character- or token-based string matching is likely to fail for content assessment whenever variation occurs.
2.1.1 Controlling Variation in Student Responses for Effective String Matching

One solution to the problems of string matching is to design exercises that limit learner input such that there are no unexpected correct forms, i.e., unexpected form variation will always correspond to an error. One ICALL system incorporated into real-life instruction that successfully applies this method is German Tutor (Heift, 2001). German Tutor uses restricted exercise types and a pipeline of modules to check for grammar errors. For instance, it checks for spelling errors, missing words, extra words, incorrect word order, and other grammar problems. But, for the most part, German Tutor exercises do not allow variation in answers. For example, Figure 2.1 is a screenshot of a sample exercise in which learners are tasked with building a grammatical German sentence from words provided in the exercise.1 Because word forms are supplied, legitimate variation is avoided.

The system offers activities such as build-a-sentence, dictation, word-order practice and fill-in-the-blank. By design, these activities restrict responses to include only expected lexical material. If there are several possible target responses, all of the possible targets are stored. Content assessment is performed by string matching against each of the stored targets, but only an exact match is considered correct.

Because of the tightly controlled exercise types and lack of variation in the expected input, the assumption that any variation in a learner response is due to form error, rather than legitimate variation, is a reasonable one. More sophisticated content analysis is unnecessary. At the same time, this severely restricts the types of exercises that can be used in the German Tutor system.

The most important benefit of string matching is its simplicity. Compared to other forms of matching, to be discussed below, string matching – either identity checking or back-off string matching – is easy to implement. And with the proper restriction on exercise types as in German Tutor, string matching can produce reliable results.

The biggest drawback of using string matching is that it is insufficient for evaluating the content of exercise types that allow response variation. Sentence-building exercises in which the words or lemmas are not given in advance allow too much variation in expected input. Since a system typically cannot store all possible acceptable variants as target responses, string matching – even using a back-off matching strategy – is likely to be insufficient for matching learner and target responses.

2.2 Pattern Matching

Pattern matching adds an additional layer of abstraction and processing capability to string matching. Acceptable variants and optional components of target responses
can be expressed compactly using patterns, which can then be used to search learner responses for a match. For example, Pulman and Sukkarieh (2005) use a manually constructed set of patterns to grade short-answer questions given on a standardized test for high school students in the United Kingdom.\(^2\) Their study focuses on identifying concepts in learner responses, which range in length from short phrases to 2-3 sentences. First, they tag and chunk a learner response to identify parts of speech, as well as noun and verb groups. To this chunked text, the hand-built patterns containing target response concepts are applied. If the input fails to match a pattern supplied for a target response, it is marked as incorrect.

Patterns were constructed manually by examining a set of correct responses for each question. These responses essentially can be thought of as paraphrases of one another, and the regular expression is a pattern representing all the paraphrases as a single target response. The only example pattern provided by Pulman and Sukkarieh is reproduced in (2). The pattern is for a question on the development of twins.

\[
(2) \text{ singular}\_\text{det} + \langle\text{fertilised egg}\rangle + \langle\text{split}\rangle; \langle\text{divide}\rangle; \langle\text{break}\rangle + \text{in, into} + \langle\text{two}\_\text{halves}\rangle,
\]

where

- \text{ singular}\_\text{det} is one of (the, one, 1, a, an)
- \langle\text{fertilised egg}\rangle is a noun phrase containing the strings ‘fertilised’ and ‘egg’, though not necessarily consecutively
- \langle\text{split}\rangle is one of (split, splits, splitting, has split, etc.)
- \langle\text{divide}\rangle is one of (divides, which divide, has gone, being broken, etc.)
- \langle\text{two}\_\text{halves}\rangle is one of (two, 2, half, halves)

\(^2\)While their strategy is not specifically designed for language learning, their approach is applicable to content assessment in ICALL since their goal is to automatically grade the semantic content of loosely restricted responses.
Patterns such as (2) are appealing because they are intuitive, simple to implement and straightforward to construct (given part of speech tags and phrasal chunks). And if the activities have a finite number of predictable answers, patterns or regular expressions can fully describe the target responses. However, there are two obvious drawbacks to implementing pattern matching in the content assessment module of an ICALL system. First, constructing answer patterns by hand is very labor-intensive. Pulman and Sukkarieh experimented with methods of automatically learning the patterns, but the results were not promising. To automatically learn reliable patterns, reliable and sufficient data are required. This problem is independent of whether the system is an ICALL system or not, but severely limits the effectiveness of learning answer patterns in an ICALL setting.

And second, with many short-answer questions, it is unlikely that any pattern will be able to cover all possible learner responses. As the authors themselves note, the above pattern will fail to match a correct answer such as one sperm has fertilised an egg...that split in two because the strings ‘fertilised’ and ‘egg’ are not both in a single noun phrase. While adding another pattern might account for this particular variation, there are likely many other variations not accounted for in the patterns or present in the example data. Successfully predicting all possible variants to exercises with loosely restricted responses, in an ICALL system or otherwise, is unlikely.

To mitigate the problem of predicting variants, another automatic grading system employing regular expressions, WebLAS (Bachman et al., 2002), uses WordNet (Miller, 1995) to identify variation. WebLAS parses target responses, searches WordNet for lexical alternatives, and creates regular expressions for scoring learner responses. The scoring key uses the regular expressions to match target elements to learner response elements similar to the Pulman and Sukkarieh (2005) system. But WebLAS requires instructors to assign particular point values to the elements in the
target responses to construct an appropriate scoring key for the variants found in WordNet. However, while using WordNet to automatically construct regular expressions takes the burden off the system designer to pre-envision valid lexical alternatives, it does not account for legitimate syntactic variation in the patterns.

2.3 Lexical Analysis and Matching

Approaches that focus on content assessment through lexical analysis, beyond how WebLAS uses WordNet, are worth additional mention. One system that illustrates the lexical matching approach is the Intelligent TErminology Learning System (ITELS) (Dicheva and Dimitrova, 1998). This tutoring system is designed to facilitate learning English vocabulary terms in domains such as computer science. The types of exercises supported by ITELS include multiple choice, multiple answer, fill-in-the-blank and matching. For all exercises, the system includes a detailed lexicon that stores parts of speech, inflection information, and meanings of words (i.e., what concepts they represent). Also included as a resource to ITELS is a hand-built ontology relating various concepts within the relevant domains. For multiple choice, multiple answer and matching questions, no language-aware processing is required because learners choose the correct answer from a list. However, the ontology is key to the system’s ability to assess the content of learner responses for the fill-in-the-blank activities.

To illustrate this lexical analysis approach, consider the exercise in (3).³

(3) **term**: object program

**kind**: fill_in_the_gaps

**type**: term understanding

**content**: Fill in the blank with the correct term:

“A __________ is the translation of a source program into an object language.”

³Exercise example from Dicheva and Dimitrova (1998).
In (3), the target response is the term *object program*. First, a learner response would be string matched against this target. The exercises are designed such that there is typically only one possible correct answer. Thus, a string match analysis *could* be employed exclusively to assess the content in responses to ITELS activities. However, Dimitrova and Dicheva make use of their domain ontology to provide feedback in the cases in which the learner response does not match the target response. This point is worth emphasizing because while it might be the case that string matching can be used to determine whether a response contains an error, error diagnosis requires linguistic knowledge or processing.

The next analysis step locates the learner and target concepts in the ontology and calculates the distance between the two terms. Nearness in the ontology is determined by counting the number of links required to connect the learner response and the target response. This information is used to provide additional feedback to the learner about how closely related the learner response is to the target response (i.e., whether it is a subtype, a supertype, a sister concept, etc.).

Assuming an ontology is available for the target language, incorporating it into an ICALL system for content assessment is appealing. Ontologies minimally provide information on hypernym-hyponym relations. If a learner response includes a term such as *animal* and the equivalent part of the target response is *dog*, then providing the system with the knowledge that *dog* is a hyponym of *animal* can be useful for providing feedback on the error, as in the ITELS system. It might also be useful for content assessment that distinguishes between responses that convey the same meaning as a target, acceptable responses that convey partial target meaning, and responses that contain errors.

---

4It can be argued that using raw counts between concepts as a measure of distance is potentially flawed if the hierarchy has uneven distribution of concepts. Refer to the later discussion of WordNet in this section for elaboration.
unacceptable responses that convey partial target meaning, and responses that fail to convey the target meaning. For instance, consider the hypothetical scenario in (4).

(4) **Exercise:** The learner is given a picture of a boy playing with a dog and asked to describe in a single sentence what the boy is doing using a present progressive verb form.

**Target Response:** The boy is playing with a dog.

**Learner Response:** The boy is playing with an animal.

If one of the goals of the exercise in (4), as defined in the activity model of a system, is to test the learner’s knowledge of the vocabulary term dog, then the response is wrong. But it less wrong than The boy is playing with a monkey. Thus, the response is unacceptable, but not entirely wrong. If the primary goal is to test the learner’s knowledge of how to form the present progressive in English, then the learner response might be assessed as correct. If both the present progressive and, to a lesser extent, the vocabulary are the target learning concepts, then the learner response might be partially correct, but less ideal than the model target.

In short, having access to relationships between lexical items through an ontology makes it easier for an ICALL system to have a content assessment module that 1) can recognize possible alternatives not explicitly listed in the target responses of an activity model and 2) is flexible in distinguishing between levels of unacceptability.

However, there are complications in using an ontology for content assessment. First, the issue of coverage is nontrivial. WordNet (Miller, 1995) is probably the most well-known general-purpose ontology for English, and versions of WordNet exist for many other European languages as well. But a general-purpose ontology may not be sufficient for an ICALL system that targets terminology in a specific field. At the
same time, domain-specific ontologies, such as the one for the ITELS system, must largely be created by hand, and ontology construction is a time- and labor-intensive process.

Assuming a general-purpose ontology is available and provides the required word coverage, there is an important issue of how best to use the resource. Using any general-purpose ontology for an ICALL system raises at least two main issues:

1. How much can the ontology be trusted?
2. How is relatedness between concepts in the ontology measured?

Given its widespread use and availability, WordNet will be used to illustrate these issues. In evaluating how much WordNet can be trusted as a reliable resource of lexical knowledge for English, it is relevant to look at both the lexical coverage of the resource and the richness of the associations within the database. WordNet includes roughly 150,000 lemmas representing nouns, verbs, adjectives and adverbs. It is organized by *synsets*, which can be thought of as representing concepts. Each synset is a set of related terms that might all be substituted for one another *in some context*. Relations including hypernymy, meronymy, synonymy and antonymy are defined between synsets. While it covers a large number of English lemmas, one area of coverage that would be particularly useful to an ICALL application but lacking in WordNet is multi-word expressions (e.g., *in the wake of*). Wray (2000) notes the importance of such expressions, which she refers to as *formulaic sequences*, to second language learning. Thus, their proper treatment should be considered in ICALL systems.

---

5In the *right context*, all the members of a given synset are interchangeable. Knowing what that context is and whether it matches the context provided by the user is a serious challenge.
The richness of the associations within WordNet is also variable. Nouns lend themselves to taxonomic organization, so the noun hierarchy is well developed. However, some subhierarchies within the noun taxonomy are better developed than others. For instance, biological concepts have a well-developed and relatively deep hierarchy compared to other concept clusters. On the other hand, verbs are not as easily placed into a hypernym-hyponym hierarchy (or an equivalent tryponym hierarchy), but such an organization is attempted. Adjectives and adverbs receive still less attention in the ontology.

This is not to say that WordNet is not a valuable resource. On the contrary, there is clearly a wealth of information within WordNet, but how useable is it in an ICALL setting? It is arguably crucial to provide a language learner reliable and accurate information about a lexical item.\(^6\) The ambiguity inherent in natural language is reflected in WordNet. This, in turn, makes reliably accessing the intended synset for a particular term nontrivial, given that a term can appear in multiple synsets to reflect multiple meanings. This points to the second issue in ontology use – how relatedness is measured.

To measure how related two terms are (in order to determine whether one is a acceptable substitute, for instance), the correct synsets (i.e., the ones that represent the intended meaning of the terms in context) must first be identified. This process is complicated by the ambiguity problem. But assuming the correct concepts have been identified, the relatedness of the synsets must be measured. A typical approach to this task is to define relatedness as a measure of the distance between the relevant synsets.

McHale (1998) describes and compares several such distance measures using both WordNet and Roget’s Thesaurus (Berrey and Carruth, 1962). The simplest

\(^6\)The assumption is that it is worse to give a language learner incorrect feedback than no feedback.
distance measure is to count the number of nodes or, more often, the number of edges between two concepts. A node is a concept; an edge is the relationship that links two concepts. However, as McHale notes, some concepts have far more developed subhierarchies than others and edges are all of equal conceptual weight. As a result, using raw counts alone produces intuitively unsatisfactory distances in some cases. As an example, McHale comments that the distance between Intellect and Grammar is identical to the distance between Grammar and Phrase Structure if edges are simply counted because the same number of edges connects the concepts in the pairs. Yet, Grammar and Phrase Structure seem closer conceptually than Intellect and Grammar. To counteract these issues, other researchers have suggested alternative distance measures based on taxonomy density (Agirre and Rigau, 1996) and depth (Richardson and Smeaton, 1995).

McHale notes such issues arise no matter what the taxonomy is. Thus, replacing WordNet with some other general-purpose ontology is likely to replace WordNet’s processing issues with some other set of issues. But this is not to say that ontology use is too issue-ridden to bother within ICALL. It just means that choices on when and how to use an ontology like WordNet must be carefully made.

2.4 Syntactic Representation Analysis and Matching

It can be argued that any ICALL system that implements a syntactic analysis module is performing, at least, a minimal content analysis because syntactic analysis identifies subcategorization requirements (argument structure) and perhaps selectional restrictions. From this perspective, Heift’s German Tutor system described in Section 2.1 falls into the category of systems that perform syntactic analysis for content assessment. But in German Tutor, syntactic representations of learner input are not compared to syntactic representations of target responses. Rather, a learner response
is analyzed syntactically to find grammatical errors using an HPSG parser modified to allow error diagnosis. Thus, content analysis is not the goal of parsing in German Tutor.

**Grammcheck/GETARUN.** Similarly, another ICALL system performing syntactic analysis is *Grammcheck* (Delmonte, 2003), one of several applications of Delmonte’s GETARUN system, which is a question answering and information extraction text understanding system. In *Grammcheck*, Italian students of German are given lemmas and required to construct well-formed German sentences similar to German Tutor’s Build-a-Sentence activity. As with Build-a-Sentence, students are supplied with all the lemmas needed, and no lexical analysis is required.

As with any syntactic processing in an ICALL task, the *Grammcheck* parsing must handle potentially ill-formed input. Heift’s system uses a full syntactic parser to identify ill-formedness. Delmonte’s system, on the other hand, only chunks input with a shallow parser based on Lexical Functional Grammar (LFG). The shallow parser identifies any constituent chunks it can, then passes the information to a module that maps the partial constituent structure (c-structure) to a functional structure (f-structure), identifying errors through the mapping process. The mapping algorithm is responsible for identifying features of lexical heads, computing agreement between constituents based on the features and imposing rules of consistency to ensure the correct number and type of arguments are present for each relevant predicate. Delmonte assumes that the students using *Grammcheck* are likely to err by using Italian structures for German sentences because Italian is the students’ native language (L1). These L1 transfer errors are anticipated by allowing the shallow parser to accept either Italian or German syntactic structures and leaving the functional mapping module to sort out which structures are errors.
A second application of GETARUN allows linguistics students to type in a sentence and have it analyzed using a full version of the LFG parser. This particular application is specific to students learning LFG and may have limited applicability to a wider language learning population. However, the application provides a full parse for each input sentence. Like Grammcheck, content assessment takes place as part of the syntactic analysis of arguments because the analysis makes use of selectional restrictions and thematic role requirements. However, the application is one in which students are not required to answer questions. They simply type a sentence of their choosing and analyze it with GETARUN. Thus, no comparison of syntactic representations is required.

ICICLE. Yet another system performing indirect content analysis through syntactic processing is the Interactive Computer Identification and Correction of Language Errors (ICICLE) system (Schneider and McCoy, 1998). ICICLE is designed for American Sign Language (ASL) signers learning written English. Students write a paragraph or essay and submit it to ICICLE for correction. The system proceeds sentence-by-sentence, identifying any grammatical errors it finds. Unlike German Tutor, which encodes error possibilities into its HPSG grammar constraints, and Grammcheck, which uses shallow parsing and mapping rules, ICICLE uses mal-rules for error detection. Mal-rules capture likely student errors in the form of rules incorporated into the grammatical rule set used in the parsing process. These mal-rules enable a parser to successfully parse ill-formed input, but the strategy requires that all possible student errors are anticipated in advance.

Examples of a grammatical rule and a mal-rule in ICICLE for number agreement errors in determiner phrases (DPs) are in (5a) and (5b), respectively.7

7The notation ?a refers to a variable for the value of the feature agr.
(5) a. DP(agr ?a) → Det(agr ?a) NP(agr ?a)
   b. DP(agr s) (error +) → Det(agr (?!a s)) NP(agr s)

The grammatical rule in (5a) states that a DP consists of a determiner (Det) followed by a noun phrase (NP), where the agreement feature of the DP, Det and NP match. In (5b), the mal-rule rule states that a DP with an error feature attached to it consists of a Det with a non-singular value for its agreement feature followed by a NP that is marked as singular. Adding this mal-rule allows the parser to provide a successful analysis for an ungrammatical phrase such as *Those book* in *Those book is on the table*, and the error feature attached to that analysis is later used in feedback to the learner about the form errors. However, again, there is no need for explicit content assessment in ICICLE because there is no pre-defined expected content, only expected form.

2.4.1 A Note on Indirect Content Analysis

Because the dividing line between form and content analysis is not a solid line, it can be argued that all of the discussed syntactic approaches perform limited content analysis through form analysis. But what analysis is performed is specifically for assessing form, not meaning. Still, these approaches are relevant, as will become apparent in the section below on semantic analysis matching. Full or partial syntactic parsing typically precedes semantic processing for content assessment. Thus, output from the approaches outlined above might feed into more explicit semantic processing.
2.4.2 Syntactic Representation Matching

The final system to be discussed in this section is BANZAI (Nagata, 2002), a tutoring system for learners of Japanese. Unlike the syntactic analysis modules above, the analysis in BANZAI does involve comparing the syntactic representations of the learner and target responses. An example exercise in BANZAI is listed in (6).

(6) You found an interesting place where couples can hang padlocks to affirm their love. Tell Ms. Satoo that when you went to the top of Enoshima, there was an interesting place.

As this example shows, the exercise setup carefully controls the variation in learner responses by telling the student how to respond. In addition, each exercise has a target response in which sentences are represented as sequences of words grouped with their parts of speech, root forms (for verbs and adjectives) and grammatical functions (for particles). Acceptable alternative words or phrase and optional words, such as particles, are noted in the target response as well.

Learner responses are segmented into words, which are passed to a morphological analyzer to identify noun-noun compounds and verb forms. The output of the morphological analysis is used to remove any alternate or optional words in the target response not present in the learner response. The result is an intermediate representation of the learner response and one for the target response. These are compared to determine whether the learner response has too many or too few words, unknown words or conjugation errors. If one of these errors is present, an error message is produced and stored for later presentation to the student.

---

8BANZAI has been marketed as Robo SENSEI, which offers five exercise types – word-level fill-in-the-blank for vocabulary testing, noun and verb phrase production, sentence production based on translation tasks, fill-in-the-blank paragraph completion and sentence dictation.

9Example exercise from Nagata (2002).
Regardless of whether there is an error in the intermediate representation of the learner response, both learner and target responses are parsed. The BANZAI parser builds constituent structures for the learner and target responses, using a bottom-up strategy and context-free rules. An error detection module then compares the syntactic representations and differences are flagged for feedback to the learner. Additional details on the syntactic matching process are not given in the available literature.

The two sources of variation in learner responses accounted for in BANZAI are lexical and structural variation. For lexical variation, word or phrase alternatives are explicitly listed in the target response specified for an exercise. Structural variation is handled by parsing the target and learner responses and comparing the resulting structures. In either case, while there is more potential variation in responses to exercises in BANZAI than in many of the other ICALL systems mentioned to this point, the expected variation is limited significantly by the specific exercise setup.

2.4.3 Syntactic Analysis and Unexpected Input

Most ICALL systems applying syntactic analysis typically do so to diagnose whether sentences have internal errors, not whether one sentence is equivalent to another sentence. Thus, these approaches can identify form errors and determine whether a sentence contains appropriate arguments (i.e., contains the necessary arguments of the correct phrase type and perhaps the correct semantic type). While that alone is no small challenge, it does not enable such systems to recognize legitimate variation in form or content. For instance, suppose the learner is given a build-a-sentence exercise as in (7a), a hypothetical exercise similar to those of German Tutor or GETARUN.
a. Build a sentence using the following lemmas: window John break that

b. **Target Response:** John broke that window.

c. **Learner Response:** That window was broken by John.

An English version of German Tutor would mark the response in (7c) as incorrect for containing extra words – *was* and *by* – even though the sentence is well-formed and uses all the words required. This is because the German Tutor model uses an extra word check to flag responses that do not use only the supplied lemmas. This is less than ideal.

But syntactic analysis approaches meet with some success because they strictly control the expected input in this manner. Even the BANZAI system, which does not list lemmas, does limit expected input, typically by linking learner responses to translation. Learners are likely to remain close to the source language when they translate, rather than rephrase sentences creatively and unexpectedly.

Restricting exercises through careful wording or guided instruction is a clever and simple way of controlling learner responses. With such restriction, it is more likely that the system will see exactly what it expects to see. But to allow more freedom in learner responses, ICALL systems need strategies to be able to recognize legitimate variation in those responses.

### 2.5 Semantic Representation Analysis and Matching

Systems that perform some form of semantic representation analysis have more flexibility to offer exercises that allow less restricted variation in responses. One ICALL application designed to handle such variation in student input is FreeText (L’Haire and Faltin, 2003). As part of its error diagnosis, FreeText incorporates a semantic
checking tool, to be applied to sentences after spell checking and syntactic analysis.\textsuperscript{10} The semantic analysis and comparison are designed to catch content errors in grammatically correct responses. The basic idea, as described in L’Haire and Faltin (2003), is that learner responses are converted into so-called \textit{pseudosemantic structures} (PSSs), and these are matched to the PSSs of the target response.\textsuperscript{11}

To ensure the correct semantic concepts are present in well-formed response sentences, FreeText compares sentences on three levels. Clauses structures (CLS) are built for predication, tense, aspect, negation, etc.; DP-structures (DPS) are used for comparing noun phrases; and characteristics structures (CHS) are for adjectives and adverb comparison and matching. These PSSs can be used individually or combined into hierarchical structures of unordered PSSs.

Figure 2.2 is an example of PSS representation for the French sentence in (8).\textsuperscript{12} The arrows in the figure indicate that the PSSs are part of an unordered PSS list within the structure the arrow points to.

(8) C’est Jean qui a mangé cette pomme rouge.

\textit{It is John who has eaten this red apple.}

An advantage of matching PSSs is that the sets of structures are unordered; it is left to the previous step of syntactic analysis to determine whether the syntactic structure, including word order, is acceptable. The matching process looks for 1) the presence of all the target DPS, CLS and CHS structures in the learner response and 2) identical values for the features within those structures. If the learner response

\textsuperscript{10}Actually, it is unclear from the available literature on FreeText whether the semantic checking tool was ever activated in the FreeText error diagnosis system. Vandeventer Faltin (2003) describes the same tool, referring to it as the \textit{coherence checker}, and notes that the checker was designed, but only partially implemented. So, there are no figures reported on its performance.

\textsuperscript{11}Similar PSS structures are also used as input to the GBGen sentence generator (Etchegoyhen and Wehrle, 1998).

\textsuperscript{12}Adapted from an example given in L’Haire and Faltin (2003).
contains all the target PSSs and all the feature-value pairs match, then the learner response will match the target response. An additional benefit is that failure to match can be traced to the exact PSS structure or feature-value mismatch. This is very useful for reporting the kind of semantic error to the learner.

However, one important point is that PSSs are a set of feature-value pairs in which the values may be abstract concepts, such as 

\textit{operator:demonstrative}, or lexical items extracted from the input sentence, such as 

\textit{property:Jean}. This use of lexical items as slot fillers limits the effectiveness of the semantic matching. For example, suppose the example in \textnormal{(8)} is the target response and the learner response is identical except \textit{pomme} ‘apple’ is replaced by \textit{fruit} ‘fruit.’ In one of the DPS structures for the target response, a feature-value pair will be \textit{property:pomme}. In the equivalent DPS in the learner response, the feature-value pair will be \textit{property:fruit}. The system
has no way to identify the relationship between *pomme* and *fruit* and would mark the answer wrong. In other words, use of hypernyms, synonyms, etc. are not taken into consideration anywhere in the system. Depending on the learning goals of the activity, this may be an undesirable result.

Where the designers of FreeText used PSSs as semantic representation, a second ICALL system, the Military Language Tutor (MILT) Program (Kaplan et al., 1998), performs semantic analysis and matching using lexical conceptual structure (LCS) theory (Jackendoff, 1990). MILT developed out of the BRIDGE project (Sams, 1995) and is intended to allow a range of activity types for learners of Arabic and Spanish. Exercises include question-answering tasks about a microworld environment.

To make this feasible, MILT uses content analysis based on matching LCS structures, as described in Dorr et al. (1995). Learner responses to questions are first parsed, then analyzed using an LCS dictionary. The result is an LCS structure for each input sentence. This is compared to the stored LCS(s) for the target response(s) to determine whether the learner response is correct.

The theoretical underpinnings of LCS are tangent to the discussion here, but a crucial idea behind LCS is that language can be decomposed into conceptual primitives that hold cross-linguistically. Primitives include CAUSE, GO, BE, HERE, etc. and can be grouped into types. For instance, HERE, THERE, LEFT, RIGHT are all location primitives. Primitives can also take arguments that are composed of primitives. The MILT lexicon is responsible for associating lexical items with primitives. For example, the MILT lexical entry for the verb *like* is reproduced in (9). A whole sentence analyzed using LCS might resemble the structure in (10).  

---

13Example sentence and LCS analysis from Dorr et al. (1995).
In matching, if the learner response and the stored answer use the same basic primitive, then the matching algorithm descends recursively into the LCS structures, determining where they are alike and different. The matching algorithm also keeps track of positions in the learner response in which there is extra or missing information. Any LCS structure in the target response is assumed to be required. Extra information may not be treated as an incorrect answer as long as the learner response contains all the required structure.

Errors – essentially incorrect or missing primitives in the structure – are reported to the student. Like FreeText, the matching process in MILT makes it possible to keep track of the kind of semantic errors the student makes for later use in generating targeted feedback.

However, one limitation of LCS analysis noted by (Dorr et al., 1995) is that it fails to recognize some legitimate semantic variation. For example, if the target answer is *The student graduated* and the learner answer is *The student received a diploma at graduation*, then the learner response will be marked incorrect. In other words, the system cannot handle inferences supported by world knowledge, which human instructors often make in assessing the content of learner responses.

Given that making inferences based on world knowledge is a challenge for any semantic processing system, ICALL or otherwise, this limitation is not particularly damning for the LCS approach. But one serious drawback of the LCS approach is a
development bottleneck. The LCS dictionary contains lexical items decomposed into the allowed primitives. To build an LCS lexicon that covers a wide vocabulary would be very time-consuming, so that the resulting system cannot be easily or quickly expanded for new exercises.

A third ICALL system using semantic analysis is Herr Kommissar (DeSmedt, 1995), a role-playing game in which the learner acts as a detective interrogating a suspect. Where the primary application of MILT is a question-answering task in which the learner answers questions about a particular microworld, the sole task for learners in the Herr Kommissar system is to ask questions of the system.\footnote{\textup{(Dorr et al., 1995) do discuss a dialogue-based application of MILT in which learners play a role very similar to the role learners play in Herr Kommissar. In that discourse lesson – intended to simulate an immersion situation – learners question the system about a particular scenario such as interrogating suspects in Spanish. Discussion about this application of MILT focuses on the extra elements needed, namely a knowledge base containing facts about the scenario and a discourse planner to keep track of what has been said and how the system should respond to new learner inputs. However, it is unclear from the literature whether this application was ever fully implemented. Even if so, there is no content assessment since there are no target responses in the task.}}

Analysis within Herr Kommissar proceeds by first performing a lexical look-up that searches the Herr Kommissar lexicon, containing detailed syntactic and semantic information, for each word in the learner response. This initial look-up catches spelling errors and unknown words.

Assuming all the words are located in the lexicon, the input passes to a case grammar parser, which produces full parses for the sentences in the learner response. For each clause, the parser identifies the operative verb, which has argument slots associated with it. The parser fills each slot with the remaining material from the clause that comes closest to matching the requirements of the slot.
Then, in order to interpret parsed sentences semantically, the Herr Kommissar system translates each parse into a knowledge representation format. The Herr Kommissar domain uses the same format and includes a concept ontology, predication constraints and postulations.

The concept ontology is a partially ordered set, meaning that concepts may be related by subsumption but every concept is not related (directly or through transitivity) to every other concept through subsumption. This ontology is responsible for recognizing synonyms or paraphrases (two concepts directly subsumed by a parent concept are synonyms) and hypernyms (a concept that subsumes the given concept is its hypernym). In addition, concepts may be cross-referenced in the ontology, and this cross-referencing enables certain inferences. For example, the concepts of the action of cracking and the state of being broken might be cross-referenced.

While the concept ontology handles static knowledge about entities and actions in the domain, predication constraints and postulations define what can and cannot be said of entities and actions in the domain. Predication constraints, which are associated with individual lexical items (rather than concepts in the ontology) prevent violations of selectional restrictions and other forms of inappropriate usage. Postulations, defined compositionally, describe facts about the microworld in Herr Kommissar.

Semantic interpretation of the input is evaluated for coherence and consistency given the knowledge base. In other words, a representation of the learner response is matched against the representations of postulations of static knowledge in the domain. Within the context of a session, there is also dynamic knowledge stored that can be accessed and compared to the semantic representation of the learner response. The set of static or dynamic postulations is ordered, using temporal indices, to keep track of sequences of events.
Once the semantic representation of the content of the learner response is matched against the domain model and consistency is determined, the system generates a response to the learner indicating any errors (both form and content).

The design of Herr Kommissar has two advantages. First, as DeSmedt (1995) points out, the ontology, predication constraints and postulations database all represent explicit knowledge in the domain, and this explicit knowledge can be combined to generate implicit knowledge representations. For instance, if there is a predication constraint that says that plants do not talk, then the fact that trees do not talk does not have to be specified explicitly; it will fall out of the fact that the plant concept subsumes the tree concept in the ontology. This interaction of different types of explicit information reduces the information that has to be manually added to model the microworld.

A second advantage is that the domain of the task restricts the expected input. Learners are cast in the role of investigator. As such, they asked specific fact-based questions to solve a crime. This limits the responses the system can expect to receive, which potentially makes responding easier for the system. The system is able to check semantic consistency of the input because the domain is very restricted.

However, this second advantage can also be viewed as a disadvantage. While Herr Kommissar is described as a dialogue system, it is restricted to one-way information exchange. Learners ask questions; the system provides answers. The system’s static domain model is never updated with facts supplied by the user. The system is essentially restricted to one kind of exercise in which students practice forming questions.

It is important to acknowledge there exists a body of literature on general dialogue systems that is clearly not addressed in this thesis. Reuer (1999) introduces
a model for dialogue processing specifically in a CALL context. And given that dialogue systems must support the processing of potentially unrestricted input, dialogue systems as they can be adapted for CALL or ICALL are relevant.

Nevertheless, such systems are not discussed in more detail because

1. dialogue systems are typically domain-restricted and use deep-processing strategies such as those already described for systems such as Herr Kommissar or MILT and
2. the processing strategies unique to dialogue systems (e.g., maintaining dialogue history of what has been said and conversation states) are not relevant to processing unconnected language activities.

2.6 Content Analysis and Assessment in Automatic Grading Systems

As mentioned in the introduction to this chapter, automatic grading systems are related to ICALL systems. While their goals may be different, the processing techniques are often similar. For example, the Pulman and Sukkarieh (2005) system discussed in Section 2.2 is an automatic grading system rather than an ICALL system; it was intended to score test item responses from native speakers. It is included in Section 2.2 because it relies on pattern matching as an assessment strategy. The systems included in this section all incorporate specific components for content analysis in processing because all are intended to grade largely unrestricted learner input.

Most automatic grading systems are designed to grade essays. One such example is another ICALL application of the GETARUN system (Delmonte, 2003). In that application, the system automatically generates summaries to use as the target summaries. These are compared student summaries. Content matching is used to determine the percentage of target concepts in the learner summary and looks at whether the concepts in learner summary are in the same order as the target summary.
Another essay grading system is AutoTutor (Wiemer-Hastings et al., 1999). AutoTutor uses Latent Semantic Analysis (Landauer et al., 1998, LSA, ) to evaluate student essays automatically. While not specifically designed to target language learning applications, LSA is applicable to language learning settings as well. LSA computes a matrix of terms and texts, and then manipulates the matrix to calculate a value that reflects the semantic relevance or relatedness of a term to a text. Matching learner responses and target responses involves comparing combined term vectors, defined using values from the matrix.

The LSA matrix for AutoTutor is constructed using a corpus of learning materials – text from lists of questions and problems the tutor needs to be able to address, textbooks, and articles on computer literacy (the domain for the system). These documents are split into paragraphs and each paragraph is identified as a “text.” Each row of the matrix is a word that occurs more than once. Each column is a text. Matrix cells are weighted frequencies for how often a word occurs in a text. The weighting function reflects a word’s importance in the passage and in general use.

The original matrix is broken into the product of three new matrices (one for terms, one for texts, and one comprised of a diagonal of scaling factors) using singular value decomposition (SVD).\(^\text{15}\) Some number \(k\) of the scaling factors is retained and the matrices are recombined using only the retained factors. The result is a compressed form of the original matrix in which frequency values are approximated (raised or lowered) depending on the number of factors used.

To use the compressed matrix, AutoTutor takes a learner response and computes a vector, using values in the matrix for each term contained in the learner’s response. This is compared to the vectors for each text, computed by summing all the row values for terms in that text. The distance between the response and text vectors

\(^{15}\)At this point, if the three matrices were multiplied, the original matrix would be restored.
is an indication of their semantic similarity. Since correct and incorrect answers are a part of the original training texts, a learner response can be marked as correct or incorrect depending on how close (i.e., within an empirically determined threshold) the response vector is to one of the vectors from the training set.

Wiemer-Hastings et al. (1999) found that AutoTutor’s judgments correlated with human judgments about as often as human judgments correlated with each other. Specifically, the human-system correlation was 0.49, while the human-human correlation was 0.51. However, overall, the correlations are rather low, suggesting that there is room for improvement.

Moreover, there are several critical issues with implementing LSA as the primary means of semantic analysis. First, it ignores word order and syntax in general. For example, *John loves Mary* and *Mary loves John* are treated the same, although they do not have the same meaning. Second, LSA is not as effective for short passages. Rehder et al. (1998) found that grading is less reliable for essays under 200 words. Given that many language learner exercise responses are no more than one sentence long, this is a significant issue. Third, it is unclear what exactly the semantic relatedness measurement is measuring. Antonyms are scored with LSA as having a high semantic relatedness, but are not remotely interchangeable as answers to an exercise. Thus, while LSA is an interesting idea, its use within an ICALL system needs to be carefully considered.

However, McNamara et al. (2006) discusses a grading and feedback system for tutoring native speakers in reading comprehension strategies for understanding science texts. This system, iStart, relies on word matching and LSA to provide feedback to the student about their understanding of passages by analyzing the student’s self-explanation of the passage. Using LSA and word matching, McNamara et al. (2006)
reported the system was moderately effective in evaluating students’ reading strategies. Boonthum (2004) proposed a modification of the iStart system to add matching beyond the word level by relying on paraphrase recognition to identify partial and full paraphrases between student and model answers, though it is unclear whether this was ever incorporated in the iStart system.

A fourth essay grading system is IntelliMetric (Rudner et al., 2005). IntelliMetric, which evaluates writing skills, compares a learner essay to a set of training essays to determine the essay quality. MyAccess!TM is the instructional tool that combines IntelliMetric scoring with feedback suggestions and tips for writing improvement. However, exactly how the model is created and used is proprietary information.

E-rater (Burstein and Chodorow, 1999), like AutoTutor and IntelliMetric, is another system designed to automatically evaluate and score essays. To score student essays, the system applies a “holistic” rubric that takes into consideration syntactic variety, discourse cues, text organization and vocabulary use.

E-Rater uses 52 syntactic, discourse and topical variables from 270 human-scored essays (training data) on a particular topic to build a model of the learner essay. The similarity is measured between the model of the learner essay and those stored for target essays. In an evaluation of E-Rater, compared to human judges, E-Rater assigned the same (or close to the same) score (a number between 1 and 6) 92% of the time. This level of agreement is comparable to agreement between human judges on the same task (Burstein and Chodorow, 1999).

The approach used stepwise linear regression to select the most relevant features (i.e., the best predictors of quality).
E-rater has been evaluated using essays from native and non-native speakers. Criterion\textsuperscript{SM} Online Essay Evaluation (Burstein et al., 2003) is the commercial system that incorporates E-rater as part of its instructional software to help nonnative students improve their writing.\textsuperscript{17}

Several other essay grading systems exist (e.g., PEG (Chung and O’Neil, 1997), BETSY (Rudner and Liang, 2002)). Refer to Dikli (2006), Marín (2004) and Phillips (2007) for more thorough overviews of the range of such systems. But as with AutoTutor, E-rater and IntelliMetric, these systems are limited in their usefulness for a wide-range of language-learning activities because they are designed, tailored and used primarily to evaluate writing skills for essay-length passages. These systems are not intended to evaluate short passages (1-2 sentences) as is the focus of the current research and do not perform well (or have not been tested on) on such tasks.

2.6.1 Short-Answer Automatic Grading

There are additional approaches to automatic grading to be mentioned here that are designed for short-answer responses. The first is the Willow system (Pérez et al., 2006). The adaptive version of the system (i.e., the version of the system with a learner model that adapts feedback to the individual learner) is called Willow; the older, non-adaptive version is called Atenea.

To overcome the limitations of LSA mentioned in the overview of AutoTutor, Pérez et al. (2005a) combine LSA with the ERB (Evaluating Responses with BLEU) algorithm in the Willow system to automatically assess short-answer responses. BLEU (Papineni et al., 2001) is a machine translation evaluation algorithm that relies on comparing word overlap in order to score the similarity of two texts. Before measuring

\textsuperscript{17}This crossover use of the E-Rater technology highlights the overlap in automatic grading and ICALL system research.
overlap with the ERB algorithm, words in sentences are stemmed and closed-class words are discarded. Then, the ERB algorithm calculates a modified BLEU score that takes into account both precision and recall.\footnote{An overview of BLEU is located in Appendix B. How BLEU is used in machine translation evaluation is discussed in Section 4.2.1.}

Willow/Atenea includes a database of plain-text target responses to questions; learner responses are compared to each of targets for a particular question. The individual LSA and ERB modules assign scores to the learner response and the individual scores are combined to produce a final score. In Atenea, if the student passes the question, their answer and the target answer are returned, with the matching concepts highlighted in the student’s response and the unmatched concepts underlined in the target response. In Willow, the system goes one step further in that if the student fails the question, the system provides scaffolding through a series of increasingly more specific questions to guide the learner to the correct answer.

In their evaluation of Atenea, Pérez et al. (2005a) achieved a maximum correlation of .50 with human judgments on the correctness of learner responses to definition, advantages/disadvantages, and justified yes/no questions on nine Spanish exams and to definition questions on one test constructed from Google Glossary. Interestingly, the individual contribution of LSA to the performance in the Atenea evaluation was minor, though it is not clear why.

The Willow/Atenea system is of particular interest for two reasons. First, the approach relies on surface-level processing strategies to perform content assessment. Thus, while the focus is on evaluating the correctness of the meaning of learner responses, it does not require extensive knowledge representation such as the ICALL systems of Herr Kommissar or MILT. Yet it does require a lot of training data for
the LSA component, and in ICALL settings, large amounts of data for training on positive and negative examples of learner responses are often not readily available.

Willow/Atenea is also of interest because it incorporates a feedback component for the learner. Again, although not specifically designed for language learning, the feedback on the presence or absence of concepts is directly relevant to the diagnosis and feedback desirable in ICALL. More will be said about diagnosis and feedback starting in Chapter 5.

Another potentially relevant automatic grading system is the Automated Text Marking (ATM) system (Callear et al., 2001). ATM reportedly decomposes both learner and target responses into conceptual units and their dependencies and then compares the resulting units for matches. The authors argue that ATM addresses the shortcomings of automatic essay grading systems by taking word order and dependencies between words into account. However, the available literature lacks a clear description of how the dependencies are constructed for learner and target responses and how they are compared.

A third short-answer grading system is AutoMark (Mitchell et al., 2002). AutoMark uses information extraction (IE) techniques to score responses to short-answer questions. Target responses are stored as manually constructed templates. These templates consist of concepts and relations between concepts that are expected in student responses. Essentially, templates encode the range of variation in student responses that will be accepted by the system. Responses are skimmed for template elements and scored based on whether all the slots in the template can be pattern-matched to concepts and relations found in the student response.

AutoMark was tested on single-word and sentence responses to science questions on standardized tests for elementary school students. On sentence-level comparisons, AutoMark agreed with human judgments for 111 of 120 responses (92.5%)
to test items requiring students to give short, explanatory sentence answers. On test items requiring student to provide an analysis of a pattern in presented data in single-sentence responses, AutoMark’s performance dropped to 83.3% agreement with human judgments. Agreement rates between human judges were not reported. In the explanatory items, the variation expected was less than in the analysis responses. The difference in performance on these items underscores the fact that the underlying activity type – and the student response variation permitted by the activity – has an impact on system performance. However, the high performance number reflect development and testing on the same data and do not reflect how the system would perform on unseen responses.

A final system for automatically grading short answers is another ETS research project, C-rater \(^{TM}\) (Leacock and Chodorow, 2003). C-Rater is an answer scoring engine designed to test equivalence in meaning. It requires a model answer (target response) that a learner response is mapped onto. Like the content assessment module to be developed in Chapter 5, the comparison of learner and target responses is automatic. However, the development of target responses for the system is more involved.

A model response in C-rater is constructed by a human content expert using a program called Alchemist created for that purpose (Leacock, 2004). The expert types target sentences and creates a scoring rubric that defines how many points are awarded for which parts of the answer string. Alchemist then presents similar words from the answer string to the domain expert, who selects synonyms and alternate expressions to include in the model answer. In this way, unexpected variation is reduced and partial credit may be assigned if the learner response contains some, but not all, of the essential concepts in the model. As Leacock and Chodorow point out, because a model answer is required, C-rater cannot handle the assessment of open-ended questions that
ask for opinion, fact or supposition from learners’ personal experiences or imagination. However, it does handle fact-based, short-answer responses like Willow/Atenea.

Before mapping learner responses to model answers, C-rater processes five levels of variation in learner responses, in an effort to normalize concept expressions and “neutralize” variation:

1. Syntactic variation (e.g., using a passive or active construction)
2. Reference variation (e.g., using a pronoun to refer to a previously named entity)
3. Morphological variation (e.g., using an -ing or -ed ending)
4. Lexical variation (e.g., using synonyms or other lexically related terms)
5. Typographical error variation (e.g., spelling errors)\(^{19}\)

Spelling correction is accomplished through an approach that relies on minimum edit distance. The most likely alternative is selected based on minimal distance and the semantic domain of the expected answer, as defined by words selected by the domain expert in developing the model response.

Syntactic variation is identified using the shallow parser CASS (Abney, 1996) to identify tuples that reflect argument structure. It is these tuples that are normalized into the canonical form.

Pronouns in the tuples are normalized using a pronoun resolution algorithm trained on student essays. Morphological variation is normalized by replacing inflected forms with base forms of words. Derived forms are also normalized in this way. Finally, lexical variation is normalized using a word similarity matrix (Lin, 1998a) to identify possible lexical alternatives to the words originally used in the learner response.

\(^{19}\)Leacock and Chodorow (2003) note that while spelling errors are not legitimate variation, correcting non-word errors was necessary to improve the system’s ability to identify legitimate variation.
Once each level of variation is processed and normalized, C-rater proceeds to rule-based concept matching. The rules identify conditions of equivalence such as under what circumstances an active and passive sentence are equivalent. But detailed description of the rules are not available.

The C-rater system was tested in two large-scale studies – one for student responses to math problems and one for responses to 11th-grade reading comprehension questions. On the latter, which is more similar to the present task, C-rater scored 16,625 responses to seven reading comprehension questions. It assigned a score of 0 (no credit), 1 (partial credit) or 2 (full credit) to each response. To evaluate C-rater’s accuracy, a sample of 100 of the responses were randomly selected and graded by a human using the same scoring scale. Compared to the human judgments, C-rater achieved an impressive 85% accuracy on the sample. This was compared to a simple bag-of-words baseline that compared overlap between the learner and target responses. The bag-of-words model achieved only 30% accuracy on the same data.

From the available description, the surface-based techniques applied in C-rater seem similar to those developed in Chapter 5. However, given the proprietary nature of the C-rater system, exact details on how the representations of learner and target responses are constructed and how the comparison of the two proceeds are unavailable.

Furthermore, it is important to underscore that as a grading system, C-rater (like AutoMark) was developed to score responses, not to provide an analysis for diagnosis or the kind of scaffolding feedback in Willow/Atenea. Similarly, C-rater was not tested against language-learner data. On the other hand, the ICALL approach to be developed in Chapter 5 addresses the issue of error diagnosis in content assessment in addition to developing an approach to recognizing variation using a range of linguistic information in language-learner data. Thus, directly comparing the results in the present study to those of any of the short-answer grading systems is not possible.
2.7 Conclusions

The ICALL and automatic grading approaches described in the previous sections each have benefits and drawbacks in processing and capabilities. Just looking at a few of the approaches using semantic matching, pros and cons are readily apparent. For instance, FreeText’s pseudo-semantic structures are categorized by type and allow targeting of specific semantic errors. However, its use of strings from the text to represent concepts limits its effectiveness since it cannot recognize relations between non-identical strings. MILT’s approach is motivated by LCS theory, and LCS provides the semantic representations to support recognition of lexical relationships such as synonymy (unlike FreeText). But providing the LCS lexical specifications – decomposing lexical items into primitives – is nontrivial. Herr Kommissar supports dialogue and recognizes limited inferences given a small model of facts about a particular microworld. However, Herr Kommissar does not support two-way communication, and the system is designed for only a single question-answering task in a limited domain. The potential to extend the system into new domains or add new exercise types is subsequently very limited. AutoTutor can provide a rough idea of the semantic similarity of learner and target responses by matching the response and training text vectors built using LSA. The matrix itself is simple to construct and does not rely on any expensive deep linguistic processing. However, in addition to being unintuitive, LSA is not applicable to short answers and ignores potentially relevant syntactic information.

Such a benefit/drawback summary might be provided for the remaining approaches described in previous sections. The point is none of the approaches is able to handle a full range of semantic issues in open-domain content assessment. However,
no system in any domain currently has the capability of full natural language understanding (NLU). And full NLU is not the goal here either. The existing approaches to content analysis highlight the fact that the choice of representation for linguistic information and depth of processing help determine what semantic evaluation the system can deal with and how scalable the system will be. But they also show there is need for more research.

Yet, while there is plenty of room for improvement in ICALL content assessment, existing approaches are not without potential. System designers have developed strategies to restrict expected learner responses to facilitate reliable processing. These restrictions (primarily on exercise types or domain) also largely eliminate the need for content analysis because unrestricted input is limited or avoided entirely. What remains to be determined is whether reliable results can be obtained once those restrictions are lifted, or at least loosened. The approach described in Chapter 5 explores some of the possibilities of surface-based processing techniques for content assessment. But first, the next chapter looks more closely at the nature of language exercises and how they can be classified to understand the challenges they pose for natural language processing and ICALL content assessment.
CHAPTER 3

LANGUAGE LEARNING EXERCISES

Effective ICALL starts with carefully selected language-learning exercises. As exemplified in the previous chapter, the systems that have been successfully integrated into real-life teaching situations, such as German Tutor and BANZAI, tightly control expected response variation through deliberate exercise choices that limit acceptable responses. Some systems, such as Herr Kommissar, have also reported success by restricting both the exercise type and the domain. Still other systems, such as ICICLE, focus on a single task, such as essay writing, but specifically provide grammatical or writing style feedback. And, for most of these systems, limiting the exercise types goes hand-in-hand with limiting the need for content assessment beyond form-based matching or holistic content scoring.

Thus, effective content assessment is tied directly to the type and nature of language-learning exercises. Different exercises require different levels of linguistic knowledge and carry different expectations with respect to the level of response variation. Such properties of language-learning exercises help determine the type of content assessment that is needed and whether it is possible given current technology.

In this context, it is useful to keep in mind that ICALL can be effectively and successfully integrated into teaching practice without trying to incorporate all types of language activities into the ICALL system. Different language teaching contexts have strengths and weaknesses, which can complement each other. The
traditional language teaching classroom readily supports role play and oral interaction
given that the teacher and the students are present in person, but individual learner
differences, class sizes, and limited teacher availability make it difficult to provide
individual feedback to all learners or to foster their awareness of language forms.
ICALL systems, on the other hand, can provide individualized feedback for exercises
and support language awareness, but content assessment based on extensive real-
world knowledge is beyond their scope. But which exercises are appropriate for an
ICALL setting? To begin to address this question, this chapter takes a closer look
at language activities and their prospects for content assessment. Incorporated into
this discussion of language activities are three critical issues:

1. the linguistic complexity permitted or required in responses to an activity,
2. the requirements, abilities and limitations of language technology, and
3. the appropriateness (i.e., authenticity) of activities given current language teach-
ing theories.

Taking these issues into consideration, this chapter examines various exercises
and evaluates how their properties may affect content assessment, defines a spectrum
of language-learning exercises and processing requirements given these properties,
and motivates a set of exercises from the spectrum to be used in testing the content
assessment model described in subsequent chapters.

3.1 Language-learning Exercises

Decoo and Colpaert (1999) note that much of the current practice-focused content of
ICALL systems (i.e., the exercises) comes from existing language-learning materials.
In other words, exercises are largely not designed specifically with ICALL in mind. It
might be argued that this practice potentially overlooks the strengths of the medium.
If exercises only mimic traditional print exercises, then strengths of computer-based instruction, such as hypertext links, multimedia presentations and instant feedback, are ignored, and there is potentially less value added in shifting exercises to a computer. However, intelligently adapting print-based exercises that would be used in a course anyway to computer-based activities provides a more seamless way for instructors to incorporate ICALL into their existing courses. The catch is in selecting exercises that can be effectively processed in ICALL systems so that learners can benefit.

More importantly, there is evidence that shows learning advantages to offering language exercises via an ICALL system, if that system provides intelligent processing and feedback to students on exercises. For instance, Nagata (1996) showed that feedback from a CALL program could be more effective than simple paper-based worksheet feedback. She compared Nihongo-CALI, an ICALL system for practicing the appropriate use of particles in Japanese, with paper-based workbook instruction for the same material. Students used either the system or workbook instruction to practice on their own. Grammatical instruction and the exercises were the same across test groups. However, Nihongo-CALI provided specific, detailed feedback on errors as learners completed exercises. The students in the workbook group received an answer sheet for the exercises to check their own work, but no detailed feedback targeting their individual errors.

Nagata found that performance on comprehension exercises did not differ for the two test groups, but performance on production exercises did. In fact, using the ICALL system with feedback significantly improved student performance. In her discussion, Nagata pointed out that, clearly, it is not being on the computer that matters, but the immediacy and quality of the feedback. This point has been noted by other CALL researchers (Pederson, 1987; Teichert, 1985), and it is worth
emphasizing in the current discussion: the presence of CALL or ICALL technology does not automatically improve performance. It is how the technology is used that matters.

To that end, it is useful to consider what types of existing language-learning exercises can be successfully adapted to an ICALL setting by analyzing exercise properties and the potential for content assessment. Successful adaptation of an exercise refers to the ability of an ICALL system to reliably and accurately process and provide feedback to learner responses to a particular activity. The position adopted here is that the variation expected in learner responses has a crucial impact on the ability of an ICALL system to provide effective content assessment.

In order to develop strategies for dealing with different degrees of variation in learner responses, it is useful to explore the relationship between exercise properties, possible response variation, and processing needs for content assessment. Different language-learning exercises carry different expectations with respect to the level and type of linguistic variation possible across learner responses. To develop adequate processing strategies for content assessment, it is important to understand the connection between exercises and expected variation because the level of variation imposes requirements and limitations on different processing strategies.

3.1.1 The Inadequacy of Categorizing by Common Exercise Types

It is worth mentioning here that in order to explore the relationship between exercise properties and content assessment, it is not enough to categorize exercises by traditional exercise types such as cloze (i.e., fill-in-the-blank), short answer or essay. To illustrate this, consider the three cloze exercises in (11).

---

1Exercise (11a) is taken from Basic English Grammar (Azar, 1996). (11b) and (11c) are from Laubach Way to English: Workbook for Skill Book 3 (Macero, 1991). Both are exercise workbooks for learners of English.
(11) a. **Direction Line:** Fill in the blanks to form questions with the forms given in parentheses.²

   **Cue:** (Anita, go) __________ to her uncle’s house every day?

   **Target Response:** Does Anita go to her uncle’s house every day?

b. **Direction Line:** Write of, for, back, to or at.

   **Cue:** I look _______ the bright lights again.

   **Target Response:** I look for/to/at the bright lights again.

c. **Direction Line:** Write the missing words.

   **Cue: A Night Flight from China.** My name is Lee Chan. I am on a night flight. I’m high in the sky. ______ night is dark. But the ______ lights of my city are ______ in sight. The bright lights ______ China are still in sight. (…)

   **Target Response:** My name is Lee Chan. I am on a night flight. I’m high in the sky. The night is dark. But the bright lights of my city are still in sight. The bright lights of China are still in sight. (…)

   To correctly answer (11a) — a common form of fill-in-the-blank accompanied by little or no contextualization — the learner needs to know how to construct questions and how to form verbs in simple present tense. While the exercise is fill-in-the-blank, the subject and base form of the main verb are given. So, there is very little opportunity for variation in a learner response. For (11b), learners are provided a list of base forms and sentences. For each sentence, there is a blank, and the learner must select the appropriate lemma and fill the blank with its correct form. For this exercise, either for, to or at might be acceptable and no change in form is required. But typically, in a complete version of such an exercise, there is only one best answer for each blank and, as with (11a), some morphological manipulation is required. In

²In the full exercise, a sample was given as well as part of the instructions as well.
contrast, (11c) is a less-restricted cloze exercise. The context is intended to guide the learner into answering in a particular way, but alternative answers might be appropriate. For example, the second blank might be filled by *bright, tiny, distant, etc.* All are appropriate in the context, and there is no clear way to rank one choice as best.

In terms of ICALL processing, these three forms of cloze have different requirements. (11a) has only one possible target response; specifying the target in an activity model for the exercise and matching the learner response against the target with string matching would be effective. This would also work for (11b). In the event more than one answer is possible in (11b), the acceptable alternatives could simply be listed in the activity model or matched with regular expressions. But processing learner responses for (11c) is not so simple. Different blanks within the activity have the potential for different degrees of variability. For The bright lights _______ China are still in sight, the word *of* is the most likely target given its use in a parallel structure of the previous sentence of the exercise, but *in* could be acceptable as well. So, there is a little variation possible, but it is both predictable and enumerable.

In contrast, is it possible to predict all the acceptable target concepts for the _______ lights of my city? Again parallel structure might indicate *bright* as the target, but many other modifiers may be equally acceptable. To take things one step further, example (12) lists modifiers that would and would not be acceptable.

(12) a. the bright lights
   b. the distant lights
   c. the comforting lights
   d. ? the ugly lights
   e. * the extinguished lights
   f. * the noisy lights

^3Unacceptable responses are marked with a ‘*’. Questionable responses are marked with a ‘?’. 55
Just specifying that a modifier must fill the slot does not fully capture the knowledge required to appropriately answer the question, although it certainly helps. This suggests that knowledge about the types of modifiers that can describe lights and which are appropriate in the given context is necessary for fully flexible processing.

Properties of exercises, linguistic processing and required knowledge will be discussed further in subsequent sections. The point here is simply that it is not enough to identify the type of exercise, such as cloze, to be able to determine the processing required for content assessment. Many different exercises fall under the heading ‘cloze’. Not all cloze exercises can be processed by listing the target response(s) and then matching against the learner response. The same is true of other traditional categories of exercises. This suggests that other features should be considered in determining what kind of processing is suitable or required for particular exercises. Furthermore, the suitability of the exercise itself from a pedagogical perspective is yet another issue not necessarily tied to the label given to an activity.

Before moving on to look at other exercises, it is worth noting two other points about the expected variation of learner responses using these examples. First, there is a trade-off between the amount of information or instruction supplied for an exercise and the expected variation of responses. For instance, in (11b), the learner is instructed to use only the prepositions in the given list. This limits the expected variation of responses to one of those five words. Any other word is automatically wrong, given the instructions. This can be used to the advantage of an ICALL system because clear and precise instructions about the kinds of target responses that are (or are not) acceptable can reduce the range of variation an ICALL content assessment module must deal with. The more explicit instruction, the less expected variation. Clear instruction also goes hand-in-hand with good pedagogical practice.
It is convenient that exercises with clear directions, examples and practice modules not only reflect good language teaching pedagogy, but also facilitate processing.

Second, there is also a trade-off between expected variation and the type of processing and level of analysis required. For the first two example cloze exercises, string matching would suffice. For the third, string matching might work for some of the target slots. But for those with more expected variation, additional processing and analysis would be required to identify parts of speech or semantic restrictions that limit acceptable responses.

3.1.2 A Sample of Language Learning Exercises

To introduce and illustrate various properties of exercises relevant to response variation, several exercise examples for learners of English are described below. These are drawn from textbooks or workbooks for English language learners. Included with each is a short description of the kinds of knowledge the learner must possess to correctly respond to the exercise and a mention of how much variation might be expected in response to each activity. This sample is obviously not intended to cover the full range of activities possible in ICALL or in language instruction as a whole. Rather, these examples reflect common activities with properties that highlight the issues relevant to a discussion of processing for content assessment.

(13) (Guided) Superlative Formation

**Direction Line:** Make sentences using *one of + superlative + plural noun*

**Cue:** a big city in Canada

**Target Response:** Montreal is one of the biggest cities in Canada.

(13), (14) and (20) are from Azar (1996). (19) is from Laubach et al. (1991), and the remaining examples are from Azar (2003).
Phrase or sentence formation exercises often focus on particular grammatical constructions. This example provides practice with forming comparisons. The student needs morphological knowledge on how to form plurals and regular and irregular superlatives. Also, learners must have some world knowledge to be able identify a city in Canada larger than most other cities in Canada. Thus, while the exercise is intended to drill word forms, there is some possible content variation in choices for completion. The form variation is limited by the direction line and a sample cue/response pair (not shown).

(14) **Vocabulary Identification**

**Direction Line:** Try to make a list of everything you see in the picture by completing the sentence “I see ...”. Try to use numbers or other units of measure. Use a for singular count nouns.

![Image of a table with various items]

**(Possible) Target Response:** I see three spoons, a box of candy, a fly, ...

Vocabulary identification exercises tend to have very little opportunity for content variation because learners often know only one word for a particular concept, especially for technical vocabulary or before the learner reaches advanced levels. So, their lexical choices often cannot vary. This vocabulary exercise, which was designed for a grammar textbook, evaluates lexical knowledge, knowledge of how to form plurals, and knowledge of how to refer to objects with different units of measurement.
The last of these increases the opportunity for expected variation because single- or multi-token words are possible in expressing measurement in learner responses. For example, the picture contains two packs of matches. This might be identified as *two packs of matches, matches, two books of matches, 10 matches, several matches, etc.* This activity also illustrates that language activities rarely test, or provide practice for, only a single linguistic structure or language skill.

(15) **Gerund or Infinitive Formation**

**Direction Line:** Discuss what you like and don’t like to do. Use the given ideas to make sentences that begin with words from this list.

**List:** *I like, I love, I enjoy, I don’t like, I hate, I can’t stand, I don’t mind*

Sample: cook

Sample Target: I like to cook | I like cooking | I hate cooking, etc.

**Cue:** fly

**Target Response:** I like to fly | I like flying | I hate flying, etc.

In this example, learners combine one of *I love, I enjoy, etc.* with either a gerund or infinitive form of a verb to construct a sentence. Minimally, the learner needs morphological knowledge to be able to answer cues in the exercise. No lexical knowledge and little syntactic knowledge are required because a sample sentence is given as a model. Students can mimic the form of the model without fully understanding it. The purpose of the exercise is to drill gerund and infinitive formation. On the surface, it seems that little variation should be expected since all parts of the sentence are supplied. But there is nothing preventing a learner from responding with *I like flying in hot air balloons.* This might be a perfectly acceptable response; it simply contains more information than is strictly required by the exercise.5

5Unless the requirements of the exercise need to be strictly enforced for a particular teaching purpose, it might be preferable for a content assessment module in ICALL to be able to mark such answers as correct despite the variation in meaning.
At the same time, it is important to recognize that just because responses can vary does not mean they will. It is an open question as to how much (and under what conditions) learner responses will vary, even for activities where variation is expected and allowed. More will be said about this in Chapter 5.

(16) By + Gerund Formation

a. **Direction Line:** Complete the following by using by + a gerund. Use the words in the list.

   List: eat, drink, guess, smile, stay, take, wag, wash, watch, write

   Sample: Students practice written English by writing compositions.

   **Cue:** We clean our clothes __________ them in soap and water.

   **Target Response:** We clean our clothes by washing them in soap and water.

b. **Direction Line:** Complete the following by using by + a gerund. Use your own words.

   **Cue:** You can cook an egg __________ it, __________ it or __________ it.

   (Possible) **Target Response:** You can cook an egg by boiling it, by scrambling it or by frying it.

This cloze activity drills gerund formation as (15) does. But in this case, the base form is either selected from a list as in (15a) or pulled from the learner’s own mind, as in (15b). For (15a), only morphological information (and knowledge of the meaning of the verbs) is required because base forms are given. There is exactly one right answer, which can be assessed with a string match.

(15b) is more interesting, at least in terms of processing. A learner might fill in the blanks with different terms in different orders. The target responses are gerund
forms of three verbs representing ways to cook eggs. Knowledge about which verbs are appropriate comes from world knowledge about the cooking of eggs. While there are presumably only a finite number of ways to cook an egg – boil, poach, fry, scramble, etc., it may be difficult to predict all the potential ways to respond correctly.

(17) **Vocabulary Definitions**

**Direction Line:** Provide definitions for the words listed below. Consult your dictionary if necessary.

**Cue:** A telephone directory is a book . . .

**Target Response:** A telephone directory is a book that lists phone numbers.

As a vocabulary drill, (17) asks learners to construct sentences defining the given terms. Clearly, this requires lexical knowledge about the meanings of terms, but it also requires morphological and syntactic knowledge to construct appropriate endings for the sentence completion aspect of the activity. In terms of content assessment, there are likely key concepts that must be present in the learner response to be considered compatible with the target response. But there is potential for a lot of response variation in the exercise. For example, the learner might reasonably respond to (17) with any of the answers listed in (18). Common to these variations is the idea that a directory contains (minimally) telephone numbers, but listing all possible variations is not feasible.

(18) A telephone directory is a book

. . . that has telephone numbers in it

. . . that contains people’s addresses and telephone numbers

. . . for finding telephone numbers
Reading Comprehension

**Direction Line:** Finish each sentence. (Learners read a roughly 500-word story titled “The Neighbor’s Dog”.)

**Cue:** Jerry Dawson was angry with Bob Shaw because …

**Target Response:** … because the Shaws’ dog dug a hole in his lawn.

Many comprehension activities test a learner’s understanding of the target language by presenting short passages and asking him to answer questions about the passages. Associated activities may take the form of multiple-choice questions, fill-in-the-blank, short answer or essay. Target responses may be single words, phrases, sentences or paragraphs. In (19), the exercise is a sentence completion activity in which the target responses are facts extracted from the passage. But it could be easily reworded as a question-answering task.6

To complete (19), the learner needs morphological, lexical, syntactic and semantic knowledge. He needs to able to understand the logical connections between components of the story and make possible inferences regarding what events caused others.7

As another example, description activities, even those guided by pictures as in (20) on the next page, have the potential for a great deal of expected structural variation. In addition to knowing how to form present tense, the learner needs lexical, syntactic and semantic knowledge in order to be able to construct a response. Moreover, content variation is also possible. Any number of equally plausible descriptions might be given for the same image. So, there may not be any single ideal target response.

---

6For instance, (19) could be *Why was Jerry Dawson angry with Bob Shaw?*

7At the end of this particular story, the dog barks in the middle of the night and someone is seen running away from the two houses. The learner must be able to infer that the person was likely a burglar and the dog kept the neighbors safe.
(20) **Picture Description**

**Direction Line:** The name of the person in the pictures is Alex. What is he doing? Why? Make up probable reasons.

**Cue:** Assume the pictures show things that Alex is doing right now and/or does every day. Use the pictures to describe some of Alex’s activities, using present tenses.

![Picture](image.png)

**(Possible) Target Response:** Alex is dining with his wife at a restaurant.

(21) **Short Composition**

**Direction Line:** Write a composition based on one of the following topics.

**Cue:** Compare and contrast the seasons of the year.

**Target Response:** (Variable)

As with (19) and (20), the learner needs morphological, lexical, syntactic and semantic linguistic knowledge to be able to answer (21). World knowledge about seasons in different parts of the country or different parts of the world may also be required in order to answer appropriately. The structure of the response must include how the seasons are similar or different in at least one aspect in order to meet the ‘compare and contrast’ requirement of the activity. Otherwise, the instructions are
not specific enough to restrict the expected content. There is no specific expected
target response. The potential content of correct answers, as well as structural,
morphological, and lexical choices, may be highly variable. While such activities can
be highly engaging for students, from a processing perspective the activity involves
essentially unrestricted lexical and structural input and requires extensive linguistic
and world knowledge to evaluate whether a learner response is appropriate. As a
result, such activities are beyond the scope of current modeling and algorithms in
NLP.

3.1.3 Characterization of Exercise Properties

The exercises in 3.1.2 illustrate several properties potentially relevant to response
variation and effective content assessment, independent of whether the exercise is
appropriate from a pedagogical perspective. These properties include

1. the nature of information provided in the activity model,
2. the linguistic knowledge and cognitive skills required of the learner,
3. the corresponding availability of “gold standard” target responses, and
4. the criteria under which the exercise is intended to be evaluated.

Each of these properties will be discussed in more depth, beginning with what
information is provided in the activity model for the exercise.

3.1.3.1 Explicit Knowledge in the Activity Model

The activity model specifies the nature of the instruction given about the intended
form and/or meaning of the expected response. Within the bounds of the require-
ments imposed by the activity model, an activity can support more or less variation

\footnote{A discussion of selecting pedagogically appropriate activities begins in Section 3.1.3.3.}
in the length, the lexical material, the forms and constructions used, or the content expressed. For example, consider the instructions in (22). The direction line in (22a) is from the activity in (21); (22b) is a hypothetical alternative.

(22) a. Directions. Write a short composition comparing and contrasting the seasons of the year.

b. Directions. Write four sentences comparing and contrasting the seasons of the year. Explain how the seasons are alike or differ with respect to the weather, using terms such as hot, cold, rainy, windy, snow, sun, etc.

The instructions in (22b) provide explicit length requirements, suggested linguistic material to use, and guidelines for what subtopics (i.e., the weather) learners should target in their compositions. Through such requirements, the activity setup explicitly controls the range of variation permitted in learner responses. The essays produced by learners assigned (22b) are likely to show less content variation than responses from learners assigned exercise (22a), but responses to (22b) may still vary greatly with respect to linguistic form. The more types of variation are possible, the more processing may be required because variation at one level may trigger variation at other levels. For instance, a change in the sentence structure may entail changes of morphological endings.

Thus, the level of information provided with the exercise clearly influences the degree of knowledge required for the learner to respond to an exercise, as well as the degree of knowledge the ICALL system must possess to perform accurate content assessment. As suggested with the example above, types of explicit knowledge provided to the learner include whether the activity is accompanied by a sample exercise, indicates words or forms for the learner to use, or makes other format/structure notes such as a specification of length. Inclusion of such information guides the learner to respond in a particular way, limiting the variation to be expected in responses.
For example, (15) and (16) are instances where sample exercises are provided to the learner. In (15), for the prompt *cook*, the sample response given is *I like to cook*, *I like cooking*, *I hate cooking*, *I don't mind cooking*, *etc.* While the activity itself is not communicative, providing samples is considered good pedagogical practice because it gives the learner a feel for what the directions are asking him to do and provides a range of acceptable example answers. It also shows the learner that no additional information needs to be added to the sentences he creates from prompts. If he follows the model precisely, the learner needs only to be able to recognize the verb in the prompt and convert it to a gerund or infinitive to answer correctly.

In (16), the exercise is split into two parts – one for which the learner selects base forms from a list and one for which the learner thinks of his own verbs to use. Again, the sample exercise in (16a) provides a model using one of the verbs from the list and shows how to make a *by + gerund* phrase from it. But the sample applies to both parts of the exercise and gives the learner a clue about the appropriate form.

In addition to a sample exercise, (16) also gives explicit information on the words and forms required. The learner is instructed to practice the *by + gerund* construction. (13) also provides this kind of explicit information, without providing a sample exercise. In that case, the activity is to build a sentence using the given content and the *one of + superlative + plural noun* construction.

Other forms of explicit structural information of format guidelines might include the length of a required essay, whether to build a sentence or phrase, how many concepts to include (ex: *List 4 qualities you admire in a person.*), etc. Taking a closer look at response length, an activity can ask the learner to input a single character (as in multiple choice), a single word, a phrase, a sentence, a paragraph, or an essay. The

---

9In the context of classroom instruction, the intended pattern might be entirely clear without explicit examples associated with the exercise.
longer the learner response, the more processing is required to evaluate that material. On the other hand, an increase in response length alone does not necessarily correspond to an increase in processing difficulty given that longer responses also provide more evidence, which can facilitate content assessment. Arguably, the complexity of content processing increases whenever the increase in response length corresponds to an increase in the range of expected variation in the response.

In assessing learner responses, an instructor may penalize learners for not following directions explicitly specified or may be lenient and allow variation from the expected target response. An ICALL system may be equally flexible. Which instructions are to be loosely enforced (and how loosely) could be encoded as part of the target response as part of the expected variation. But enforcing a strict adherence to explicit instructions facilitates processing.

3.1.3.2 The Nature and Availability of Target Responses

A “gold standard” target response refers to the specific correct answer (or set of answers) that is explicitly and completely defined for a given activity. For many activities, there is no precise set of correct answers; for other activities, target answers are readily available. The nature and availability of target responses is intertwined with the level of expected variation in learner responses because the nature of target responses goes hand in hand with the required linguistic or world knowledge and the potential for content variation in learner responses.

Obviously, different exercises target different types and levels of linguistic knowledge. The four broad types of linguistic knowledge distinguished here are morphological, lexical, syntactic and semantic.\textsuperscript{10}

\textsuperscript{10}The focus here is on text-based exercises, but phonetic knowledge would also need to be included for speech-targeted ICALL systems. And pragmatic knowledge would be an appropriate category for discourse-based activities.
Morphological knowledge includes knowledge of word forms, inflection, parts of speech, etc. Syntactic knowledge is knowledge of structural grammatical form. Lexical and semantic knowledge cover aspects of meaning. Lexical knowledge encompasses word meaning, while semantic knowledge covers “compositional” meaning – how individual words are combined into larger meaningful units within and across sentences. The latter would include knowledge of selection restrictions (e.g., the object argument of the verb to drink must combine with something drinkable) and how to organize concepts logically.

How the examples in Section 3.1.2 vary with respect to required linguistic knowledge is included above with the description of each exercise. However, one important observation about the example exercises is that many of them test multiple aspects of linguistic knowledge at the same time. For instance, the activity in (17) on vocabulary definitions provides practice with lexical knowledge (learners must define words), as well as syntactic knowledge (learners must provide grammatical completions for sentences).

In addition to linguistic knowledge, exercises also vary in the extent they assume world knowledge, which is an umbrella term for facts and propositions describing the relationships between concepts. However, even within a given activity, the level of world knowledge assumed can vary. For instance, in the example item in (13), learners must know the name of at least one city in Canada to be able to complete the activity at all. To complete it correctly, the learner also needs to know which cities in Canada are large (in terms of size or population). This is geographic world knowledge about a particular place. But other exercises within the same activity (not included in (13)) did not require any world knowledge.

This points to the first issue concerning recall of required world knowledge: it is not necessarily predictable. Within a given activity, one cue may require world
knowledge, while another does not. And, as (13) suggests, problematic activities are not just those that allow unrestricted-input, such as the essay on seasons in (21). Even an exercise that has very restricted input, like (13), can require access to world knowledge.

The second issue relates to the processing of and access to world knowledge. For (13), the information might be provided in the form of a list of city names, sizes and locations. But other types of world knowledge are less straightforward to provide in an open-domain ICALL system.\textsuperscript{11}

In addition to linguistic variation in the learner response, the semantic content of a learner response may sometimes deviate from that of the target response. For a given exercise, there may be multiple ways to respond to a question correctly. Any time an exercise allows for multiple realizations of the same semantic content, there is potential for variation in the learner response. But there may also be some legitimate variation in the semantic content as well. The type and degree of variation that may be expected depends on the exercise.

Types of expected variation include lexical choice, syntactic structure, word forms, sentence meaning, etc. Basically, variation can surface for one or more of any of the types of linguistic or world knowledge previously described. For instance, in (19), the target response must contain the fact the dog in the source story dug a hole in a lawn. But there are many ways to formulate this, including those listed in (23).

\begin{itemize}
\item the Shaws’ dog dug a hole in his lawn
\item his dog dug a hole in Jerry’s lawn
\item the dog tore up the lawn by digging a hole
\item the dog used his claws to dig a big hole in his beautiful lawn
\end{itemize}

\textsuperscript{11}This lack of access to world knowledge in ICALL system puts many, but not all, activities relating to the culture of a target language beyond the scope of what can be processed in ICALL because the more open-ended the culture-related activity, the more world knowledge is required.
These hypothetical responses vary in lexical choice, syntactic structure, and how much information is conveyed, even though all convey the required target concepts.

For specifying target responses and processing variation, one aspect to consider is whether or not the expected variation is predictable. A second, related aspect is whether the variation is enumerable. In (14), the learner looks at a picture and lists the things he sees. Variability is possible in the number of items listed, the word choice for referring to an item and the unit of measurement selected for items that appear more than once. That is a lot of expected variation, but it is predictable. That is, the range of variation is known. This is different from whether or not the variation is enumerable (i.e., whether all possible variations can be listed in the target response). For the activity in (13), in which the only source of expected variation is in the choice of the city in Canada, it is predictable and possible to list all the cities. In contrast, for the task in (19), although the variation is predictable, it is unlikely that all possible ways to answer each exercise can be reliably enumerated. And as a final example, in (21), learners are asked to write a composition comparing the seasons. It is not possible to predict or list all variants of legitimate responses to such open-ended activities.

It should be clear that the instructions provided to the learner have a crucial impact on the degree of expected variation. If the learner is given a list of words to choose from and a syntactic structure to use, then the expected variation is confined to the words on the list in the form provided. If the learner is given forms, but not words, the variation increases. Without a list of words, the potential for variation is greater still. The potential for content variation also increases as expected answers allow less restricted input.
3.1.3.3 External Criteria for Activity Evaluation

Every language activity has external criteria that specify the intended goals of assessment for the particular activity. For instance, an activity can be created to assess a learner’s ability to use the grammatically correct forms and word order in targeted constructions. Or activities can be designed to assess a learner’s ability to use language for a particular task, e.g., to understand questions and identify the relevant information in a reading comprehension activity. Naturally, there is a range of assessment options between these two possibilities. The type of assessment performed is often correlated with the type of learner response an activity supports. Activities tightly controlling the form of learner responses, e.g., in a fill-in-the-blank or build-a-sentence exercise, will typically be used for form assessment. For learner responses less restricted in form, such as essays, the focus typically is on content assessment. However, this is only a typical correlation - the activity type does not determine the assessment criteria. For example, a learner essay can be assessed exclusively in terms of its form, such as the correct choice of tenses, and form errors in a fill-in-the-blank activity could be ignored as long as a meaningful lexical choice was made by the student. The assessment criteria thus do not simply fall out of the activity design but need to be explicitly stated. The particular pedagogical goals behind an activity, which may change depending how or why an instructor chooses to modify and use existing exercises, directly impact the type and degree of analysis required.

To elaborate further, consider that any language exercise presumably reflects at least one learning goal. How these learning goals are defined may vary from instructor to instructor, theory to theory, etc. However, the classic and influential classification of learning goals is the Bloom taxonomy (Bloom, 1956), which identifies six categories of cognitive skills involved in the learning process. Activities fall into
one or more of these categories, which include Knowledge, Comprehension, Application, Analysis, Synthesis, and Evaluation.\textsuperscript{12} Under this approach, learning goals are hierarchical, with attaining knowledge as the most basic goal and attaining evaluation skills as the most complex of the goals in the hierarchy.

Exercises (13)-(17) all reflect the Knowledge learning goal. Each involves recognition and/or recall of language knowledge (e.g., vocabulary, grammatical forms, etc.). Exercise (19) requires Comprehension skills to finish sentences about a reading passage, and the exercise in (20) requires Application skills to use language knowledge to create a description about the given image. The most complex of the exercises is (21), which requires Analysis skills to compare and contrast the seasons.

More will be said about using cognitive and knowledge skills to classify language exercises in Chapter 7. However, it is reasonable to assume that the more complex (i.e., higher in the hierarchy) the learning goal for an exercise is, the more complex a response to that exercise will be. And the harder it will be to automatically process and provide feedback to that response.

One final, crucial point to discuss here with respect to external criteria is how current teaching practices impact response variation through exercise design. Since the 1970s, language teaching has leaned toward a more communicative focus, which in its most extreme version assumes students will acquire a language simply through being exposed to, and required to communicate in, that language (Howatt, 1984). Under that model, explicit instruction on grammar or structure is avoided or eliminated. A more recent approach to language teaching that evolved from this communicative focus is task-based instruction (Nunan, 1989), which organizes language teaching around natural tasks such as reading newspapers, writing and editing

\textsuperscript{12}Bloom’s taxonomy includes other, non-cognitive dimensions of learning, as well. These are not discussed here.
essays and other activities that emphasize meaningful interaction in the target language. The basic tenets of task-based instruction are that 1) learners do not master language one structure or set of terms at a time, 2) learners need to focus on meaning through exposure to comprehensible language input from different contexts, and 3) learners need to practice producing language (Willis, 2004). Under this approach, the more closely an exercise corresponds to a real-life task, the better. Because real-life language tasks often only loosely restrict the form or content of language, the potential variation in responses to activities design to be task-based is very high. And some task-based activities allow too much variation to be feasible in an ICALL setting. However, there are many ways to implement task-based instruction to work in a practical teaching environment that also includes an ICALL component.

Nerbonne (2003) argues that ICALL systems are probably best served by remaining as theory-neutral as possible because theory is not always straightforward to implement and such ICALL systems provide a flexible format that can adapt to developments in actual teaching practices. However, this does not mean that ICALL systems cannot be evaluated by how well they accommodate current language teaching methods. The example exercises selected for discussion here are similar to exercises found in a range of language learning texts from within the last decade for English, French and Spanish (Muyskens et al., 2002; Pérez-Gironés and Elliott, 2005; Long and Macián, 2005; Oxenden et al., 1997). The range of texts suggests that the exercises do reflect the types of activities used in actual teaching situations.

But not all of the activities are task-based. For instance, the activities in (13), (15) and (16) are drills. These are in contrast to the examples in (19) and (21), which are reading-comprehension and essay-writing task-based activities, respectively. Incidentally, the least “communicative” exercises (i.e., the drills) are those that have the least potential for legitimate variation, making them simpler to process in an ICALL
system. This is not surprising since, as mentioned, potential variation increases with the freedom learners have in constructing responses, and task-based/communicative activities are more likely to allow loosely restricted responses.

Of course, communicative activities that require open, unrestricted dialogue between the learner and ICALL system are beyond the reach of current technology. These activities are still best left to the classroom and human interaction. But that leaves a wide range of task-based activities, and it still remains to be determined which of these can be effectively processed in ICALL systems.

3.2 The Spectrum of Language-learning Exercises

The discussion to this point suggests that the more variation possible in learner responses to a language exercise, the more processing is often required for content assessment. A spectrum of language-learning exercises falls out of this relationship between expected response variation and natural language processing (NLP). Figure 3.1 depicts this spectrum.

Figure 3.1: The Language Exercise Spectrum
As with many spectrums, the continuum of language exercises can be described by first focusing on the spectrum extremes. At one extreme in Figure 3.1, there are exercises with tightly restricted responses (i.e., limited variation allowed) that requiring minimal analysis in order to assess meaning. Learner responses to activities on this end of the spectrum tend to converge on a single target response. For example, (16a) in Section 3.1.2 requires very little analysis to assess content. Because the activity is a cloze drill with a single correct answer, content assessment can be purely form-based matching.

In fact, for the activities with predictable and enumerable responses, string matching or regular expression matching would often be effective form-analysis strategies for content assessment. To illustrate this, consider the example in (15), in which learners are given a verb phrase in base form (ex: *cook*) with a sentence opener (e.g., *I like*) and instructed to form a sentence using gerund or infinitive forms of the base form verb in the supplied verb phrase (ex: *I like cooking* or *I like to cook*). The content of the activity is fully specified. All learners need to do is convert the provided base form verb into gerund or infinitive form and combine it with the sentence opener of their choosing.

Processing such activities with string matching or regular expression matching is entirely appropriate. For string matching, the target response(s) can simply be listed in the activity model of the exercise. For each part of the exercise in (15), there are two verb forms and seven sentence openers to choose from. Thus, there are 14 possible target responses for each part of the exercise. These can be stored in the activity model and matched against the learner response.

Regular expression matching is also appropriate as a simple processing strategy for activities at the minimal-analysis end of the spectrum. In (15) for example, the sentence openers are canned text. Given that this repeats for all parts of the activity,
it is straightforward to store a regular expression for the target response, rather than all 14 combinations. Thus, the expression might be as in (24), where <Gerund> and <Infinitive> are the variables for the alternate verb forms specific to each part of the activity.

(24) /(I like|I love|I enjoy|I don’t like|I hate|I can’t stand|I don’t mind)
    (<Gerund>|<Infinitive>)/

Such a pattern-matching strategy allows compact representation of the target responses, while still capturing all possible targets. Content assessment in this case would involve searching the learner response for a match to the pattern specified as the target response.

With activities such as (15), processing with string or regular expression matching is simple, but the activity itself is not particularly meaningful from a communicative standpoint. And as the learner is required to provide more information or is given more freedom in response to more meaningful activities, these strategies will break down. For example, a learner response to the vocabulary identification activity in (14) may list items in a different order using different lexical items than those in the target response. Subsequently, representing the target response as an ordered list would certainly not be ideal.

Turning toward the other extreme of the exercise spectrum in Figure 3.1, there are loosely restricted exercises requiring extensive form and content analysis to assess meaning due to the wide range of variation allowed in learner responses. The short composition activity in (21) is one example that falls on this end of the spectrum. Acceptable answers to such activities tend to diverge from one another, making precise specification of a target response difficult or impossible.
It has been argued here that predictability is an important factor in determining whether the content of an exercise can be processed in ICALL systems. This includes both whether there is a target response for the activity and whether it is possible to anticipate how the learner response will vary from that target. The more unpredictable, the harder ICALL content assessment becomes. (21) is a good illustration of the unpredictability in learner responses. To evaluate the content of a response, an ICALL content assessment module needs to have some idea of what the target response is. But in an essay comparing seasons, what is the target? Consider the two hypothetical responses in (25).

(25) a. There are four seasons in a year. Winter is cold, and summer is hot. It rains in spring and gets warm. Leaves grow on trees. In autumn, it is cooler and the leaves fall.

b. There are four seasons in a year. Summer and winter have long holidays from school. Fall is the beginning of the school year, and spring is the end.

Between (25a) and (25b), which is the better answer? (25a) perhaps conforms to a “typical” response to such a question, but (25b) meets the criteria of the activity just as well. While an ICALL system can certainly evaluate whether there are grammatical errors in the responses to provide guidance for revision writing, content errors are difficult to identify when the target response cannot be well defined. Exercises on this extreme of the spectrum lack such well-defined targets. However, human instructors can easily provide content or grammatical feedback.
Moreover, any open-ended activities that require the learner to provide subjective or person-specific responses, such as in (26), are on the end of the exercise spectrum that ICALL content assessment cannot handle.\textsuperscript{13}

(26) \textbf{Cue:} Would you like to live in the woods? Why or why not?

To respond to (26), the learner needs morphological, syntactic and semantic knowledge to write a short essay justifying his opinion. But it is impossible to specify any absolute target response; any number of answers would be suitable. No two learner responses will be exactly alike.\textsuperscript{14} Presumably, ICALL assessment of such questions would be in terms of grammar and possibly rhetorical structure, rather than the semantic content of the learner response.

3.2.1 Placing Exercises on the Spectrum

While it is clear that some exercises fall at particular extremes of the spectrum, it is important to emphasize that assigning any exercise an absolute position on the spectrum based on characteristics discussed in Section 3.1.3 is not straightforward. To illustrate this, several features discussed in Section 3.1.3 and how these features relate to the exercises in 3.1.2 are summarized in Table 3.1.

\textsuperscript{13}Example from Laubach et al. (1991), a workbook for learners of English.

\textsuperscript{14}Cheating notwithstanding.
<table>
<thead>
<tr>
<th>Exercise</th>
<th>Explicit Instruction</th>
<th>Linguistic Knowledge</th>
<th>Type of Variation</th>
<th>Degree of Variation</th>
<th>Cognitive Level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sample exercise</td>
<td>Other format/structure notes</td>
<td>Morphological</td>
<td>Syntactic</td>
<td>Semantic</td>
</tr>
<tr>
<td>Superlative Formation</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Vocabulary Identification</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Gerund/Infinitive Formation</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>By+Gerund Formation</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Vocabulary Definitions</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Reading Comprehension</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Picture Description</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Short Composition</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 3.1: Example Exercises and their Properties
Looking at the summary of linguistic knowledge in Table 3.1, the last four activities – from (17), (19), (20), and (21) – have nearly the same features. These activities are the description from a picture, the reading comprehension exercise, short essay composition and sentence completion of dictionary definitions. Because responses are clausal, all four tasks require the full range of linguistic knowledge and can vary at each of the levels. Only one – the picture description task – includes guidance on form in the instructions. The real differences between these exercises are in (i) whether the content of learner responses is predictable and (ii) what the cognitive level of the question is. For the picture description and essay composition, there is no way to predict from the available information how a learner will respond, although there are concepts likely to be in the responses. For instance, for the essay composition, there are numerous ways to describe the seasons. It is possible to define concepts that are likely to be in the composition (weather, temperature, holidays, etc.), but not to specify the exact details. On the other hand, the dictionary definition and reading comprehension tasks have clear target responses. It is the predictability that makes processing “easier” because the system has a clear definition of what to look for in a learner response. But what determines predictability?

To some degree, the level of explicit instruction can determine whether an exercise has predictable responses as illustrated in (22). However, in the absence of explicit constraints to guide the learner to produce a target response, the linguistic knowledge and variation type features in Table 3.1 for the last four activities offer no clues for how to determine the predictability (or process-ability) of an activity.

The progressively higher levels of cognitive skills reflected in Table 3.1 for the last four activities do seem to offer a more clear-cut way to differentiate between them. Each exercise corresponds to a different learning goal. (17), (19), (20), and (21) are Knowledge, Comprehension, Application and Analysis activities, respectively. In this
very small sample, higher cognitive skills do seem connected to the relative difficulty in predicting likely responses and, hence, to the relative difficulty in providing reliable content assessment. This discussion will be continued in Chapter 7.

As for using the given features to categorize activities on a spectrum, it is possible to argue that the first four activities in Table 3.1 should be placed somewhere toward the end of the spectrum that requires limited processing, while the remaining four lean toward the loosely restricted side of the spectrum. However, more exact categorization is difficult using these features. For example, the more explicit the instructions, the less expected variation, but not necessarily less freedom in responses. For instance (14) involves identifying vocabulary in a picture. Quite a bit of variation is still possible even though the instructions restrict the input substantially.

Similarly, the more linguistic knowledge required, the more opportunity for variation. But if semantic knowledge is required, then all three of the other types of knowledge (morphological, lexical and syntactic) are likely to be required as well. In fact, any exercise is likely to involve all four levels of linguistic knowledge and subsequent potential for variation if the response for that exercise is a sentence (or sentences) for which the structure of the sentence is not pre-specified (as it was in (13)). Thus, broad features of linguistic knowledge and variation types, while useful for highlighting issues in processing, do not precisely locate exercises on the spectrum.

However, precisely locating exercises relative to one another on the spectrum is not the ultimate goal. The goal is to determine what makes exercises possible or impossible to process reliably for content assessment and which strategies are effective to that end. It is likely a combination of properties that contribute to this determination. Identifying the range of exercises the can be processed for content assessment will take further exploration of exercises not on the extremes of the spectrum, but somewhere in the middle.
3.2.2 Focusing on Exercises in the Middle Ground

To begin to address the degree to which content assessment is possible given existing technology, the present work focuses on exploring a subset of exercises in the middle ground of the exercise spectrum. This space between the opposite ends of the spectrum is an interesting intersection between what is pedagogically interesting and what is computationally feasible to realize in ICALL system. The assumption is that focusing on strategies to effectively process language activities in the middle ground of the spectrum will result in effective processing of a wider range of activities including those on the tightly restricted extreme of the spectrum. This full range of coverage is referred to in Figure 3.1 as the *viable processing ground*.

But the middle ground still covers a very wide range of exercises. To narrow the scope for the present work, two basic selection criteria were taken into account. The first consideration for selecting suitable exercises was that the exercises emphasize meaning – comprehension or production. This emphasis is important from the perspectives of both ICALL system evaluation and second-language instruction.

For the former, it seems obvious that exercises that require the production of meaningful language are crucial to evaluating the effectiveness in processing meaning in a content assessment module. There is simply more opportunity in such activities for variation in learner responses that can test the strengths and weaknesses in ICALL content assessment.

As for second-language instruction, few today would dispute the claim that meaningful interaction in the foreign language is an essential component of second language acquisition. Various theories of language learning reflect this idea: communicative language teaching, content-based instruction and task-based language teaching all stress the importance of meaning and exchange of information in language
learning (Richards and Rodgers, 2001). And research in the field does suggest that meaningful practice does enhance learning. For example, in a study comparing mechanical drills and more meaningful practice, VanPatten and Cadierno (1993) found that learners performed better on comprehension and production tasks if they had meaningful practice; whereas, those students who were given mechanical drills performed well only on the production aspect of language use.

For instance, Lee (2000) defines communication as the “expression, interpretation, and negotiation of meaning” (p. 1). He argues that, for acquisition, output (i.e., the language the learner produces) is as important as input (i.e., language presented to the learner). This suggests that the learner must produce meaningful responses to activities in order to facilitate language acquisition. But meaningful output is difficult to achieve in response to traditional language drills or practice in which the learner can mimic a particular pattern without understanding its meaning. Thus, the less restricted the form of the learner response, the more opportunity for the kind of meaningful production of language emphasized in current teaching practices.

To clarify the term meaningful, the present work borrows the term from Paulston (1971). Paulston establishes a taxonomy of three types of language practice – mechanical, meaningful and communicative. Mechanical exercises have only one correct response and the learner does not have to attach meaning to sentences in such exercises in order to correctly respond. Meaningful practice also has only one correct response, but the learner must attach meaning to the exercise (and their own response) in order to answer correctly. Communicative practice, under Paulston’s taxonomy, also requires meaning attachment but involves activities that may have more than one correct answer, which the instructor does not know in advance.

Using this categorization scheme, the current focus is on activities that constitute meaningful practice. Learners must understand the activity and the language
in order to construct the correct response, but the instructor (or system) knows in advance what the correct answers are. Communicative activities that do not have predefined answers (ex: *Name three qualities you look for in a friend*) cannot be evaluated for correctness.\(^{15}\)

A second criterion for exercise selection was that they allow legitimate variation in learner responses. As argued, variation is tied directly to how restricted the expected answer is. Suitable exercises are loosely restricted. That is, there are potentially multiple ways to express the right answer in a learner response. Expected variation is possible on several linguistic levels, including lexical, morphological, syntactic and possibly semantic levels. Furthermore, answers may be phrasal or clausal.

This requirement is obvious from the standpoint of content assessment evaluation. If there is no variation in learner responses, then evaluating a content assessment module’s ability to reliably process variability in those responses is not possible. But it also makes sense from the perspective of language instruction. Exercises that allow students to creatively produce language are the very exercises that allow variation in responses. This is pedagogically desirable for the reasons outlined above.

Yet even with taking these criteria into account, there are still too many activities that might be explored. For instance, numerous summarization, information gap and reading comprehension activities meet the requirements, just to name a few.\(^{16}\) But for reasons that will be described in the next section, the focus for the remainder of this thesis is on the last of these possibilities — reading comprehension questions.

\(^{15}\)Of course, such activities may be evaluated by a human instructor for semantic consistency and whether the response is reasonable given the question, but such evaluations are not addressed here.

\(^{16}\)The activities described or mentioned up to now, with the exception of information gap activities, are traditionally text-based. This is a deliberate choice. The focus here is on text-based activities because speech-based activities in ICALL require an extra layer of processing to convert speech signals to text that can be processed. Otherwise, the processing requirements for content assessment are the same. To eliminate the extra layer of processing, which does not contribute to the functionality of the content assessment module, speech-based activities are not considered further.
Table 3.2: Example Comprehension Questions from Brown and Hood (2002)

<table>
<thead>
<tr>
<th>Task Label</th>
<th>Question</th>
</tr>
</thead>
</table>
| Scanning                    | • Look back at the text to find the definition of sexual harassment adopted in this reading.  
                               • Look back at the text to find the percentage of men and women who had quit a job because of sexual harassment. |
| Reading for Detail          | • What are three kinds of prisoners for in U.S. prisons?  
                               • What does *probation* mean?  
                               • Why does rehabilitation often fail?  
                               • How has a global youth culture come about?  
                               • What are some factors common to all cults? |
| Understanding Implied Meanings | • Why is burglary a more frequently occurring crime than robbery?  
                                  • Why is Saturday night the most likely time for homicides to occur? |

3.3 A Case Study: Reading Comprehension Questions

Loosely restricted reading comprehension (RC) questions can be used to evaluate both the learner’s understanding of a text and their production of language to provide evidence of that understanding. Several examples of typical RC questions from Brown and Hood (2002), a textbook for intermediate learners of English, are presented in Table 3.2.

RC questions such as these meet the two basic criteria discussed in the previous section. First, because such activities give the learner a lot of freedom in response, high levels of variation are possible. If the answer is explicitly presented in a text, learners may provide responses that are very similar in wording to the source text if the learner has access to the source when responding. Alternatively, responses may express the target content in an entirely different way than the source. Thus, learner
responses are potentially variable on several linguistic levels. And depending on the wording of the question, responses may be single-word, phrasal or clausal, adding to the potential variation.

Second, because they combine comprehension and production tasks, RC questions can provide meaningful interaction in the target language. This is important from a pedagogical perspective because as realistic tasks, RC activities make a good test case for exploring how ICALL systems can provide activities and feedback in line with communicative approaches to language instruction. Thus, while the most communicative, open-ended activities are best left to the classroom setting, RC questions may prove to be the kind of meaningful activity that is suitable for an ICALL setting.

While the potential for linguistic and answer-length variation in learner responses makes loosely restricted RC questions suitable representatives of middle ground activities for evaluating a content assessment module, these qualities are not the only desirable traits. Another argument for their suitability can be made terms of the frequency of their use, both in and out of the second-language instruction domain. RC questions are likely so common because, as Nuttall (1982) notes, they encourage students to think about their answers, they can be used in multiple contexts, and they are relatively easy to develop.

The popularity of RC questions is relevant first because RC questions are unlikely to disappear from the teaching toolkits of language instructors (or instructors in any other domain). Reading comprehension is a task that learners must do in real life, and therefore, instructors often use them in the classroom. Thus, they make good test cases to evaluate the ability of a content assessment module to automatically analyze and assess learner responses to activities that are commonly used in
language instruction. And the fact that they are common across domains is relevant for evaluating the generalizability of a processing approach beyond the ICALL domain.

But RC questions are not without their critics. First, using questions to evaluate reading comprehension has been criticized, in part, because instructors often ask literal, recall questions (Guszak, 1967). By focusing on such RC questions, students are not given opportunity to develop and practice advanced reading skills beyond recall. But as the next section suggests, literal recall RC questions are only one of many types of RC questions possible. And this criticism of recall questions is somewhat dulled by the fact that they do have value and a place in teaching and learning.

Nuttall (1982) argues that another criticism of RC questions is that they have often been used to test reading comprehension skills when emphasis has shifted to teaching, not testing. However, Nuttall herself points out that RC questions can be useful in teaching if they encourage students to think about how language conveys meaning.

A third criticism of loosely restricted RC questions, also pointed out in Nuttall (1982), falls out of the fact that students may comprehend what they read without being about to produce language to convey their comprehension. In other words, the production skills required by loosely restricted RC questions can get in the way when RC questions are used as a tool for evaluating reading skills separate from writing (or listening or speaking) skills. Yet, Krashen (1981) notes that, language teachers often find it impractical to focus on only one of these four skills at a time anyway.

The goal here is not to fully explore and debate the pedagogical usefulness of RC questions. The fact is that all teaching and evaluation methods have strengths and weaknesses. And RC questions are just one of many tools in an instructor’s toolkit. For the current purposes, it is enough that RC questions can be used effectively as
a part of a larger teaching strategy regardless of any pedagogical weaknesses. These weaknesses in RC questions can be overcome by combining them with other tools in the kit and through careful question design that takes into consideration what knowledge such questions are trying to evaluate.

One final note about the RC questions used in the remaining chapters is that the questions follow the format exemplified in Table 3.2. This is important because, as with all language activity types, reading comprehension activities come in many flavors. Comprehension questions may be multiple choice or cloze. Or they may be application questions that have no single target response or no right answer at all. In other words, reading comprehension activities may vary widely in format and complexity. They can reflect the full range of the spectrum, and not all can be reliably processed in a content assessment module.\footnote{This point is an other illustration of the categorization issue discussed with respect to cloze activities in Section 3.1.1.} The selected RC questions were chosen because these activities present an interesting challenge for testing an effective content assessment model, and a model that successfully provides content of assessment of such exercises would extend the capabilities of ICALL systems to more meaningful language activities.

3.4 Summary and Conclusion

In his overview of natural language processing and computer-aided language learning, Nerbonne (2003) points out that ICALL has suffered from the problem of “inflated expectations,” resulting from the belief that an ICALL system should do nothing less than have a natural conversation with students and be able to diagnosis their errors in the process. This kind of full natural language understanding (NLU) is simply beyond current technology. The less restricted the input, the closer an exercise gets
to needing NLU. That is, the closer it gets to the loosely restricted extreme of the spectrum. As such, not all exercises on the spectrum can be processed reliably in an ICALL system. Recall that the perspective taken here is that ICALL technology is not intended as a replacement for human instruction. Rather, the focus of the technology should be on supplementing and supporting human instruction (and only substituting for such instruction when humans are simply not available). Thus, the fact that ICALL systems cannot handle all types of language activities given current technology does not negate the usefulness and relevance of ICALL systems. The role of ICALL assumed here is to provide language activity practice to the learner in independent self-study situations. To be effective in this capacity, ICALL systems need to be able to provide relevant and accurate feedback to the learner on activities, but the complete range of language activities need not be available.

At the same time, where the boundary of reliable content assessment actually lies has not been established. Tschichold (2003) argues that the state of the art in ICALL technology is at a stage in which it is only possible to reliably respond to “low-level” errors of morphology, spelling, (some) syntactic constructions and lexical choice. In other words, she argues ICALL can handle exercises on the limited processing side of the spectrum. However, there is a very large middle ground between the kinds of activities that ICALL systems can definitely handle and the kind of activities ICALL can definitely not handle. And while ICALL is not yet prepared to tackle full NLU, it still remains to be determined what level of content assessment and semantic error diagnosis ICALL systems can provide reliably.

Before turning to developing an approach to content assessment in the next chapters, it might be useful to first recapitulate the major points of discussion in this one. The various example exercises in Section 3.1 were introduced to illustrate how
the different properties of language-learning activities also introduced in that section might impact an ICALL system’s ability to provide effective content assessment. Specifically, the difficulty of processing a response to an activity is influenced by the degree of variation allowed in learner responses. Response variation is, in turn, controlled by i) the amount of instruction provided to the learner about how to respond to an activity, ii) the levels of linguistic and world knowledge required to respond to an activity, iii) the ability to specify a target response (or set of responses) based on the linguistic and world knowledge required, and iv) the external criteria such as the learning goals of an activity.

The expected variation in learner responses and its relationship to processing for content assessment also suggest that activities can be organized along a spectrum, from activities that have loosely restricted responses (and are often trivial to automatically process for content assessment) to those that are simply beyond the capabilities of current NLP technology because they allow unrestricted responses from the learner. One of the overarching goals of this thesis is to explore the middle ground of this spectrum to help establish the boundary of reliable content processing.

However, because there are too many exercises in the middle ground of the language exercise spectrum to investigate in a single thesis, this chapter has argued for using loosely restricted reading comprehension questions as test cases for ICALL content assessment. RC questions are ideal for such a case study. They represent common task-based activities used in current teaching methods, they require meaningful interaction in the target language, and they allow variation on multiple levels of linguistic knowledge. RC questions also enable the definition of an explicit activity model with a finite set of target responses, include a range of question types to test the limits of content assessment, and extend easily to other fields for evaluating the applicability of an processing approach beyond the language-learning context.
This chapter started by saying that effective ICALL starts with carefully selected language-learning exercises. And for the reasons outlined in this chapter, loosely restricted RC questions promise to be effective examples for testing ICALL content assessment technology. Few existing ICALL system offer such activities to language learners, one notable exception being the recently developed TAGARELA system for learners of Portuguese (Amaral, 2007). Using strategies inspired by the current work, TAGARELA incorporates simple content assessment for evaluating learner responses. However, much work on content assessment remains. Thus, an ICALL module that can process some subset of responses to RC questions pushes the boundary of ICALL technology further than it has been pushed before. The next two chapters focus on describing and motivating an implementation of a content assessment module that aims to do just that.
CHAPTER 4

DESIGN INFLUENCES

Before describing the design of CAM, it is useful to outline and discuss approaches to other computational linguistics (CL) tasks that have inspired the development of CAM. These tasks, which include natural language question answering and machine translation evaluation, among others, have different goals from ICALL content assessment and from each other. However, all share in common with content assessment the need to recognize multiple realizations of the same semantic content when evaluating the compatibility of two or more linguistic expressions. Thus, comparison techniques are often similar across fields, and the strategies used in these applications can inform an approach to comparing responses in ICALL content assessment. This chapter provides an overview of several CL tasks, their techniques for surface-level text comparisons, and the applicability of those techniques to ICALL content assessment. While not all of the techniques are ultimately used in CAM, there are clear parallels to the content assessment task that deserve attention here. The focus is on the tasks listed in Table 4.1, which provides a snapshot of the applications and the types of decisions made in each application that can be related to the decisions made in an content assessment module.

The intuitive similarity of the tasks in Table 4.1 has been acknowledged through recent research into recognizing textual entailment. In fact, it has been
<table>
<thead>
<tr>
<th>Application</th>
<th>Description</th>
<th>Relevant Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Natural Language Question Answering</strong></td>
<td>Given a query, find the answer in a text or set of documents</td>
<td>Does the candidate answer textually entail the query?</td>
</tr>
<tr>
<td><strong>Machine Translation Evaluation</strong></td>
<td>Determine similarity between machine and human translations</td>
<td>Do the machine and human translations mutually textually entail one another?</td>
</tr>
<tr>
<td><strong>Summarization and its Evaluation</strong></td>
<td>Identify similarities across texts for extracting main ideas</td>
<td>Does the text from one document textually entail the text from another?</td>
</tr>
<tr>
<td><strong>Paraphrase Recognition</strong></td>
<td>Determine whether two texts convey largely the same meaning</td>
<td>Do two candidate texts mutually textually entail one another?</td>
</tr>
</tbody>
</table>

Table 4.1: CL Tasks Related to Content Assessment

argued that textual entailment is the more general task of these CL applications (Dagan et al., 2005). One text *textually entails* another if the latter can be reasonably concluded from the former. Textual entailment includes logical entailment and is arguably a useful kind of inference. Recast as a textual entailment problem, ICALL content assessment is a matter of determining whether a learner response and a target response mutually textually entail one another. This is not the first time ICALL has been viewed from this perspective. For instance, Delmonte et al. (2005) applied their ICALL system, GETARUN, to the 2005 PASCAL Recognizing Textual Entailment (RTE) Challenge (Dagan et al., 2005). And given the similarity of the decisions to be made across tasks, further investigation into approaches to these tasks is relevant.

However, note that a large body of research exists for each of the listed applications. It is unreasonable to think that the entire breadth and depth of these fields could or should be covered in a single thesis. Rather, the focus here is on how
these tasks might approach the issue of identifying multiple realizations of the same semantic content and how those approaches relate to the task of content assessment.

4.1 Natural Language Question Answering

Natural language question answering (QA) is the task of automatically providing an answer (in natural language) to a given question. One important application of QA technology is in the field of information retrieval (IR). In QA-IR systems, the task is to accept a natural language question as a user query – as opposed to a set of keywords – and to return an answer to that question – rather than a set of documents that might contain a possible answer. From a set of candidate answers extracted from search engine results, a QA-IR system must determine which is the best response to the original query. In doing so, the QA-IR system compares the candidate answers to question representation to make its judgments.

Another application of QA technology searches a single document or passage (rather than a document collection) and returns the sentence most likely to contain the correct answer from within the passage. This application can be thought of as essay-based question answering, though it has been referred to in the literature as a reading comprehension (RC) task (Hirschman et al., 1999; Xu et al., 2006, among others). This is parallel to the reading comprehension exercises often given to second-language learners, with the computer in the role of the learner.

Both QA-IR and QA-RC systems offer comparison techniques relevant to ICALL content assessment, though the tasks differ in significant ways from the content assessment task.
4.1.1 QA-IR Systems

Much of the work in QA-IR since 1999 has been developed and tested through the periodic, NIST-sponsored Text Retrieval Conferences (TRECs). Many of the current systems share a common processing strategy. For instance, the typical approach to question answering in the 2004 TRECs was to i) identify the expected answer type, ii) search for and extract candidate passages from texts likely to contain the answer, and iii) match a representation of the question to a representation of each candidate to determine the most likely answer (Voorhees, 2004).

For example, suppose the query is *Who wrote ‘Blink’?* The question word in this query might be processed and replaced with the answer type PERSON. Keywords in the query are used to search for candidate texts. From these texts, passages such as the following might be extracted and ranked based on the similarity between the question and candidate passage:

- Malcolm Gladwell, in his book Blink, wrote about how humans tend . . .
- Blink – Author: Malcolm Gladwell . . .
- As I wrote in my review of “Blink:” . . .
- I wrote Blink Flash, a python script for uploading photos . . .

To be highly ranked, the semantic content of a candidate passage must be sufficiently similar to that of the query, where the latter may be rephrased in form such as *PERSON wrote ‘Blink’*. “Sufficiently similar” might mean that the candidate passage logically entails the query or that the query is a reasonable inference to make given the passage.

In either case, to answer the question correctly, the system must be able to recognize the semantic relationships between *Malcolm Gladwell, in his book Blink,*

---

1The candidate passages listed here were hits from a Google search performed March 29, 2006.
and the processed query PERSON wrote ‘Blink’ or between Blink – Author: Malcolm Gladwell and the processed query. It would not hurt if the system could also recognize the lack of equivalence between Blink and review of “Blink” or Blink Flash to eliminate the last two candidates.

A range of strategies have been adopted for comparing linguistic expressions in QA-IR systems. For instance, in their 2004 TREC submission, Cui et al. (2004) analyze the question and each candidate answer using the MINIPAR (Lin, 1998b) dependency parser. Their matching algorithm then compares the dependency relations in the question with those of each answer to find the most similar match. Another system, AnswerFinder (Mollá and Gardiner, 2004), combines word overlap, grammatical relations overlap and logical form overlap for matching answers to questions.

A common matching strategy uses hand-crafted surface-level patterns. For example, Kaisser and Becker (2004) employ this technique in their QuALiM system. Each question has a pattern (a combination of the question word, constituents and other lexical tokens) associated with one or more surface level answer patterns. If one of the candidate answers has a pattern linked to the question, then it is selected as the answer. For effective question answering, this approach requires that all possibly relevant question and answer patterns have been defined in advance.

There are any number of configurations of external resources, search strategies, scoring methods and answer extraction approaches possible in a QA-IR system, and it is not always clear what contribution each component makes to the success or failure of a system. One exception to this is a study by Light et al. (2001) that looks at the impact of several factors on QA-IR system performance. These factors, the study findings and the impact on QA-inspired strategies for ICALL are described below. The factors include multiple answer occurrences, word overlap scoring, weighting schemes and answer typing.
Multiple Candidates. The Light et al. (2001) study first looked at whether the number of correct passages in the document collection for a particular question impacted performance in answering that question. Their findings were affirmative: if there are multiple instances of the correct answer in the document collection, a QA system is more likely to find one of the correct answers.

This finding is not applicable to an ICALL setting. In a QA system, the role of the system is parallel to the role of the student learner, trying to provide the correct answer to a given question. The relevant QA strategies are those that involve matching content representations of questions and answers, rather than strategies for selecting candidate answers.

Overlap Scoring. Once candidate answers have been identified, a QA system compares how similar the question and answers are. In many cases, this comparison is based on simple word overlap between questions and candidate answers. With simple word overlap, questions and candidate answers are represented as 'bags of words.' Word overlap is how many words are shared in common. Word overlap scoring can be absolute counts or relative percentage of shared words or lemmas, with or without stop words. Light et al. (2001) examined the effectiveness of word overlap by studying whether overlap scores impacted a system’s ability to select the correct answer. What they found was that absolute word overlap scores did not improve a system’s performance. Rather, it was the relative overlap score. In other words, a candidate answer with a high overlap score does not indicate it is the correct answer. A better indicator is when a candidate answer has a high overlap score compared to the overlap scores of other candidates. Thus, QA scoring based on word overlap performs better in situations in which there is more than one candidate answer.
In an ICALL setting, there is only one candidate answer – the learner response. The decision to be made is whether the candidate is suitable, given the question and the target response. The finding from Light et al. (2001) – that absolute word overlap is not a useful measure but relative word overlap is – has important implications for using a QA-style bag of words surface approach for measuring similarity (either the similarity between learner response and question or between learner and target responses). The target and learner response may have a high overlap score, but high compared to what?

**Weighting Schemes.** The third factor studied in the Light et al. (2001) work is weighting schemes. Basic word overlap assumes all words in the bags of words are weighted equally. However, various weighting schemes have been tested to enhance basic overlap. The study calculated the best and worst performance for a QA system with a weighting scheme and found that between 21% and 35% of all questions could not be answered with word overlap.

This suggests that word overlap alone is insufficient for handling ICALL content assessment. This is not terribly surprising. Consider the additional example learner responses in (27).

(27) a. **Learner Response 2:** Mozart was born in 1791.

   b. **Learner Response 3:** Mozart was born in Salzburg.

Both (27a) and (27b) have high word overlap with the question, but neither answer is compatible with the target response. (27a) is simply wrong; word overlap cannot recover from this error. (27b) is true but does not answer the question. More will be said about the limits of word overlap in Chapter 5. The error in (27b), however, can be caught through answer typing.
Answer Typing. The final factor examined in the Light et al. (2001) study was the effect of answer typing on the ability of a QA system to extract the exact answer from the most likely candidate passage. QA systems commonly employ answer typing to identify the kind of answer a question is looking for. The content and complexity of answer type hierarchies vary from system to system, but most systems rely on the surface question word to identify the answer type. For example, a question such as *When was Mozart born?* is typed as some form of DATE answer type because of the question word *when*. Passages that do not contain the desired answer type are automatically rejected before questions and candidate answers are matched. This has been a successful passage selection or filtering strategy since the first TREC question answering task (Moldovan et al., 1999). In (??), if the expected answer type is DATE, then (27b) is ruled out because it does not have a DATE entity in it.

However, candidate passages may contain more than one entity of the appropriate answer type. For instance, a passage containing an answer to *When was Mozart born?* might be the fragment *Wolfgang Amadeus Mozart, 1756-1791, ....* This fragment has two entities, 1756 and 1791, which would be labeled as DATE entities. Answer typing alone has no way to identify which one is the correct date. This situation is so common in the TREC test data that Light, et al. found that the maximum accuracy for a QA system answer typing in selecting the exact answer was only 59%. Still, this last point relates more to answer selection in QA than content matching in ICALL. If crafted carefully, the target response in an ICALL situation presumably provides enough information to decide whether the learner response contains the correct answer. The catch is figuring out how to access that information in content assessment.
4.1.2 Essay-based Question Answering

Strategies for comparing candidate answers to question representations in QA-RC systems are not that different from those in QA-IR systems. For example, Hirschman et al. (1999) describe a bag of words approach for analyzing answers to a reading comprehension test in which the system, Deep Read, automatically extracted passages from a single text to answer short-answer questions. Test materials for the system came from standardized reading comprehension tests for evaluating 3rd to 6th grade reading skills. To perform the reading comprehension (RC) task, Deep Read processed the question and text sentences into a “content representation,” then searched the text sentences for the content in the question. Their content representation was simply a bag of stemmed words, augmented with the entity types and rudimentary pronoun resolution. The pronoun resolution strategy was simply to replace he, him, his, she, her to the first previous PERSON entity. Their entity types were restricted to PERSON, LOCATION and TEMPORAL types. For instance, if Mozart were in a bag of words, then the type PERSON would also go into the bag. Hirschman et al. (1999) found that these enhancements improved performance once overlap of the enhanced bags was calculated. They also experimented with removing stop words but found it hurt performance.

The Deep Read system attempted to select the best possible sentence from a text that answered a given question. To evaluate performance, the authors used precision and recall of matched content words, along with HumSent, defined as the percent of test questions for which the system has chosen a correct sentence from the text, where the correct sentence (or sentences) have been identified by a human.

However, for any given set of questions and a text, there may be questions that have no answer represented by a single sentence. In such cases, the upper bound of
performance for the system is not 100%, but reduced by the number of questions that cannot be answered adequately using such means. In the original evaluation of Deep Read, 11% of all the questions had no single-sentence answer in the texts, reducing the upper bound to 89%. Still, at best, the system answered questions correctly only 40% of the time.

Of course, the task to be discussed in this thesis is less like the QA-RC task and more like the task of evaluating such QA-RC systems. However, in QA-RC system evaluation, only identical matches to sentences selected from the text are considered correct. And in the content assessment task, the answers are not restricted to sentences from the text. Thus, the complexity of is potentially greater.

Yet, the surface-level approaches to matching (e.g., stemming, answer typing, etc.) included in the Deep Read system itself for finding answers are useful for comparing answers. Thus, it is worth looking more closely at systems developed for the reading comprehension task.

Moving beyond a bag-of-words approach, Xu et al. (2006) employ a maximum entropy model, which answers questions by selecting the sentence that maximizes the probability that a candidate sentence answers the question. This probability is computed using a normalized exponent function based on the sum of binary features. An example of how the value of a feature is determined is given in (28).

(28) If a noun and a verb match between the question and candidate sentence, the value of the noun-verb matching feature for the candidate is 1 and 0 otherwise.

In addition to such features, the system included features based on dependency triples and grammatical relations. The system achieved a 44% HumSent score on the same corpus used to evaluate Deep Read, the Remedia\texttrademark collection of texts and RC questions. In comparison, Deep Read obtained roughly 40% as a top score.
In another system to answer RC questions, Du et al. (2005) the bag-of-words approach by matching bags of words using annotations such as the main verbs, named entities and base noun phrases. This approach assigns weights to the metadata features to reflect their relative importance and considers these weights when scoring the final sentence candidates. In addition to the metadata, the Du et al. (2005) system searches texts using answer patterns derived from the Web. With these enhancements, the final system correctly answered 42% of the questions. In their analysis they argue that their 20.7% performance gain (over Deep Read’s performance) was due to the inclusion of the annotations in the matching process.

4.1.3 More on Answer Types

In QA-IR systems, answer types are used in the candidate passage selection or filtering process to improve the likelihood of finding an answer. Before candidate answers are scored for similarity to the question, the set of candidates considered must be extracted from the documents that were selected to most likely contain the answer. One system that combines a surface-based approach to answer type recognition and deeper semantic processing for answer selection is from Language Computer Corporation (LCC) (Harabagiu et al., 2003).

The LCC QA system has three basic modules – question processing, passage retrieval and answer processing.

- **Question Processing:** The type of question determines the type of question processing. Factoid (ex: *When was Mozart born?*) and list (ex: *What movies star Jodie Foster?*) questions are parsed and syntactic dependencies identified. The head of (one of) the question phrases is used to identify the answer type. An answer type is 1) one of the semantic classes in LCC’s Cicero Lite™ Named
Entity Recognizer (Surdeanu and Harabagiu, 2002) or 2) one of the semantic classes defined in a hierarchy of concepts built around WordNet’s (Miller, 1995) noun and verb synsets. For example, *actress* would be labeled as the head of the question phrase in *What actress starred in ‘Silence of the Lambs’?* In the WordNet-based hierarchy, *actress* would be a part of the hierarchy for humans and would then be associated with the Named Entity type PERSON.

For definition questions (ex: *What is a hookah?*), the questions are parsed to identify noun phrases and then matched to handcrafted patterns. For example, the question *What is TB?* might be mapped to the answer pattern *AP such as QP*, where QP is the question phrase and AP the target answer phrase. This example, from Harabagiu et al. (2003), would match a passage containing the phrase *infectious diseases such as TB*.

- **Passage Retrieval:** Passage retrieval returns a set of passages from a document collection using keywords identified during question processing. Candidate passages are filtered using the answer types. If a passage does not contain at least one instance of the identified answer type, the passage is discarded.

- **Answer Processing:** Answer processing also depends on whether the question is a factoid, list or definition question. For factoid answers, *Cicero Lite* is applied to identify and extract the exact named entity from the passage. If it is unable to do so, then it is identified with the COGEX theorem prover (Moldovan et al., 2003).\(^2\) For list answers, exact answers are extracted from ranked passages. For definition answers, pattern matching is applied.

Harabagiu et al. (2003) report that of the 289 correct answers LCC’s system provided for the 500 TREC-2003 factoid questions, 234 were identified by the answer

\(^2\)Since the focus here is on surface-based processing strategies, COGEX will not be discussed further.
type taxonomy or Cicero Lite. The remaining 65 were identified using the theorem prover. With almost half of all answers correctly identified using surface-based matching strategy and only 13% using deep processing methods, this suggests that shallow processing makes a significant contribution to question answering systems.

In general, information extraction tasks use surface-based patterns to identify concepts and relations between them. Patterns may be handcrafted or learned automatically, but typically include a combination of character strings, parts of speech or phrasal information (Grishman, 1997). For example, $<\text{Mr. } \text{+ Capitalized Word } \text{+ Capitalized Word}>$ might be a simple pattern for recognizing a NAME entity.

The modules used by Cicero Lite to identify entities include a lexical recognizer to match words or groups of words to dictionary lists or gazetteers, a preprocessor to recognize entities such MONEY, DATE or TIME entities from numerical patterns, and a named entity recognizer that also uses surface-level patterns for PERSON, LOCATION and ORGANIZATION entities.

The idea of identifying answer types and surface patterns for use in recognizing expected variation in ICALL content assessment is appealing. Such types and patterns could be used to determine whether the learner has supplied semantically appropriate arguments. This line of research will be discussed further in Chapter 7.

4.2 Machine Translation Evaluation

In order to evaluate the quality of automatically translated documents, automatic (machine) translations are often compared to manual (human) translations. The underlying assumption is the more similar the machine and human translations, the better quality the machine translation output. Many automatic methods of comparing human and machine translations have been proposed and tested, but most rely on surface-based methods of evaluating the quality of the machine translations.
The quality of a machine translation is measured in terms of its adequacy and fluency. According to the 2005 TIDES\textsuperscript{3} specification for human evaluation of machine translation quality (LDC, 2005), adequacy refers to how close the machine translation is to conveying the information contained in the source document. Fluency refers to how grammatical the translation is.

Assessing adequacy and fluency in machine translation is parallel to assessing content and form, respectively, in ICALL. Specifically, the human translation is equivalent to the stored target response in an ICALL system, and the machine translation is the learner response. The focus in this thesis is on content/adequacy assessment. However, automatic means of translation evaluation do not necessarily make separate distinctions for content and form evaluation.

4.2.1 Basic Strategies for Comparing Translations

Strategies for automatic translation assessment rely heavily on surface form and shallow processing to match or compare the similarity of a machine and human translation. Two basic strategies for comparing translations are edit distance and n-gram overlap. Minimum edit distance, or Levenshtein distance (Levenshtein, 1966), measures the cost of \textit{transforming} the automatic translation into the human translation through a series of insertions, deletions and substitutions. The higher the cost, the less desirable the automatic translation.

Word Error Rate (WER) is a basic edit-distance measure for evaluation (Niessen et al., 2000). Although machine translation evaluation typically involves comparison of a machine translation with multiple human translations, to make the description simpler here, assume the comparison is between a single machine translated (MT)

\textsuperscript{3}TIDES stands for Translingual Information Detection, Extraction and Summarization and is a DARPA-funded program to promote the development of cross-lingual language processing technology.
sentence and a human (reference) sentence. WER is the sum of all the insertions, deletions and substitutions required to transform the translation sentence into the reference sentence, divided by the length of the reference sentence. Using this measure, the closer the WER is to zero, the better quality the MT sentence.

However, WER is sensitive to the order of words in a sentence. For example, consider the sentences in (29).

(29) a. **Reference Translation:** My little terrier hides under the bed.

b. **Machine Translation 1:** My small dog hides beneath the couch.

c. **Machine Translation 2:** Under the bed is where my little terrier hides.

Even though (29c) is a perfectly reasonable variation of the hypothetical reference translation and a better translation than (29b), it will have a higher (worse) WER score because it costs more (takes more steps) to rearrange the words in (29c) than substitute words in (29b).

Tillmann et al. (1997) implement an alternative edit distance measure that ignores word order. Position independent error rate (PER) assumes that the reference sentence and the MT sentence are bags of words. Overlapping words in the two bags are matches and have no cost. Words that do not match are considered substitutions. Remaining words are counted as deletions (if the MT sentence is longer than the reference sentence) or insertions (if the reference sentence is longer than the MT sentence). Using this method of counting costs, the PER score is calculated in a similar manner as WER. The main appeals of using some form of edit distance are that it is easily implemented and fully automatic as a form of evaluation.

The second basic evaluation strategy that is similarly easy to implement is n-gram analysis. In this context, a *n-gram* is a continuous segment of text, *n* words long. The **BLEU** (BiLingual Evaluation Understudy) metric (Papineni et al., 2001) basically counts the overlap of n-grams in the machine and human translations. It also relies
on the idea that the closer the machine output is to a string match (here, calculated in terms of the number of substring matches), the better the translation. Papineni et al. (2001) argue the longer the n-gram matches, the better the fluency; the more words in common, the better the adequacy. Given that the development of BLEU triggered widespread research into metrics for machine translation evaluation and its continued use as such a metric\(^4\), additional discussion is warranted. An overview of the BLEU algorithm itself is located in Appendix B.

Like edit distance, one appeal of BLEU is its simplicity, not to mention low cost and speed of the evaluation. But, more importantly, Papineni et al. (2001) found that on an entire test set, BLEU’s scoring results positively correlate with human judgments. This suggests that BLEU is a good measure of the overall performance of an MT system. However, BLEU is not without its faults, all of which are relevant to the current discussion of MT evaluation methods for ICALL:

- **No Recall.** The BLEU score pays attention to how much of a candidate translation is correct (i.e., its precision) but completely ignores how much of a reference translation is actually represented by the candidate (i.e., the candidate’s recall). So, a candidate translation set that covers only half the content of a reference translation (but covers that half well) could get a decent BLEU score. This problem has been noted by several authors (Lavie et al., 2004; Banerjee and Lavie, 2005; Melamed et al., 2003, among others), who have proposed alternatives to address the issue. In an ICALL context, this problem would be equivalent to marking a learner response correct because it contains only content from the target response, even though it does not contain *all* the target content.

\(^4\)A version of BLEU is currently used as the evaluation metric for the regular machine translation “bake-off” hosted by the National Institute of Standards and Technology (NIST).
• **‘Big Picture’ Performance.** The score produced by BLEU is for an entire test corpus. The high correlation with human judgments is at this test-set level, not at the sentence level. In fact, Blatz et al. (2003) found that at the sentence level, BLEU’s judgments *do not* correlate with human judgments. This is a particularly critical issue in the current context because the comparisons to be made between target and learner responses for one exercise will be independent of comparisons made for another exercise, and most comparisons will be made at the single-sentence level (or below).

• **Lack of Syntax.** An n-gram approach does not explicitly account for syntactic variation. Liu and Gildea (2005) discuss this issue and how evaluation can be improved through incorporating an awareness of syntax. This, too, is relevant in considering the usefulness of an n-gram approach for ICALL content assessment. Ideally, an assessment model should be sensitive to legitimate syntactic variation.

• **Multiple Reference Translations Required?** Papineni et al. (2001) mention investigating how many reference translations were required for each candidate for BLEU to perform well. But it is unclear what exactly the impact of multiple references is. Their initial findings were that BLEU did fine\(^5\) on the evaluation set as a whole when only one reference translation was available for each comparison, as long as a single author did not provide every reference translation in the set. This last point is an important potential limiting factor since multiple target responses (or single targets from different authors) is unlikely in an ICALL system in which all target responses are usually specified by a single instructor.

\(^5\)No numbers were reported.
Given the drawbacks of the BLEU metric and subsequent work to address those issues, there are more appropriate MT evaluation strategies relevant to ICALL content assessment. One such possibility is the METEOR metric (Banerjee and Lavie, 2005).^6^ METEOR (or Metric for Evaluation of Translation with Explicit ORdering) was designed to overcome some of the concerns with BLEU, while still relying on surface methods of evaluation. Processing with METEOR has two steps: unigram mapping and translation scoring.

Given a candidate machine translation (C) and a reference translation (R), METEOR first attempts to align every unigram \( c \in C \) with a unigram \( r \in R \) and vice versa. To define an alignment, the system first lists all possible unigram mappings. For instance, if \( c_i \) and \( c_j \) are two instances of the same unigram in C and \( r_k \) is another instance of that unigram in R, then the mappings \( (c_i, r_k) \) and \( (c_j, r_k) \) will be suggested.

There are three separate modules responsible for suggesting mappings – Exact Match, Porter Stemmer, and WN Synonymy. Exact Match is applied first to search for mapping identical unigrams. For example, it would match \textit{walked} in C with \textit{walked} in R, but not with \textit{walking} in R. After Exact Match identifies possible mappings, the unmapped unigrams from C and R are passed to the Porter Stemmer module, which (as the name suggests) uses the Porter stemmer (Porter, 1980) to strip the endings off words. So, if \textit{walk} was in R but not \textit{walked}, then \textit{walked} in C could be mapped to \textit{walk} in R. Once the Porter Stemmer module has mapped what it can, the remaining unigrams are passed to WN Synonymy. This module maps unigrams if they are synonyms, using WordNet (Miller, 1995) to identify the synonyms. For instance, WordNet lists \textit{locomote} as a synonym of \textit{walk}. If the former is in C and the latter in R, that mapping would be suggested. The order of module application can be adjusted, but this particular order produced the best results for Lin and Och’s study.

^6^ Another is the ROUGE metric (Lin and Och, 2004), to be discussed in Section 4.3.3.
After all possible mappings are suggested by the modules, the largest subset of mappings from the list are selected such that every unigram from C maps to at most one unigram in R. If there is more than one subset of the same length that meets this criteria, then the system selects the subset with the fewest “crosses.” A cross occurs if for \( c_i \) and \( c_k \) in C and \( r_j \) and \( r_l \) in R, where there are mappings defined as \((c_i, r_j)\) and \((c_k, r_l)\), there is a negative value for the formula in (30).

\[
\text{(30) } (\text{position}(c_i) - \text{position}(c_k)) \times (\text{position}(r_j) - \text{position}(r_l))
\]

Position\((xy)\) refers to the position of the unigram \(xy\) in translation \(X\). Thus the subset selection process favors mappings between unigrams that occur in the same linear proximity in C in R. This is a way of factoring in word order. Also note that it is possible that some unigrams in C and R will not be mapped to anything after this stage.

The second phase of METEOR processing is translation scoring. To produce a score for a candidate C, given reference R, METEOR computes unigram precision, unigram recall, Fmean and a Penalty. The measurements are given in (31) and described below.

\[
\text{(31) a. Unigram Precision (P) } = \frac{\text{Count(Mapped Unigrams)}}{\text{Count(Unigrams in C)}}
\]

\[
\text{b. Unigram Recall (R) } = \frac{\text{Count(Mapped Unigrams)}}{\text{Count(Unigrams in R)}}
\]

\[
\text{c. Fmean } = \frac{10(PR)}{(P+9R)}
\]

\[
\text{d. Penalty } = 0.5 \times \frac{\text{Count(Chunks)}}{\text{Count(Mapped Unigrams)}}
\]

Again, precision is a measure of how many of the unigrams in C are correct. Recall measures how many of the unigrams in R are also in C. Fmean is a variation of the harmonic mean of precision and recall that puts more weight on recall.\(^7\)

\(^7\)Standard harmonic mean (or F-measure) is defined in the following formula, \(F\)-measure = \[
\frac{2PR}{P+R}.
\] In many situations, as recall increases, precision decreases or vice versa, and F-measure is a way to balance precision and recall (Jurafsky and Martin, 2000). Weighted F-measure allows more importance to be placed on either precision or recall.
METEOR’s creators justify this preference for recall with their earlier study, which showed that recall is a better measure than precision for indicating a candidate translation’s correlation with human judgments (Lavie et al., 2004).

Finally, the penalty measure takes into consideration longer-than-unigram matches between C and R. All mapped unigrams in C are placed into chunks such that the unigrams are adjacent in the original string in C and in R. Thus, the longer the n-gram, the fewer the chunks. The maximum number of chunks for any C is the total number of mapped unigrams, while the lowest number of chunks is 1 (if R and C are identical). The Penalty score is 0.5 times the count of chunks, relative to the number of mapped unigrams in C. So, in the worst case – when there are no n-grams where n > 1, the number of chunks equals the number of unigrams and the penalty is 0.5.

Using this information, METEOR calculates a final score using the formula in (32). If multiple reference translations are available, a METEOR score is calculated for each and the highest score selected.

(32) METEOR Score = Fmean * (1 - Penalty)

METEOR scoring is sentence-by-sentence, but the version of BLEU they used computed a single overall score for the entire test set. So, to compare results, Banerjee and Lavie computed METEOR scores for all the candidate translation segments, then combined the scores into one measurement reflecting system performance. They found that METEOR’s aggregate Pearson’s r correlation score was .964, as compared to .817 for BLEU.\(^8\)

\(^8\)The Pearson’s r correlation score is a value between 1 and -1. It is a measure of linear relatedness. A positive correlation means the system’s evaluation of translations are at least somewhat in tune with human evaluations of those same translations. A score of 1 means there is a perfect, positive linear relationship. This means that when the MT evaluation score assigned to a translation is high, the human score for that translation is also high. When the MT score is low, the human score is low. The closer a Pearson’s r score is to 1, the more likely it is that the MT evaluation metric is making
However, looking at METEOR’s sentence-by-sentence correlations, the correlation scores are rather low. The tests were performed on translations of Arabic into English and on translations of Chinese into English. On the Arabic data, the correlations with human judgments on individual sentences averaged only .347. On the Chinese data, the average was only .331. However, these scores outperform the correlation scores when either precision or recall was used as the translation evaluation metric. Still, the scores have implications for using an approach like METEOR in ICALL content assessment system.

4.2.2 Applicability to an ICALL Task

If an evaluation strategy like METEOR is used to determine whether a learner response contains the same content as a target response, then it must make decisions similar to those of human instructors. Thus, the low correlation with human judgments on individual candidate translation sentences is troubling. This suggests that the quality of judgments made by METEOR is not high enough to use METEOR alone in an ICALL content assessment system.

However, this does not mean the approach is a loss. One particularly appealing aspect of METEOR is the set of mapping modules. The modules described in Lavie et al. (2004) are limited to matching exact words, stems or synonyms. This covers situations when there is no variation between the learner and target responses or there is expected variation based on morphological form and synonyms. But there is nothing limiting the approach to just these three modules. One modification to METEOR worth exploring is a matching module that maps unigrams of concepts, rather than judgments comparable to those of humans. Formally, the Pearson’s $r$ correlation is calculated as the following, where $X$ and $Y$ are observed values for the variables, and $N$ is the number of instances (Weiss, 1993):

$$r = \frac{\sum XY - \frac{\sum X \sum Y}{N}}{\sqrt{\left(\sum X^2 - \frac{(\sum X)^2}{N}\right)\left(\sum Y^2 - \frac{(\sum Y)^2}{N}\right)}}$$

112
unigrams of words or stems. This, of course, would require processing the target and learner responses to replace words with their most likely concepts. And the question of where these hypothetical concepts might come from is nontrivial. Strategies from concept recognition might include

- consulting WordNet or the Cyc ontology (Siegel et al., 2004) for concept labels,
- identifying broad concepts such as PERSON or LOCATION with Information Extraction (IE) tools, and
- borrowing techniques from paraphrase recognition for recognizing and mapping between identical concepts.\(^9\)

But assuming mappings between words or concepts can be strengthened through additional mapping modules, ICALL exercises that might vary in word choice (rather than syntactic structure) could be processed using a METEOR-like strategy. Consider the picture in (33), which was used in an exercise for practicing descriptions of objects in relation to other objects.\(^{10}\)

\(^9\)Refer to Section 4.4 for more explanation.

\(^{10}\)Exercise adapted from Oxenden et al. (1997).
If the associated task is to describe objects in the picture, the activity is essentially vocabulary labeling. But learner responses can vary in how descriptive they are. Two acceptable responses with possible variants are listed in (34).

(34) a. two wooden chairs, a pair of chairs, chairs

b. a vase of flowers, flowers, daisies

Mapping two to a pair of and wooden chairs to chairs, as well as daisies to flowers or flowers to vase of flowers would improve the match if one of these acceptable responses is actually the target response. If there is little or no possible syntactic variation, METEOR’s strategy would work.

However, if the expected variation is syntactic, then METEOR is not as equipped to handle such differences. The algorithm penalizes candidate translations that do not have the same word order as the target. For an ICALL system targeting a language with flexible word order, such as Spanish, the algorithm is less ideal. Syntactic variation means the word order will not necessarily be the same. Learner responses containing syntactic variation would need to be handled by a different mechanism.

This points to a general comment about machine translation evaluation strategies. MT evaluation metrics largely ignore grammar to the point that severely ungrammatical translations can be scored very highly when compared to a reference translation. Grammatical, but semantically incorrect translations in which arguments are reversed might also be scored highly if the word overlap is high enough. Input to the content assessment module may have already been evaluated for grammatical errors. But grammatical syntactic variation cannot be evaluated using any approach that ignores syntax entirely. The extent to which syntactic variation can be dealt with indirectly – through word order or distance – remains to be determined.
4.2.3 An MT Evaluation Approach to Automatic Grading

The use of MT evaluation strategies for grading tasks is not unprecedented. Recall that Chapter 2 provided an overview of the Willow/Atenea system (Pérez et al., 2005a), which is actually based on an MT-approach. Specifically, Willow/Atenea relies on a modified version of the BLEU algorithm, called ERB (Evaluating Responses with BLEU), to grade responses to questions about computer science concepts. The basic algorithm compares the student answer to a set of reference answers. The more similar the responses, the higher the ERB score.

\[
ERB_{\text{score}}(A) = MBP(A) \times e^{\sum_{n=1}^{N} \log(MUP(n))/N}
\]

ERB is calculated as in (35), where \(A\) is a student answer, \(MBP(A)\) is a modified brevity penalty (described below) of the answer, \(n\) is the length of n-grams, \(N\) is the maximum n-gram length and \(MUP(A)\) is the Modified Unified Precision score of the answer. The MUP score reflects the overlap of n-grams between the student answer and a reference answer, and ERB calculates MUP for several n-gram lengths. The MBP is penalizes short responses to take recall score into consideration in addition to the precision score that is already reflects in the MUP values.

The Willow/Atenea system incorporates Latent Semantic Analysis (LSA) in the scoring as well to measure the semantic similarity of student and reference answers. In one study, Pérez et al. (2005a) found that ERB and LSA scores were highly uncorrelated. That is, a high ERB score might not get a high LSA score. Thus, ERB and LSA are used to score the pairs independently and then combined into a final overall score.

Marín (2004) use ERB in Atenea/Willow for both formative and summative assessment. The latter is the final score, while for the former, the concepts from the student’s answer are highlighted in different colors depending on how well those
Figure 4.1: A Screenshot of a Graded Response in Willow

concepts matched with concepts in the reference answers. For example, the feedback in Figure 4.1 includes the score and the student answer, with overlapping terms highlighted in green. According to the activity directions, a passing score would have been 0.5, as determined by the instructor, with the question worth up to 1 point.

In their evaluation of the ERB and LSA components, the authors found that ERB outperformed both the original BLEU and LSA systems alone. When combined, the LSA component trained on a large corpus combined with ERB performed best, though the overall contribution of LSA to the performance was small. Alfonseca and Pérez (2004) also compared BLEU performance when combined with several linguistic annotations including stemming, word sense disambiguation and lexical dependencies. However, they found no significant difference in performance with any of the configurations. This may be because student responses showed less linguistic variation from the targets and each other, given that all question cues related to technical computer science terminology.
4.3 Summarization and its Evaluation

Automatic text summarization is the task of reducing one or more source texts into an abridged version (i.e., a summary). The summary may be an abstract in which the contents are largely original and may be indicative (i.e., intended to highlight what a text is about) or informative (i.e., intended to be sufficiently detailed as to serve as a replacement for the source texts) (Edmundson, 1999). A relatively simpler summarization task is to produce an indicative or informative extract for which the most important words, phrases or sentences in a text are extracted from the source text(s), organized and (possibly) edited as the final product of the system. It is extract production and evaluation that is the focus of much summarization research.

Just as research into question answering has been spurred on by the TREC conferences, summarization research has recently been encouraged through the Document Understanding Conferences (DUCs) in which participants’ summarization systems are evaluated against a common test set in order to gain some measure of progress in the field.

Hovy (2005) summarizes the general strategies for machines to produce automated summaries. He outlines three stages of the process:

- **Topic Identification:** Identifying the most important topics (or text chunks containing topics) to be used in the extract or abstract.

- **Interpretation:** Topics are fused together and combined with material not in the source text(s). This stage produces abstracts, rather than extracts.

- **Generation:** From the identified concepts or text chunks, generate a summary. For extracts, this stage might including smoothing extracted phrases and sentences to make them more grammatical. For abstracts, natural language generation is required.
Topic identification has received the most attention to date in the summarization literature. Extracts can be produced at this stage with no additional interpretation or generation required. This form of summarization that relies on surface features and extraction is the focus here.

4.3.1 A Basic Summarization Strategy

Any multi-document summarization technique that identifies the points of similarity between two texts potentially has something to say about ICALL content assessment. One such system is from Mani and Bloedorn (1997). Their approach builds a representation of the similarities and differences of two texts to be summarized by using surface-level methods of extracting entities and relations between them. The process in Mani and Bloedorn’s system has four steps:

1. Build a graph of concepts and relations for each source text contributing to the extract.
2. Search the graph for semantically related concepts using spreading activation.
3. Find the similarities and differences between the graphs.
4. Extract sentences representing these similarities and differences as the summary.

In building the graphs, concepts represented by words, phrases and proper names in a text are defined as the nodes of the graph. Arcs between nodes represent lexical, syntactic and associative relations between concepts. The position of a word/concept in the source text is captured by the position of the node in the graph. Thus, the graph preserves the order of concepts in the source text. Each node has a set of features associated with it. These features include the position of the word in the document, the position of a word in its sentence, and its activation weight (to be
described). Mani and Bloedorn (1997) explored the following types of arcs between the nodes, although other relations might be easily added to the representation:

- Adjacency arcs to connect nodes for words that appear next to each other
- Identity arcs to connect nodes that are different instances of the same word
- Phrase arcs to tie together nodes whose words belong to the same phrase
- Coreferential arcs to connect nodes whose words refer to the same entity
- Synonym arcs to connect different nodes whose words have the same meaning
- Hypernym arcs to connect nodes whose words participate in a hyponym-hypernym relation

To build a graph, each text is passed through a sentence and paragraph tagger to identify word positions. This output is part-of-speech tagged using the Alembic tagger (Aberdeen et al., 1996). The entities (proper names, people, organizations, etc.) were identified using SRA’s NetOwl information extraction system (Krupka and Hausman, 1998).

Weights reflecting salience were then associated with each word. A word’s weight is the inverse document frequency calculation given by (36), where $k$ is a term in document $i$, $tf_{ik}$ is the number of times $k$ appears in $i$, $N$ is the total number of documents in a reference corpus and $df_k$ is the number of documents in the reference corpus containing $k$.

$$tfidf_{ik} = tf_{ik} \times (\ln(N) - \ln(df_k) + 1)$$

Inverse document frequency ($tfidf$) is a measure of the relative importance of a term. It is used in information extraction to identify words that are good at distinguishing one document from another. To generate useful $tfidf$ scores, the authors needed a large collection of data (the reference corpus). They used the corpus from a TREC conference (Harman, 1994).
Given that phrases can represent important concepts, the Mani and Bloedorn system identifies phrases using the part of speech tags and a set of surface patterns. Once phrases were identified, they were also assigned a weight, calculated as in (37).

\[
\text{Weight}(\text{phrase}, i) = \beta(n) + \frac{\sum_{k=1}^{n} \theta(ik) \cdot \text{tfidf}_{ik}}{n}
\]

In (37), \(\beta(n)\) is a bonus value proportional to the length of the phrase (n). This means the weighting favors longer phrases, which tend to be more specific. Otherwise, the weight of a phrase in text \(i\) is the average of the sum of \(\text{tfidf}\) weights for the words in the phrase multiplied by \(\theta(ik)\), which is either 1 if the word is new or 0 if the word has been seen before. This last factor \(\theta(ik)\) avoids redundancy in phrases.

Once a graph is built for each source text, spreading activation is applied\(^\text{11}\) in order to find all nodes that are semantically related to the “active” nodes. How active a given node is depends on the distance from that node to an entry node and the type of arc(s) that connect the nodes. For Mani and Bloedorn’s system, an entry node is based on a topic specified by the user. This allows them to tailor summaries to the interest/needs of different users.

Activation, a measure of the salience or importance of a concept, spreads from one node to another via arcs of the types described above and from one text to another via identity arcs. In other words, if a word or its stem is active in one text, it will be activated in any other text it (or its stem) appears in. Crucial to the idea of spreading activation is that how active a node is decays the further away from the source of activation or the weaker the links to the source of activation. For instance, there is a bigger “penalty,” meaning the activation is weaker, when the distance between nodes is across sentences as compared to within sentences. Entry nodes have the highest weights, and weights of other nodes are raised or lowered through the activation process.

\(^{11}\)Mani and Bloedorn cite Chen et al. (1994) as their source for the spreading activation approach.
When a node is activated, its weight is recalculated to reflect its importance. If the arcs connecting a node to an activating node are adjacency arcs, then reweighting of the node depends on both its distance from its activating node and the weight of that activating node. Reweighting for other arc types depends on the type of arc connecting a node to its activating node and the activating node weight.

Once activation is calculated, the third stage is to find the similarities and differences between activated graphs. The goal at this stage is to compare graphs to find the intersection of nodes. These are the similar concepts in the graph, defined as in (38), where \( c \) is a concept and conceptMatch\( (c, \text{GraphX}) \) holds if there is a concept \( c_1 \) in the other graph such that \( c \) and \( c_1 \) are identical words, have identical stems or are related by synonymy. The activated weights play a role in that the user can define the minimal weight required for a concept to be considered in the final sets of similar and different concepts.\(^{12}\)

\[
(38) \text{Common} = \{c \mid \text{conceptMatch}(c, \text{Graph1}) \text{ and conceptMatch}(c, \text{Graph2})\}
\]

The last phase of this approach to summarization is to extract sentences containing similar concepts and different concepts, using the weights to score sentences with respect to which sentences are most representative of the relevant concepts.

4.3.2 Applicability to an ICALL Task

On the surface, it seems that the tasks of summarization and ICALL content assessment have little in common. In summarization, the goal is to produce a summary that is shorter than the original document(s), but contains all the important content. For content assessment, the goal is to determine whether a learner response contains all the semantic content of the target response. However, in both tasks, the semantic

\(^{12}\)The differences in two texts are calculated by taking the union of concepts and removing those in the set of Common concepts.
content of texts must be evaluated. In multi-document summarization, two or more source texts are processed to identify important topics (i.e., concepts) that are shared among documents and important concepts that are document-specific. All common and divergent concepts must be extracted and merged to create a complete summary that covers all the concepts of interest. Recast as a summarization problem, the learner response and target response in ICALL are the texts. Their similarities and differences must be evaluated as in summarization. If there are different concepts in either the learner or target texts, that is a signal that the learner has made a content error.

In adapting the spreading activation approach to an ICALL setting, the entry nodes would be required concepts, as specified either separately in the activity model or as part of the target response. Because the approach intends to connect concepts, rather than words or phrases, it already assumes variation between wordings of extracts is likely. This is a helpful assumption for adapting the approach to ICALL content assessment. The set of common concepts generated by the algorithm would identify the concepts in the learner response that are also in the target response. These are the concepts the learner got right. The set of different concepts, however, would need further analysis.

Concepts in the target response not in the learner response are ones the learner missed. If required, these concepts represent learner errors of omission. Concepts in the learner response not in the target response are either 1) extraneous, but not incompatible with the target or 2) incompatible with the target response. Spreading activation weights may help distinguish between the two; extra concepts within some weight threshold might be acceptable, while those below the threshold would be incompatible given the target response.¹³

¹³One issue with weights is what initial weighting scheme to use. Mani and Bloedorn (1997) use $t_f d_f$ weights, which require a large corpus of text to generate. For English, such corpora are available.
ICALL activities involving writing essays seem suited to this type of evaluation strategy. There may not be a single correct target response, but there may be a set of concepts the instructor requires in the learner essay and a *model* response to compare the learner response against. For example, consider the exercise in (39).

(39) After reading a short story about the lives of two men, Gene Bridges and Gordon Chang, the learner is asked to write a paragraph answering the following question: What were four ways in which Gene Bridges and Gordon Chang were alike?

In this exercise, there are four points the learner needs to include in his response. But the level of detail the instructor is looking for might be flexible. A model answer might contain both the required concept points and optional additional information. Spreading activation might be a good way to capture degrees of importance.

However, the suitability for using spreading activation on short answer evaluation is unclear. There are likely fewer concepts in a single-sentence response than in an paragraph-length essay. Activation changes weights based, in part, on the distance of the activating node from the node whose weight is being changed. If there are fewer concepts because the answer is shorter, the distance between any two concepts will automatically be shorter. This alters the ability of the approach to distinguish between highly salient concepts, and less salient concepts. On the other hand, if salience is partially determined by a threshold, then adjusting the threshold for shorter answers may resolve the issue.

---

for other languages, perhaps not. But there is nothing about the spreading activation approach that requires the use of *tfidf* weights. Another weighting scheme might be more appropriate for the ICALL context.

14Exercise from Laubach et al. (1991).
4.3.3 Summarization Evaluation

Turning away from summarization strategies, a related task that is perhaps more intuitively similar to the content assessment task is summarization evaluation. The evaluation of automatically generated summaries can be automated in a way very similar to the evaluation of machine translations. Given one or more reference summaries, the content of the reference and candidate summary can be compared and evaluated for similarity just as candidate and reference translations are.

One automatic, surface-based strategy for evaluating the responsiveness of summaries that is very similar to the strategies used for machine translation evaluation is ROUGE (Lin, 2004). In fact, ROUGE has been applied to machine translation evaluation as well (Lin and Och, 2004). ROUGE is comprised of several evaluation strategies, but the two basic strategies to be discussed here are ROUGE-L and ROUGE-S, following the naming conventions of Lin and Och (2004).

ROUGE-L is based on identifying the longest common subsequence (LCS) between a candidate and reference summary and calculating a weighted F-measure as the similarity score, much like METEOR does. To explain LCS and how ROUGE-L works, consider the example in (40).

(40) a. **Reference Summary**: brown bears eat salmon and honey

b. **Candidate Summary 1**: brown bears eat fish and honey
c. **Candidate Summary 2**: bears eat salmon
d. **Candidate Summary 3**: fish eat brown bears and honey
e. **Candidate Summary 4**: bears attack tourists

The LCS is the longest sequence of words in a candidate summary that is also in the reference summary *in the same order*. LCS requires that the words be in sequence, but not that words be consecutive. For instance, the LCS of (40b) is *brown*
<table>
<thead>
<tr>
<th>Candidate</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Candidate 1</td>
<td>5/6</td>
<td>5/6</td>
<td>.83</td>
</tr>
<tr>
<td>Candidate 2</td>
<td>3/3</td>
<td>3/6</td>
<td>.67</td>
</tr>
<tr>
<td>Candidate 3</td>
<td>4/6</td>
<td>4/6</td>
<td>.67</td>
</tr>
<tr>
<td>Candidate 4</td>
<td>1/3</td>
<td>1/6</td>
<td>.22</td>
</tr>
</tbody>
</table>

Table 4.2: ROUGE-L Scores for Hypothetical Candidate Summaries

*bears eat and honey* and for (40d), the LCS is *brown bears and honey*. To compare a candidate (C) to a reference summary (R), ROUGE-L calculates an LCS-based precision, recall and F-measure scores, given in (41). In (41c), $\beta$ is a parameter that controls the weight given to precision or recall. When $\beta = 1$, precision and recall are given equal weight. The results reported in Lin and Och (2004) make this assumption.

\[
\begin{align*}
\text{a. Precision}_{LCS} &= \frac{\text{length}(\text{LCS}(C,R))}{\text{length}(C)} \\
\text{b. Recall}_{LCS} &= \frac{\text{length}(\text{LCS}(C,R))}{\text{length}(R)} \\
\text{c. F-Measure}_{LCS} &= \frac{(1+\beta^2) (\text{Precision}_{LCS} \ast \text{Recall}_{LCS})}{(\text{Recall}_{LCS} + \beta^2 \text{Precision}_{LCS})}
\end{align*}
\]

Using these measurements, the scores for the hypothetical candidates in (40) are given in Table 4.2. These scores for best and worst summary – (40b) and (40e), respectively – are reasonable, given the reference summary. However, intuitively, (40c) is much better than (40d) even though they get the same summary score. (40c) is missing information, whereas (40d) mangles the order of the arguments. Lin and Och (2004) suggest a weighting scheme for calculating LCS such that consecutive words are preferred. This would make a clearer distinction between (40c), which is entirely consecutive, and (40d), in which *brown bears* is consecutive and *and honey* are consecutive, but the entire subsequence is not.

The second ROUGE strategy, ROUGE-S, also uses a variant of F-measure, but the variant is based on skip bigrams. Skip bigrams are all possible word pairs,
in sequence but not necessarily consecutive, in a sentence. For example, Candidate 1 in (40b) has 15 skip bigrams: *brown bears, brown eat, brown fish, brown and, brown honey, bears eat, bears fish, bears and, bears honey, eat fish, eat and, eat honey, fish and, fish honey, and and honey.* To calculate a score for a candidate (C) and reference (R) summary, ROUGE-S counts the number of matching bigram skips and applies formulae in (42).

\[(42)\]

\[\begin{align*}
\text{a. } \text{Comb}(L, 2) &= \frac{L!}{2!(L-2)!}, \text{ where } L \text{ is the length of the string} \\
\text{b. } \text{Precision}_{\text{skip}} &= \frac{\text{SKIP}(C,R)}{\text{Comb(length}(C),2)}, \text{ where } \text{SKIP}(C,R) \text{ is the number of skip bi-} \\
&\text{gram matches between } C \text{ and } R \\
\text{c. } \text{Recall}_{\text{skip}} &= \frac{\text{SKIP}(C,R)}{\text{Comb(length}(R),2)} \\
\text{d. } \text{F-Measure}_{\text{skip}} &= \frac{(1+\beta^2)(\text{Precision}_{\text{skip}} \cdot \text{Recall}_{\text{skip}})}{\beta^2 \text{Precision}_{\text{skip}} + \text{Recall}_{\text{skip}}}
\end{align*}\]

The final ROUGE-S score, the F-measure in (42), is calculated the same as in ROUGE-L. Variations on ROUGE-S include limiting how far away two words can be to be considered a skip bigram.

Much like translation evaluation, the automatic summarization evaluation strategy focuses on identifying words common to a candidate and target summary. In fact, the METEOR strategy for MT evaluation in Section 4.2.2 and the ROUGE system are very similar. Both use precision and recall calculations as crucial factors in scoring text relatedness. Both are surface-based approaches that only indirectly address syntactic issues by paying attention to word order. For instance, METEOR penalizes candidates that have word order that varies from the target, and ROUGE-L prefers the longest identical match between candidate and target. However, ROUGE-S is slightly less focused on matching identical word order and this is an advantage of adapting ROUGE for content assessment. For example, compare (43a) and (43b).

\[(43)\]

\[\begin{align*}
\text{a. } &\text{Brown bears eat salmon and honey.} \\
\text{b. } &\text{Salmon and honey are what brown bears eat.}
\end{align*}\]
Even though these examples have equivalent meaning, METEOR and ROUGE-L penalize the alternative syntactic structure. ROUGE-S, on the other hand, focuses on bigrams that occur in sequence. While “in-sequence” bigram counting does factor in word order, like METEOR and ROUGE-L, it is a little more flexible than other two methods. Thus, if (43a) and (43b) were a target response and learner response, respectively, ROUGE-S has a better chance of finding a match.

4.4 Paraphrase Recognition

A paraphrase is an expression that conveys approximately the same semantic content as its source expression, using a different surface form. Paraphrase recognition is the task of determining whether two words, phrases or sentences are paraphrases of one another. In most cases, meaning need not be strictly identical across two expressions for those expressions to be considered paraphrases. For example, the sentences in (44) may be considered paraphrases even though the content conveyed does not share a one-to-one correspondence between words and phrases.\textsuperscript{15}

(44) a. After the latest Fed rate cut, stocks rose across the board.

b. Winners strongly outpaced losers after Greenspan cut interest rates again.

There is a direct parallel between two paraphrases and two translations of the same source text. The key difference is length. In the literature, the length of the text to be compared range from short phrases to short paragraphs, whereas translations may be of longer source texts. Summaries and paraphrases also share similarities. Summaries tend to convey the same semantic content as \textit{the major points of} its source text, possibly in identical form (in the case of extracts). However, in

\textsuperscript{15}Example from Barzilay and Lee (2003).
practice, depending on the length and content of a summary/paraphrase, there may be little difference in the semantic content represented in or the grammatical form expressed by a paraphrase and a summary.

Much of the research into paraphrase recognition for phrases and sentences has focused on automatically identifying patterns using machine learning techniques (Glickman and Dagan, 2003; Duclaye et al., 2003; Barzilay and McKeown, 2001; Poibeau, 2004, among others).

Surface-level approaches to paraphrase identification use these learned lexico-syntactic patterns to find paraphrases in text, much like the patterns used to extract entities in information extraction tasks. Shinyama et al. (2002) take such an approach to finding paraphrases in news text. Their assumption is that named entities (proper names, locations, etc.) can act as anchors for defining paraphrasing patterns. Matching candidate paraphrases involves using a similarity score based on \( tfidf \) weights and the number of named entities in common between the candidates. The key components – \( tfidf \) weighting and named entity concepts – help identify which concepts are more important in the final comparison of texts.

The machine learning approaches often require aligned monolingual corpora in order to have pairs of aligned sentences for training. For instance, Barzilay and Lee (2003) use a parallel corpus of aligned literary text to learn patterns. While machine learning approaches are of interest, the lack of a sufficient amount of available alignment parallel learner corpora is no small issue. More will be said about this issue in Chapter 7.

But given the similarities of the tasks (and the relevance of comparing content at or below the sentence level), paraphrase identification and/or evaluation warrants further exploration with respect to its usefulness to ICALL content assessment.
4.4.1 Techniques in Paraphrase Recognition

Two techniques in paraphrase recognition that have ties to machine translation and evaluation are those employed by Quirk et al. (2004) and Hatzivassiloglou et al. (1999). Quirk et al. (2004) present an approach to paraphrase generation that relies on alignments across a parallel corpus. Their corpus is a set of news articles discussing the same event drawn from the Web.

To align sentences, the researchers used minimum edit distance (Levenshtein, 1966) to calculate the number of operations (insertions, deletions and substitutions) to convert one sentence to another. Pairs of sentences with an edit distance below an empirically determined threshold were selected as possible paraphrases. For each pair of sentences, the words were aligned using Giza++ (Och and Ney, 2000). From the word-aligned pairs, phrases that were likely paraphrases were extracted and stored in a database for generation purposes.

In a more linguistic approach, Hatzivassiloglou et al. (1999) propose a machine learning technique for automatically learning paraphrases that relies on what the authors refer to as “primitive” and “composite” features. The primitive features include word overlap, noun phrase matches, synonym overlap, semantic verb class overlap, and proper noun overlap. These primitive features are much like what has already been discussed in terms of mapping similar tokens across texts in METEOR translation evaluation or Mani and Bloedorn’s summarization technique. Composite features are intended to capture the similarity between relations expressed within texts. These features include the following:

1. **Ordering.** Two primitives in a given text map to corresponding primitives in another text and those primitive pairs appear in the same relative order in both texts.
2. **Distance.** Two primitives in a given text fall within a predefined window of primitives and in the corresponding text, the other primitive pair fall within the same window of length.

3. **Primitive.** Primitives in a primitive pair may be restricted to particular types to require more precise matching. For instance, subject-verb pairs may be restricted to particular types of subjects and verbs and matched to a pair in the corresponding text only if that pair shares the same subject-verb features.

Together, these features can be viewed as “poor man’s syntax,” capturing word order and rudimentary constituent structure. Full syntactic analysis is not required, but argument structure is captured indirectly. Hatzivassiloglou et al. (1999) combine the features in a vector for a pair of short paragraphs and feed the vector to a classifier trained on labeled vectors from a corpus of paragraph pairs. When compared against simple information retrieval techniques for measuring text similarity, using vectors based on tfidf scores, the approach using primitive and composite features far outperformed the traditional information retrieval techniques. However, the performance was still relatively low, with a recall of 36.6% and precision of 60.5%, possibly reflecting the difficulty of the task.

In their analysis of performance, the researchers also compared contributions of the different features. They found that the primitive features of word overlap, noun overlap and simple noun phrase overlap performed the best independently of other features, but that combining features also proved useful in identifying similar texts. For instance, verb overlap alone was not a particularly good indicator of similarity, but when combined with the composite feature of distance, captured verb-argument and verb-collocation relations, which were highly useful in identifying text similarity. This suggests that a range of features might be useful in text comparison, but not necessarily independently.
This approach to paraphrase recognition also illustrates how the lines between tasks have been blurred and how techniques from one task may be adapted to another. The primitive and composite features from Hatzivassiloglou et al. (1999) were incorporated into a MT evaluation approach by Russo-Lassner et al. (2005). They found that an approach that used the paraphrase recognition features for MT evaluation correlated better with human judgments of translations than measurements such as METEOR or BLEU alone.

4.4.2 Applicability to an ICALL task

While the approaches outlined in the previous section rely on training corpora that are typically not available in a language learning context, the techniques applied to the corpora are still relevant here. The paraphrase recognition task is perhaps more closely related to ICALL content assessment than any other task discussed in this chapter. The length of texts is comparable, as is the potential range of expected variation. However, the key difference is in the purpose of comparison. ICALL content assessment goes beyond paraphrase detection in that once a pair of texts is determined to be dissimilar (i.e., not paraphrases), then diagnosis is required to provide feedback to the learner. One goal in developing content assessment using the types of features discussed in the preceding sections is to select features that may be used both in detecting similarities and informing the diagnosis of dissimilarities.

4.5 Summary of Design Insights

The different goals, not to mention the different test sets and evaluation metrics, for the applications outlined above make direct comparison of many of the processing approaches described to this point impossible. But the common task of comparing texts
and measuring similarity is reflected in the overlap of techniques used across applications. At times, whole approaches have been adopted for other applications, as in the case of using the paraphrase recognition approach from Hatzivassiloglou et al. (1999) in MT evaluation and information retrieval research, or using summarization evaluation ROUGE in MT evaluation research, or even using an MT evaluation approach for automatic grading (Pérez et al., 2005a).

Together, these approaches provide useful design pointers for developing an ICALL content assessment application. These include the following:

- Linguistic information relevant to comparing meaning can be captured using shallow processing strategies.
- Features used to compare meaning may be more or less effective depending on other features and the underlying texts.
- A modular structure for comparison allows for easier inclusion and expansion of comparison features.

Shallow processing strategies for analysis of meaning are important in an ICALL domain for at least two reasons. First, deep processing strategies, which build a detailed, formal representation of meaning before comparing texts, require linguistic and world knowledge resources that are time- and labor-intensive to construct. They also tend to be restricted to limited domains. That is, as seen in the Herr Komissar system in Chapter 2, ICALL content assessment using deep analysis of meaning is restricted to handling a limited set of activities on restricted topics and are not easily adapted to new languages or expanded to handle more expressions.

Second, it is no secret that shallow NLP processing approaches tend to be more efficient and robust than deep NLP processing (Crysmann et al., 2002). Given that the responses provided by language learners may contain ill-formed language, robust shallow processing strategies seem more appropriate for analysis.
Thus, the shallow processing strategies will be used to identify different aspects or linguistic features of texts in order to compare the meaning of those texts. But as suggested by the work of Mani and Bloedorn (1997) and Hatzivassiloglou et al. (1999), the importance of a feature used to compare meaning is relative. Some features work best alone or in combination with other features. And different comparison features may work best for different texts, or may be more or less reliable than other features. Comparisons that have proved effective in the tasks described above include comparison of words, lemmas, synonyms, noun phrases, and argument relations. Still, whether these features are useful in ICALL content assessment is to be determined.

How processing for content assessment is organized is a relevant consideration as well. For example, the METEOR system incorporates a modular approach supporting concept matching that is flexibly applied to tokens, stems and synonyms. Transferred to the ICALL domain, such a modular architecture allows the system to be flexible to the needs of particular activities, it readily supports extensions with modules targeting specific foreign language teaching or language-specific needs, and it lends itself to the evaluation of individual component contributions.

However, as stated at the outset of this section, there are clear differences between the tasks described above and ICALL content assessment. For instance, METEOR provides a similarity score that rewards translations that are as close as possible to the reference translation in terms of both lexical choice and word order. In contrast, for content assessment in ICALL, one typically will want to accept syntactic and lexical variation as long as the targeted meaning is conveyed. Another important difference is that the output of comparing the translation with the reference in MT evaluation typically is a similarity score, which then can be compared to the score obtained for another translation to determine which is closer to the reference translation. Thus, scoring is the primary goal of MT evaluation, as it is for paraphrase
recognition, summarization evaluation, and even question answering. ICALL content assessment, on the other hand, must go beyond assigning a global score and instead provide a diagnosis, ideally of the type and location of any errors, for it to be useful in formulating specific learner feedback.

More will be said about diagnosis in subsequent chapters, but the scoring used in the tasks above provide a kind of summative evaluation. For ICALL such evaluation is helpful for giving a general sense of the learner’s overall progress. When the learner is getting most responses right, a scoring approach works well. But when the student get responses wrong, the feedback will be less helpful because a global score does not provide sufficient information to guide the student to a more appropriate response. The design of CAM described in the next chapter seeks to build on the techniques described here to use the linguistic information collected in a way that can offer relevant diagnosis information for the learner.
CHAPTER 5

CAM DESIGN

This chapter describes the design and initial implementation of an ICALL Content Assessment Module, henceforth CAM. This design targets activities in the viable processing ground of the exercise spectrum, as described in Chapter 3. Note that while an implementation of the design of CAM might be used as a stand-alone system, it is intended as a plug-in component to a complete ICALL system. Thus, many system architecture issues relating to activity selection, organization and presentation; learner modeling; and grammar assessment, among others, are orthogonal to this thesis and will not be discussed except where such issues impact processing learner responses for content assessment and meaning error diagnosis. For a description of the kind of complete ICALL system the CAM design is intended for, refer to Amaral (2007).

Also note that a more in-depth discussion of the data used to evaluate CAM is deferred to Chapter 6. However, it is important to mention here that the development of CAM was heavily guided by the analysis of actual learner responses to language exercises. A corpus of 566 responses to loosely restricted reading comprehension questions were collected for developing and testing the CAM design. All responses came from intermediate English as a Second Language (ESL) students at The Ohio State University.

Responses collected in Autumn 2006 were used as the development data for CAM. These data consisted of 311 responses from 11 students to 47 loosely restricted
reading comprehension questions. All response pairs, examples and findings referred to in this chapter come from this development set, except where explicitly indicated. The remaining pairs in the corpus were used as the test set. Again, refer to Chapter 6 for corpus details.

5.1 Architecture Overview

The architecture of CAM is depicted in Figure 5.1. Simply, for activities that have one or more correct answers, this CAM design compares the learner response to a stored answer (i.e., the target response) and decides whether the two responses are possibly different realizations of the same semantic content. Inspired by the techniques described in Chapter 4, the design relies on a series of increasingly complex comparison modules to “align” or match compatible concepts. Aligned and unaligned concepts are used to diagnose content errors.

Input to the module consists of a learner response and information from the activity model of the exercise. Following the discussion of language exercises in Chapter 3, the design of CAM assumes the activity model contains, minimally, a target response for comparison with the learner response. Other optional components of the activity model include

- the text of the question cue presented to the learner,
- a list of keywords, and
- a pointer to additional source text associated with the activity (i.e., the reading passages that question cues were based on).
Figure 5.1: CAM Architecture Overview
How the optional inputs are used by CAM will be discussed in subsequent sections, as will each of the other architecture components. However, in brief, the comparison of an input pair proceeds first with an Analysis Filter, which determines whether linguistic analysis is required for diagnosis. Then, for any learner-target response pair that requires linguistic analysis, CAM assessment proceeds in three phases – Annotation, Alignment and Diagnosis. The Annotation phase uses Natural Language Processing (NLP) tools to enrich the learner and target responses, as well as the question text (if available), with linguistic information, such as lemmas and part-of-speech tags. Alignment maps relevant concepts in the learner response to concepts in the target response using the annotated information. And Diagnosis analyzes the alignment to determine whether the learner response contains content errors. If multiple target responses are supplied, then each is compared to the learner response and the target response with the most matches is selected as the model used in diagnosis. The output is a diagnosis of the input pair, which might be used in a number of ways to provide feedback to the learner.

5.2 Analysis Filter

The first step of CAM processing is the Analysis Filter. The Analysis Filter is designed to handle learner responses that do not require linguistic analysis for content assessment. But it is the design of the activity that determines whether the filter is applied. For instance, if the activity were such that there were only one possible answer (i.e., no permitted linguistic variation), then linguistic analysis would be unnecessary. In such cases, if the learner response did not match the target response exactly, then it would be automatically wrong. Filtering these cases out is more efficient. Cached answers might also be used in the Analysis Filter. That is, if the
learner response has been seen, analyzed and evaluated before, there is no reason to process it again. Results from processing should then be cached for future processing.

The activity used here as a test case is loosely restricted reading comprehension questions that may have more than one acceptable answer, but the Analysis Filter exploits the nature of the language activity in a similar way. In theory, the learners could have constructed their responses using a wide range of linguistic expressions. However, in the corpus collected, this was not the case. Because the nature of the activity was such that learners had access to the reading passages while answering the questions, learners used lifting as a response strategy and often extracted sentences directly from the source text as their response. As a strategy, the lifting of answers directly from the source text is not surprising. Comprehension skills typically outpace production skills in language acquisition (Clark and Hecht, 1983; Davies, 1976), and students may use sentence extracts to avoid making production errors.

In the development set, 161 of 311 responses contained phrases or whole sentences extracted directly from the text. In other words, 52% of all submitted answers were copied all (or in part) from the text itself. However, of these only 104 (65%) were correct answers. This suggests that as a learner strategy, extracting answers directly from the text is only questionably successful. That is, learners often copied sentences that were incorrect or failed to recognize that the question could not be answered by a single sentence in the text.

Still, some of the reading comprehension questions could be answered by a single sentence in the underlying source text. For example, (45) is a question-answer pair from the development set in which the learner response, which is correct, is a sentence from the reading passage.
(45) a. **Question Cue**: What is the reason that broadcasters cut and paste?

b. **Learner response**: They say that if they are quoting someone who is not a very skilled speaker, they have to edit the talk.

Given the frequency that sentences were extracted as answers, whether right or wrong, it seems reasonable to exploit this phenomenon. To elaborate, a learner does not know in advance whether there is a sentence in the text that is a complete answer. However, in constructing their response the learner may use lifting (i.e., scanning and copying) as a strategy. Their response may be

- copied and correct,
- copied and incorrect (e.g., the learner copied the wrong sentence), or
- not copied but constructed, using a different strategy.

In processing, if the learner provides a response that matches the explicit answer sentence, then the response can be assessed as correct without using linguistic analysis. If the learner provides a response that matches some other sentence in the source text that is *not* the explicit answer sentence, then the response is automatically incorrect, and linguistic analysis is again unnecessary. Otherwise, the learner response may be correct, and linguistic analysis is required to make that determination.

To apply this analysis filter, CAM checks to see if a source text file has been provided. If not, this analysis filter is skipped because the question cannot be answered with a sentence from the text. If there is a source text file, CAM compares the learner response to the sentence from that file that is the correct answer using a bag-of-words methods. That is, the learner response and explicit answer sentence are converted to sets of lowercase tokens. If the relative set overlap, as defined in (46) is greater than a given threshold (85%), then learner response is judged as correct.

(46) Relative token overlap = \[
\frac{\text{Number of overlapping tokens}}{\text{Total tokens in explicit answer sentence}}
\]
Relative token overlap is used rather than token-based string matching because the learner may have inserted grammatical errors (e.g., misspelled or extra function words) into their responses that would not affect whether or not they conveyed the correct meaning, but would affect whether the string match succeeds. Relative overlap and a threshold accounts for these minor differences while requiring the response to be “similar enough” to the explicit answer sentence before the answer is labeled as correct.

If the learner response fails to match the explicit answer sentence, it is checked against every other sentence in that source text. If it matches one of those sentences (also using relative token overlap), this is considered negative evidence and the response is labeled as incorrect. If the learner response is not matched to either the explicit answer sentence or the incorrect text sentences, the filter labels the comparison as unresolved. In such cases, linguistic analysis is required. The text answer is added to the list of target responses for the exercise and linguistic annotation begins.

5.3 Linguistic Annotation

In the Annotation phase of comparing a learner response to a target response, CAM relies on a number of basic NLP components for tokenization, part-of-speech (POS) tagging, morphological analysis, dependency parsing and lexical semantic analysis. Table 5.1 contains an overview of the annotation tasks and the resources, tools or algorithms used. Keep in mind throughout this section that the design of CAM does not depend on the particular tool selected for each of the annotation tasks. The tools used here were chosen largely based on availability, though their performance on their respective tasks is noted when alternative tools were available.
<table>
<thead>
<tr>
<th><strong>Annotation Task</strong></th>
<th><strong>Language Processing Tool</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sentence Detection, Tokenization, Lemmatization</td>
<td>MontyLingua (Liu, 2004)</td>
</tr>
<tr>
<td>Lemmatization</td>
<td>PC-KIMMO (Antworth, 1993)</td>
</tr>
<tr>
<td>Spell Checking</td>
<td>Edit distance (Levenshtein, 1966), SCOWL word list (Atkinson, 2004)</td>
</tr>
<tr>
<td>Part-of-speech Tagging</td>
<td>TreeTagger (Schmid, 1994)</td>
</tr>
<tr>
<td>Noun Phrase Chunking</td>
<td>CASS (Abney, 1996)</td>
</tr>
<tr>
<td>Lexical Relations</td>
<td>WordNet (Miller, 1995)</td>
</tr>
<tr>
<td>Similarity Scores</td>
<td>PMI-IR (Turney, 2001; Mihalcea et al., 2006)</td>
</tr>
<tr>
<td>Dependency Relations</td>
<td>Stanford Parser (Klein and Manning, 2003)</td>
</tr>
</tbody>
</table>

Table 5.1: NLP Tools in CAM

5.3.1 Sentence Detection and Tokenization

Linguistic analysis starts with sentence boundary detection and tokenization. Sentence boundary detection separates a text string into individual sentences. Tokenization splits raw strings of characters within the sentences into individual tokens, which often correspond to words. Tokenization is a significant problem for languages with writing systems that do not use whitespace to delimit words, such as Chinese (cf., e.g., Lu, 2006). But even for a language such as English, the task is far from trivial. Tokenization is partly dependent on the particular application it is intended for. For example, strings such as *in spite of* or *New York City* arguably should be treated as single tokens consisting of multiple words when the application is a concept-matching algorithm as part of an ICALL system. On the other hand, a multiple-token analysis could be preferable for application domains such as text-to-speech systems. The status of hyphenated words (e.g., *Columbus-based* vs. *based in Columbus*), contractions
(e.g., *doesn’t* vs. *does not*) or abbreviations (e.g., *NYC, Mass.*) also provide interesting challenges. While such cases need to be considered, state-of-the-art tokenizers for English achieve 99.7% accuracy (Grefenstette and Tapanainen, 1994).

For content assessment, accurate tokenization is a critical first step of processing. As an example, consider learner-target response pair in (47).

(47) a. **Target Response**: They say that they have to edit if the speaker is not very skilled.

b. **Learner Response**: If they are quoting someone who isn’t a very skilled speaker, they have to edit the talk.

While (47b) is a perfectly acceptable variation of (47a), one difference is in the learner’s use of the contraction *isn’t*. If tokenization fails to properly handle the contraction, the concepts may not align correctly, resulting in falsely detected errors.

In the CAM implementation, the sentences and tokens are detected using the MontyLingua (Liu, 2004) tool suite. MontyLingua includes a tokenizer, lemmatizer, tagger and noun phrase chunker. The tokenizer uses regular expressions to

- recognize common abbreviations (e.g., Mr., U.S.),
- split possessive ’s as a separate token,
- preserve hyphenated words, and
- split and (optionally) expand contractions.

After adding new regular expression patterns to handle punctuation found in the development data, the MontyLingua tokenizer was evaluated against a sample of 100 responses to verify its suitability for the current task. Of the 2,377 tokens identified (702 unique token types), there were only 5 errors, for an accuracy of 99.8%. The five instances of errors identified were in two responses, and all were errors in processing bullet points in responses.
5.3.2 Morphological Analysis

Once character strings are split into tokens, variation in word forms can be handled through morphological analysis, which converts the tokens to lemmas, or stems. Consider the pair in (48).

(48) a. **Target Response:** Images can include beauty, wealth, an idyllic location or lifestyle, romance or popularity.

   b. **Learner Response:** Presents an image of beauty, wealth and idyllic location or lifestyle, romance or popularity, things that we as the audience are assumed to desire or want to be associated with.

The concept expressed by *images* in (48a) is expressed as *image* in (48b). And while token comparison fails to identify the connection, a morphological analysis of the sentence would reduce *images* and *image* to the same lemma so that later processing can align the concepts accordingly.

In CAM, the approach to morphological analysis relies on a combination of lemmatizers. Initial analysis is provided by PC-KIMMO (Antworth, 1993), which is distributed with a rule set for analyzing English derivational and inflectional morphology.\(^1\) Thus, PC-KIMMO recognizes that *advertises*, *advertising* and *advertisement* share a common root.

But while PC-KIMMO provides detailed morphological analysis for many English words, it is insufficient for lemmatizing in this context because it does not handle unknown words adequately. Misspelled words (e.g., *ralates*) and rare words (e.g., *bill-paying*) produce no analyses.

To supplement PC-KIMMO, the MontyLingua Lemmatizer is used to identify lemmas for inflected forms whenever PC-KIMMO fails to produce a useable output.

\(^1\)For formatting reasons, CAM uses the tool KText (McConnel, 2005) to access PC-KIMMO.
For example, it identifies *bill-pay* as the lemma of *bill-paying*, while PC-KIMMO provides no analysis at all. Once a token has been analyzed, the results are cached to speed processing when later instances of that token are encountered.

This combination of tools was evaluated against the 100-response sample from the development set. Table 5.2 lists examples of lemmas (some of which were not changed in the root form) identified in the sample.

<table>
<thead>
<tr>
<th>Token</th>
<th>Identified Lemma</th>
</tr>
</thead>
<tbody>
<tr>
<td>smaller</td>
<td>smaller</td>
</tr>
<tr>
<td>her</td>
<td>her</td>
</tr>
<tr>
<td>left</td>
<td>leave</td>
</tr>
<tr>
<td>bill-paying</td>
<td>bill-pay</td>
</tr>
<tr>
<td>ralates (spelling error)</td>
<td>ralate</td>
</tr>
</tbody>
</table>

Table 5.2: Lemmatizer Examples

Of the 2,377 tokens in the 100-response sample, the lemmatizer tools identified 412 tokens with base forms different from the token form. Of the 2,377 tokens/lemmas (702 types), only four were incorrect, for an accuracy of 99.8%.

5.3.3 POS Tagging

Part-of-speech (POS) tagging assigns lexical categories (i.e., tags) to tokens in the target and learner responses. POS tags are used in later CAM processing, such as chunking, so accurate POS tags are essential. Automatic part-of-speech taggers achieve around 97% accuracy (Brants, 2000) when trained and tested on English newspaper data. Performance depends heavily on the language and the specific part-of-speech tagset used. Taggers differ in the type of part-of-speech category distinctions made and whether tokens can reliably be disambiguated based on the distributional properties of the tokens in the training data.
To select an English tagger for CAM, the MontyLingua Tagger was compared to the TreeTagger system (Schmid, 1994), both of which are freely available. The MontyLingua Tagger uses the Penn Treebank tagset (Santorini, 1990), listed in Appendix C, with a model based on the Brill tagger (Brill, 1994). Briefly, the Brill tagger is an error-driven transformation-based tagger. That is, in training, the algorithm assigns an initial category to each input token based on frequency, then learns rules for assigning or changing one tag to another based on error reduction. These learned rules are applied to new text.

In evaluating the MontyLingua Tagger, the 2,377 tokens of the 100-response sample were tagged. Of these, there were 80 errors, for an overall accuracy of 96.3%. Most of the tagger’s errors came from mixing up nouns and adjectives, verb tenses, and verbs and nouns. Examples of the last of these include the ambiguous forms figures, quotes, appeals, presents, causes, plays, spy, size, use, link, and set.

TreeTagger tokenizes, lemmatizes and tags raw string inputs. The tokenizer and lemmatizer were also evaluated on the sample. In tokenization, TreeTagger has no awareness of abbreviations (e.g., U.S. is treated as two tokens) and does not expand contractions. Thus, in the sample, TreeTagger identified 2,386 tokens. Of these, there were 44 errors for an overall 98.2% accuracy, worse than the MontyLingua tokenizer. The lemmatizer, which lemmatizes inflected forms only, performed worse than MontyLingua as well, with 63 errors (97.4% accuracy).

However, in POS tagging, TreeTagger outperformed the MontyLingua tagger. TreeTagger is another probabilistic POS tagger, for which the available model for English was again trained on the Penn Treebank corpus. TreeTagger uses a slightly different set of tags than the Penn Tagset, but the output was mapped onto the Penn tags in order to compare results to MontyLingua.\(^2\) In the 2,386 tags, there

\(^2\)The mapping is presented in Appendix C.
were 59 errors, for an accuracy of 97.2%. Given the slight advantage in performance, CAM uses TreeTagger for tagging. And because other tools assume Penn tags, the TreeTagger output is mapped to Penn tags for later processing as well.

It is worth pointing out here that the Penn Treebank is largely well-formed newspaper text. Retraining a tagger on learner data might improve tagging results, though the performance on the sample set was more than satisfactory for illustrating the CAM design.

5.3.4 Spelling Correction

Most ICALL systems cannot ignore misspellings. Without the basic units of language, the system cannot “say anything” about the learner input. And the system cannot identify the basic units if they are misspelled.

The CAM implementation relies on minimum edit distance to identify non-word spelling errors. Minimum edit distance (Levenshtein, 1966) is the shortest number of operations (insertions, deletions or substitutions) required to convert one string of characters into another. For example, the minimum edit distance between flob and flood is 2 – one insertion of o and a substitution of d for b.

Here, the learner response is first scanned for non-words, using Spell Checking Oriented Word Lists (SCOWL) for American English (Atkinson, 2004). If any tokens in the learner response are not on the SCOWL list, they are compared to the target response tokens to identify the most likely intended word. Minimum edit distance is calculated between target tokens and the misspelled token. If the minimum edit distance is less than three, then the target token is considered a possible candidate, as long as both the target and misspelled tokens are of length greater than three.\footnote{This is because the overall number of incorrect spelling corrections due to short target tokens incorrectly suggested to replace short misspelled learner tokens would be too high otherwise.}

147
Using the target tokens to find a suitable replacement for non-word tokens in the learner response is a simple way to restrict the search space of suitable correctly spelled candidates. It eliminates the need to calculate the distance between each misspelled token and every word on the SCOWL, which would include many candidate replacement tokens that have a small edit distance to a given misspelled word but are inappropriate in context.

But inflection is also taken into consideration in suggesting candidate corrections. Sometimes comparing lemmas is necessary in resolving spelling problems. For example, the error *looving* might map to *loves*, though they differ by a minimum edit distance greater than 2 because of the difference in inflection. Thus, the spelling correction algorithm searches lemmas of target tokens, if a token comparison fails. Clearly, there will be cases when mapping tokens is preferred as well. For example, *campaing* should map to *campaign* with a minimum edit distance of 2. Yet the lemmatizer incorrectly applies inflection rules, reducing *campaing* to *campae*. This makes the edit distance 3. Thus, *campaign* would not be suggested as a candidate correction.

To avoid such problems, the CAM implementation makes token- and lemma-based searches for candidate corrections.

Note that the solution chosen here is incomplete. There is no guarantee that the learner misspelled a word on the target list. However, spelling correction for language learner mistakes is the subject of an entire thesis itself. Still, as with all the tools used in the Annotation phase, the design of CAM is not tied to this approach to spell checking. The spell checking algorithm and other tools have been chosen to illustrate the design, but may be readily replaced with better tools if available.
5.3.5 Synonym Identification

Because the learner may make different, but equivalent, lexical choices than those expressed in the target response, CAM uses WordNet (Miller, 1995) to identify potential synonyms. This can be critical for matching concepts as exemplified by the development pair in (49). Here, *influence* and *effects* should be recognized as equivalent concepts. But this requires synonym identification.

(49) a. **Target Response**: The *influence* of TV violence on child behavior is the main focus.
   b. **Learner Response**: The *effects* of television programs on behavior, especially the relationship of TV violence.

To identify synonyms, the lemmas and tags annotated prior to synonym identification are used to look up synsets in WordNet. Open-class words – nouns, verbs, adjectives, and adverbs – are the only terms included in the WordNet database. For each open class token in the learner and target responses, synsets are collected and stored as synonym lists.

5.3.6 Similarity Scoring

Similarity scoring differs from synonym identification in that it is used to identify terms that may be semantically similar, but not necessarily interchangeable (as synonyms are). For example, consider the pair in (50).

(50) a. **Target Response**: Alliteration is repetition of the *initial* sound in sequential words.
   b. **Learner Response**: where sequential words *begin* with the same letters or sounds

---

4Synsets represent concepts and contain words that may be used interchangeably in some context.
The verb *begin* in the learner response is intuitively similar to the adjective *initial*, but they would not be associated as synonyms. Similarity scoring is a way to capture relatedness between words.

Several theses could be written on the subject of measuring semantic similarity. The WordNet::Similarity suite of tools (Pedersen et al., 2004) alone includes several measures that rely on the WordNet hierarchy to generate similarity scores.

However, because the taxonomy is divided into noun, verb, adjective and adverb hierarchies and not all of the measurements allow comparisons across parts of speech, not all the measures are useful in this context. In addition, the results are not always straightforward to interpret. For example, consider the comparisons in Table 5.3, using three similarity measures from WordNet::Similarity.\(^5\)

<table>
<thead>
<tr>
<th>Word 1</th>
<th>Word 2</th>
<th>HSO</th>
<th>Lesk</th>
<th>Vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>lie</td>
<td>prevarication</td>
<td>16</td>
<td>304</td>
<td>1</td>
</tr>
<tr>
<td>lie</td>
<td>prevaricate</td>
<td>5</td>
<td>17</td>
<td>0.4886</td>
</tr>
<tr>
<td>liar</td>
<td>prevaricate</td>
<td>0</td>
<td>6</td>
<td>0.6464</td>
</tr>
<tr>
<td>liar</td>
<td>prevaricator</td>
<td>16</td>
<td>164</td>
<td>1</td>
</tr>
<tr>
<td>liar</td>
<td>prevarication</td>
<td>0</td>
<td>7</td>
<td>0.5785</td>
</tr>
</tbody>
</table>

Table 5.3: Similarity Module Results

*HSO* is the Hirst and St. Onge (1998) measure, which uses lexical chains linking the sense and gloss information to measure similarity as the length of the path between two concepts. The maximum relatedness is 16. *Lesk* refers to the Banerjee and Pedersen (2003) measure based on the gloss overlap of two concepts. The higher the number, the more related the two concepts. And the *vector* measure (Patwardhan, 2003) calculates the distance between two vectors constructed from co-occurrence counts from a corpus. The highest similarity score for *vector* is 1.

\(^5\)The scores were generated using the web interface to WordNet::Similarity at [http://marimba.d.umn.edu/cgi-bin/similarity.cgi](http://marimba.d.umn.edu/cgi-bin/similarity.cgi) on January 25, 2008.
Intuitively, each pair of words in Table 5.3 should have a high similarity score, in spite of the part-of-speech change in some cases. But this is not the case with all pairs and all measures, although these measures seem to capture some intuition of the relatedness of the terms.

To address this issue of what the measures are measuring, Resnik and Diab (2000) studied the power of three types of similarity measures to predict human judgments (intuitions). The first is the class of similarity measures – like those mentioned above – that rely on taxonomic relationships and a semantic network. The second approach relies on distributional similarity identified through corpus analysis. The final approach is Lexical Conceptual Structure (LCS), a theoretical model of semantic structure, described in several works including Dorr et al. (1995).

Perhaps not surprisingly, Resnik and Diab (2000) found that the ability of a similarity measure to predict human similarity judgments on pairs of verbs depended heavily on the verb pairs chosen. LCS performed poorly because the pairs chosen eliminated distinctions in subcategorization frames, which LCS relies on in making semantic relatedness judgments. The performance of distributional similarity measures, on the other hand, depends on the corpus used to generate similarity statistics. Taxonomic measures outperformed both LCS and distributional similarity measures on the test set, but the pairs chosen were all of the same part of speech (verb), as required by taxonomic comparison. These results suggest that choosing the best measure is nontrivial.

Rather than wade any further into the morass of similarity measurement techniques, the approach to generating similarity scores chosen here is one used by Mihalcea et al. (2006). It was chosen because the paraphrase recognition task they used to test the similarity measures is very similar to the detection task used in the evaluation of the CAM design, and their results on that task were encouraging.
The Mihalcea et al. (2006) approach uses a simple method suggested by Turney (2001) to calculate the semantic similarity between two words using point-wise mutual information. Point-wise mutual information (PMI) calculates the probability of observing two words together and compares that value to the probabilities of observing each word alone. If the two words appear together often enough, then the PMI will be greater than if there is no association between the words at all.

Turney’s method relies on the Web to generate mutual information statistics for word pairs. Turney tested several query types using the AltaVista search engine and found that that queries using the NEAR operator, which returns hits based on whether one word in the pair is found near the other in a document, obtained the best results in a test to recognize synonym pairs used in the TOEFL test. On a set of 80 questions in which the system was given a word and four possible synonym choices and asked to select the most likely choice, Turney’s PMI-IR scoring method obtained 74% accuracy, 10% higher than the average final score for a nonnative English speaker on the same task.

Following Turney, Mihalcea et al. (2006) calculate PMI-IR as in (51a), which is approximated as (51b), where $w_1$ and $w_2$ are words and WebSize is approximated as $7 \times 10^{11}$.

\begin{equation}
\text{(51) a. PMI-IR}(w_1, w_2) = \log_2 \left( \frac{p(w_1 \& w_2)}{p(w_1)p(w_2)} \right)
\end{equation}

\begin{equation}
\text{b. PMI-IR}(w_1, w_2) = \log_2 \left( \frac{\text{hits}(w_1 \text{ NEAR } w_2) \times \frac{\text{WebSize}}{\text{hits}(w_1)\text{hits}(w_2)}}}{\text{hits}(w_1)\text{hits}(w_2)} \right)
\end{equation}

Mihalcea et al. (2006) combine PMI-IR scores for all word pairs of the same part of speech class in a sentence pair to calculate an overall similarity score for the sentence pair in order to determine whether the sentences in the pair are paraphrases of one another. In Mihalcea et al. (2006), the PMI-IR measure is compared to the similarity measures used in the WordNet::Similarity (Pedersen et al., 2004).
Comparing results for 1,725 test pairs of sentences, PMI-IR obtained 69.9% accuracy, outperforming individual WordNet::Similarity measures and an approach using LSA.\(^6\)

To generate hits using the NEAR operator, the CAM implementation accesses the search engine Exalead.\(^7\) The hits returned by Exalead are used to calculate the PMI-IR score, following the approach described above.\(^8\)

Similarity scores are generated for nouns, verbs, adjectives, adverbs in the learner and target response. To limit the number of comparisons the system has to make, comparisons are restricted to noun-noun (e.g., `house-home`), noun-verb (e.g., `speech-speak`), verb-verb (e.g., `speak-argue`), adjective-adjective (e.g., `soft-quiet`), adverb-adverb (e.g., `quietly-softly`), and adjective-adverb (e.g., `quiet-softly`).

### 5.3.7 Noun Phrase Identification

Annotation beyond the token level includes chunking to identify noun phrases (NPs). To select an NP chunker for CAM, three programs were evaluated: MontyLingua’s REChunker, the chunker in TreeTagger and the CASS partial parser (Abney, 1996). MontyLingua’s REChunker identifies noun, adjective and verb groups using simple regular expressions, but as with the other two tools, only NP identification is evaluated here. As with MontyLingua, TreeTagger provides a simple chunking program, trained on NP chunks identified in the Penn Treebank. The third chunker, CASS, identifies base noun phrases, as well as larger noun phrase groupings (though the latter were not evaluated). CASS is a pipeline of finite-state automata that recognize patterns of noun phrases based on tag information.

\(^6\)When the WordNet::Similarity measures were combined, the combination performed slightly better (70.3%). However, such a combination could only be applied to words of the same part of speech since not all the measures compare words across POS categories.

\(^7\)As of July 2007, Exalead was available at http://www.exalead.com/search. AltaVista was not used because it no longer appears to support the NEAR operator.

\(^8\)To speed processing, the final scores are stored for reuse if that word pair is seen again.
<table>
<thead>
<tr>
<th>Chunker</th>
<th>Identified NPs</th>
<th>Correct</th>
<th>Missed</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>MontyLingua</td>
<td>611</td>
<td>458</td>
<td>182</td>
<td>75.0%</td>
<td>71.6%</td>
</tr>
<tr>
<td>TreeTagger</td>
<td>670</td>
<td>500</td>
<td>140</td>
<td>74.6%</td>
<td>78.1%</td>
</tr>
<tr>
<td>CASS</td>
<td>646</td>
<td>554</td>
<td>86</td>
<td>85.8%</td>
<td>86.6%</td>
</tr>
</tbody>
</table>

Table 5.4: NP Chunker Comparison

In the 100-response development sample, there were 640 base noun phrases (i.e., noun phrases not containing other noun phrases). The identification performance for each of the three tools is listed in Table 5.4. Given that CASS outperformed the other tools, it was selected for CAM.

5.3.8 Semantic Type Identification

Chapter 4 briefly mentioned the use of answer types and named entity (NE) recognition in question answering (QA) systems. In those systems, answer types are used to map questions to potential answers. Their answer type hierarchies were developed with specific kinds of questions in mind. A sample of these questions from the 2004 TREC question answering test set are listed in (52).9

(52) a. Which cities have Crip gangs?
    b. When was James Dean born?
    c. What does AARP stand for?
    d. Who founded the Black Panthers organization?
    e. What Las Vegas hotel was made famous by the Rat Pack?

The questions in (52) ask for different answers types— a list of places, a date, the expansion of an abbreviation, a person and the name of a hotel, but most are

---

proper nouns. Similarly, many of the lexical terms in the question string itself are proper nouns (e.g., *Black Panthers, James Dean, Rat Pack*). Such terms may not be found in a general-purpose dictionary, and NE recognition software is useful for labeling them with a semantic type, such as PERSON, ORGANIZATION, CITY, etc. In turn, these may be used in question classification patterns to identify the semantic answer type of the question (e.g., PERSON, CITY, YEAR).

However, in analyzing the development set questions and answers to development semantic types, it became apparent that basic NE recognition software has limited usefulness for the specific questions in the current study, primarily because there are simply not a lot of proper nouns in the data. Of the 311 learner responses in the development set, there were only 15 instances of six named entities that were not available in WordNet. These named entities were all of type PERSON and included *Bob Hope, Al Gore, Clinton, Lewinsky, Bell,* and *Gary Hart.*\(^\text{10}\) Target responses and question text had a similar small number of named entities, all of type PERSON.

The answer types for the questions as a whole also show a lack of simple named entities (i.e., PERSON, ORGANIZATION, LOCATION) as types. The breakdown of answer types for the 47 questions in the development set are listed in Table 5.5.

As this table suggests, most of the questions asked for reasons, definitions, ways of doing something, etc., and few of the answers required proper nouns. Thus, rather than using NE software to assign semantic classes to tokens, WordNet was used as a starting point for semantic type identification for common nouns in the learner and target responses. What few proper nouns there were in the data were manually added to a list of named entities with their semantic types.\(^\text{11}\)

\(^\text{10}\)Other proper nouns in the data that could be found in WordNet included *U.S. Congressman, Paris, American* and *Internet.*

\(^\text{11}\)Future versions of CAM should include NE recognition. While labeling proper nouns with semantic types has limited usefulness for the current data, it may be more useful for a larger corpus.
However, because one eventual goal is to classify questions with answer types and develop patterns for those answer types (refer to Chapter 7 for details), the starting point for the list of semantic type are a set of types adapted from Li and Roth (2002) in their system to automatically learn question classifiers. Their answer types were developed from examining the 500 questions in the TREC 10 data set. Because the nature of the questions differ in the current study, these types were augmented to include semantic types for nouns found in the development set. The full set of types used are listed in Appendix C.

For the initial implementation, types are only associated with nouns. For each noun, the CAM implementation identifies the noun phrase the noun token belongs to and searches the manually constructed named entity list for that token or phrase. For example, if the token is *Hope* in the chunk ['Bob', 'Hope'], the named entity list is searched for *Hope* and *Bob Hope*.

<table>
<thead>
<tr>
<th>Type</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>PERSON</td>
<td>2</td>
</tr>
<tr>
<td>LOCATION</td>
<td>2</td>
</tr>
<tr>
<td>ENTITY</td>
<td>2</td>
</tr>
<tr>
<td>QUANTITY</td>
<td>2</td>
</tr>
<tr>
<td>DEFINITION</td>
<td>5</td>
</tr>
<tr>
<td>MEANS</td>
<td>10</td>
</tr>
<tr>
<td>REASON</td>
<td>3</td>
</tr>
<tr>
<td>PURPOSE</td>
<td>2</td>
</tr>
<tr>
<td>IDEA</td>
<td>10</td>
</tr>
<tr>
<td>ATTRIBUTE</td>
<td>4</td>
</tr>
<tr>
<td>EVENT</td>
<td>2</td>
</tr>
<tr>
<td>MAIN-IDEA</td>
<td>1</td>
</tr>
<tr>
<td>EXPLANATION</td>
<td>1</td>
</tr>
<tr>
<td>JUSTIFIED-YES-NO</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 5.5: Answer Types in the Development Data
If no match is found, CAM searches the type list. Each type has been seeded with at least one term that is of that type. For instance, the terms *food* and *nutrient* have been manually associated with the type FOOD. In addition, the type list caches previously identified lemmas and tokens with their associated types so that the more terms CAM sees, the fewer times it must search the entire WordNet hierarchy in subsequent searches.

If the token’s lemma has not been seen before by CAM and is not on the type list, then WordNet is searched. The procedure for searching WordNet is to collect all the hypernyms associated with the input lemma. These hypernyms are compared against the types list. For example, suppose the unseen lemma is *hamburger*. The hypernym list associated with this term in WordNet includes [*sandwich, snack_food, dish, nourishment, food*]. If any of these hypernym terms appear on the type list, then its type is associated with the new noun. Thus, hamburger is associated with the type FOOD.

Because words may be ambiguous, more than one type might be associated with a single lemma. In such cases, all types are collected in a list and stored for that token. For instance, *hamburger* may also be associated with the type SUBSTANCE. If no type can be found a default type of ENTITY is associated with the lemma.

5.3.9 Dependency Parsing

Liu and Gildea (2005) show that measures of evaluating machine translation that rely on overlap of tokens for comparison are improved by the inclusion of syntactic information. Specifically, they found improvement in the system’s grammaticality judgments. While grammaticality judgments are not the goal here, the hypothesis is that the inclusion of some degree of syntactic information can provide more accurate
comparisons between sentences. For example, syntactic analysis might be useful for recognizing equivalent linguistic expressions such as the development pair in (53).

(53) a. **Target Response:** He was eating breakfast at his home.

   b. **Learner Response:** He was at home and eating his breakfast.

(53a) and (53b) differ only in syntactic structure, not semantic content. An analysis that identifies dependencies between the words in each sentence would be useful in identifying similarities across sentences. For instance, the dependency triples for the sentences in (53) are given in Figure 5.2.\(^{12}\) These triples capture the facts that *breakfast* is the direct object of *eating* in both sentences and that *he* is the subject. The argument information is also essential for pronoun resolution (to be discussed).

<table>
<thead>
<tr>
<th>Target Response</th>
<th>Learner Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>He was eating breakfast at his home.</td>
<td>He was at home and eating his breakfast.</td>
</tr>
<tr>
<td>nsubj(eating, He)</td>
<td>nsubj(was, He)</td>
</tr>
<tr>
<td>aux(eating, was)</td>
<td>dep(was, at)</td>
</tr>
<tr>
<td>prep_at(eating, home)</td>
<td>dep(at, home)</td>
</tr>
<tr>
<td>poss(home, his)</td>
<td>conj_and(at, eating)</td>
</tr>
<tr>
<td>dobj(eating, breakfast)</td>
<td>poss(breakfast, his)</td>
</tr>
<tr>
<td></td>
<td>dobj(eating, breakfast)</td>
</tr>
</tbody>
</table>

Figure 5.2: Dependency Triples for Response Pair

However, while syntactic information might help in principle, accuracy in parsing is an issue. Depending on factors such as the language, the text genre, the coverage of the grammar used, and the nature of the representations evaluated e.g., syntactic bracketing as in PARSEVAL (Black et al., 1991) or labeled dependency triples (e.g.,

\(^{12}\)These triples are output from the Stanford Parser (de Marneffe et al., 2006b).
Lin, 1995) - the accuracy of parsing differs significantly. For instance, accuracy is above 90% in identifying the correct syntactic bracketing in English newspaper text (Petrov et al., 2006) but below 30% in a labeled dependency evaluation identifying dative and PP arguments of verbs in German newspaper text (Boyd, 2007). Moreover, these evaluations tend to be based on well-formed newspaper data, not on potentially ill-formed language learner data. The degree to which syntactic information can be useful when the input sentence is syntactically ill-formed is unclear.

CAM uses dependency parsing because in MT evaluation research, Fox (2002) found that dependency triples correlate more across translation pairs than constituency trees. This follows the intuition that dependencies between words should be preserved across translations because translations preserve meaning. If the comparison of two translations of the same text is parallel to the comparison of learner and target responses, then there should be matches of lexical dependencies across learner and target responses, assuming the learner is aiming to convey the information in the target response.

To identify dependency triples, CAM uses the Stanford Parser (de Marneffe et al., 2006b) to parse the target and learner responses. A Stanford parser triple consists of a grammatical relation and two lexical items that participate in that relation, as shown in the examples in Figure 5.2. The full set of relations recognized by the Stanford Parser are listed in Appendix C.

The Stanford Parser is a phrase-structure parser trained on the Penn Treebank. Because it is both fast and robust, the Stanford Parser is useful in the current context. It produces an output even given ill-formed input, such as the ungrammatical sentences in the development data. The parser extracts triples from the constituency trees using a set of patterns and rules that map constituents into grammatical relations.
de Marneffe et al. (2006b) argue that differences in grammatical relations and how dependents are defined make comparing the Stanford parser to other dependency parsers such as Minipar (Lin, 1998b) nontrivial. However, the Stanford Parser was used as the basis for a system comparing sentences for the second Recognizing Textual Entailment (RTE) challenge. That system outperformed all other systems in the confidence-weighted evaluation (de Marneffe et al., 2006a). Given the similarities of the RTE task to the current task, these results suggest the Stanford Parser is suitable here.

5.3.9.1 Lexical Head/Keyword Identification

It is worthwhile to consider exploiting the fact that some words are more important than others for conveying information. The relative importance of words is probably most often exemplified by the split between so-called content and function words. Thus, the function word *the* is typically considered less important to the meaning of most sentences than a content word such as *gravitation*.

Important words, henceforth called keywords, may be extracted from target responses in several ways. The keywords can be identified manually by the instructor as they develop target responses. Except for the time required to create the lists, this is the most straightforward approach. And perhaps who best to know which words are necessary to meaning than the creator of the target sentence?

But in the absence of manually identified keyword lists, there are several potential approaches to automatically identification of keywords. The first picks up on the content-function word split and, using part of speech labels, collects all content words (i.e., nouns, adjectives, adverbs, verbs) as keywords.

---

13 This is not to say that function words are not important in second language learning or at all to meaning. Clearly, mastery of function words is necessary for fluency, though not necessarily for the ability to convey information.
Another simple approach might be to collect lexical heads – those words (i) that control the function of the phrase they are contained in and (ii) that other words in the phrase depend on (Crystal, 1997). Lexical heads are frequently content words, though not always.

Other approaches to automatic keyword identification might involve choosing a statistical measurement of word importance based on the frequency a term appears in a corpus, or using answer type to identify which words in the target response are necessary for a correct response.

But as a prototype, CAM employs the simple method of collecting lexical heads identified by the Stanford Parser as keywords in the absence of a keyword list that may be optionally provided by the instructor. Negation words (e.g., no and not) are added to the keyword list if present in the target, as well numbers. More sophisticated techniques may be employed in future implementations of CAM.

5.3.10 Pronoun Resolution

Pérez et al. (2005b) studied the effect of anaphora resolution on content assessment for automatic grading. They compared four approaches to incorporating anaphora resolution in their grading system:

1. **NP Replacement.** Replace each anaphor with a unique NP referent (the first NP in the coreference chain) before comparing the learner response to stored answers,

2. **Set Replacement.** Replace each anaphor with a set of coreferents before comparing the learner response to stored answers,

3. **Pronoun-only Resolution.** Replace each pronominal instance of “it” with its most likely referent before comparison, and

4. **Answer Generation.** Using the coreference sets to generate alternate answers.
Of the four methods, the first three replace some or all detected anaphors with one or more referents before comparing learner responses and stored answers. In theory, this anaphor replacement reduces the variation and increases potential for the matches between responses. But Pérez et al. (2005b) found that none of the approaches significantly improved the system performance. And in the case of the Set Replacement approach, performance actually declined.

The fourth approach – using coreference sets to generate alternate answers – outperformed the other techniques and improved system performance overall, improving correlation up to 52%, depending on the test set.

This dependence of performance on the underlying test corpus is important for considering the impact of anaphora resolution on system performance. The test set used in the Pérez et al. (2005b) study are answers related to questions about computer science terminology. Thus, “it” is the most common pronominal form; personal pronouns are non-existent. It might be argued that the task is simpler because “it” is the only pronoun and the set of referents for any technical term in the computer science domain is perhaps smaller than for terms in less restricted domains.

However, resolving “it” is far from trivial, given the frequency of non-referential uses of “it” in general use.14 And it is not clear from the Pérez et al. (2005b) results how well the anaphora resolution component performs on texts independent of the automatic grading system.

14Boyd et al. (2005) found that in a subset of the British National Corpus, of 2,337 instances of “it”, 28% were non-referential.
Thus, it is difficult to draw any hard conclusions about the impact of anaphora resolution on i) their approach, ii) automatic grading systems, or iii) content assessment in ICALL. The assumption here is that anaphora resolution would benefit content assessment, but the approach taken focuses specifically on pronoun resolution as a first step.

In this CAM implementation, pronoun resolution associates referents with pronouns in the learner or target response. Because most of the pronouns in the development set resolve to a referent in the question text, pronoun resolution is applied during question processing, though candidate referents are not restricted to noun phrases in the question text. The simple approach to pronoun resolution adopted here is based on the Resolution of Anaphora Procedure (RAP) algorithm (Lappin and Leass, 1994).

To start, the algorithm collects a list of all pronouns to be resolved. In the case of “it”, simple checks provide a rough filter for determining whether “it” is referential or pleonastic. For example, if “it” is followed by a form of be and then a modal adjective such as certain or necessary (e.g., It is necessary to go.), then it is flagged as pleonastic. Similarly, if “it” is followed by be and a cognitive verb such as assume (e.g., It was assumed he would go.), the pronoun is also assumed to be pleonastic.

After instances of pleonastic “it” have been removed, the algorithm collects a set of possible NP referents for each remaining pronoun in a response. The initial set of candidate referents is extracted from the response itself along with the question text. This initial candidate set is filtered to remove referents that meet certain syntactic conditions. For example, if the pronoun and candidate NP referent share the same head, then the NP is removed from the candidate referent list. Refer to Lappin and Leass (1994) for the full set of syntactic filters, but note that the current
implementation ignores the agreement filter. The agreement filter rules out refer-
ents based on person, number and gender. But this does not take into consideration
that language learners often make errors in pronoun agreement. Hence, this filter is
not applied. After the remaining syntactic filters have been applied, any surviving
candidates are scored using the points value in Table 5.6.

<table>
<thead>
<tr>
<th>If the referent is...</th>
<th>Add...</th>
</tr>
</thead>
<tbody>
<tr>
<td>a surface subject</td>
<td>80</td>
</tr>
<tr>
<td>an underlying subject</td>
<td>70</td>
</tr>
<tr>
<td>a direct object</td>
<td>50</td>
</tr>
<tr>
<td>an indirect object</td>
<td>40</td>
</tr>
</tbody>
</table>

Table 5.6: Pronoun Resolution Scoring Rubric

The grammatical features used scoring are identified using the dependency
triples collected in the previous annotation. Sentence recency is taken into considera-
tion by adding 20 to the scores of candidate referents contained in the sentence most
immediately preceding the pronoun being resolved. After scoring each pronoun, the
highest scoring candidate is selected as the most likely referent for the given pronoun.

5.3.11 Reliability and Availability of Processing Tools

In the context of language learning, two issues that merit further discussion with
respect to using the technology are reliability and availability. First, the use of NLP
tools for ICALL applications raises an important question: how reliable are they?
As touched on with the various technologies described above, the reality is that the
reliability of NLP components depends on factors such as the nature of the language,
domain, task and specific implementation. Performance can vary with each of these
factors, as the accuracy numbers reported above suggest. And those reported num-
bbers reflect the performance of the technology under relatively ideal conditions. The
models were developed and tested on English newspaper data, which are typically edited several times for grammaticality and content. The technology in question was not designed with learner input in mind. Language learner data are likely to contain far more errors, which is a real concern for ICALL systems.

Still, while how well each component will perform on learner input remains to be determined, the prospect of using imperfect technology is not necessarily grim. There are two mitigating considerations. One is that human performance on these tasks is often not 100%. In a teaching context, instructors do make errors in grading. If human graders do not perform at 100%, should the machine? It likely depends on how the output of the tools is to be used.

Another consideration is that the types of errors each technology makes are not evenly distributed over all cases that technology must handle. In other words, there are difficult or ambiguous cases for tokenization, tagging, parsing, etc. But these cases can be identified, isolated and handled accordingly. Good activity design can help ICALL systems avoid those hard cases in which the technology is likely to fail. And application of the most reliable technology first, whenever possible, can lessen the impact of unreliable technology. Still, it remains to be determined what level of reliability can be expected with NLP tools and whether this is sufficient for effective content assessment in ICALL.

The second issue with respect to using NLP tools for content assessment is one of general availability. English NLP tools are available for each of the above processing areas. This is not the case for all, or even most, languages. The model presented is based on English because the tools are available, but it is not intended to be a language-specific design for meaning assessment. Yet, the widespread applicability of the design relies heavily on the availability of the individual tools.
5.3.12 Pre-Alignment Filters

After linguistic information has been added to the target and learner responses, CAM moves into the Alignment phase. The Annotation phase produces a list of positions for CAM to attempt to align, but before alignment proceeds, the system runs two filters to edit the position list.

First, positions corresponding to punctuation are removed from consideration. Essentially, punctuation is ignored. Presumably, the components of the ICALL system that handle form assessment might handle errors in punctuation. The second filter uses the question text to identify given information. Consider the following question cue and associated target-learner response pair in (54).

(54) a. **Cue:** What was the major moral question raised by the Clinton incident?

b. **Target Response:** The moral question raised by the Clinton incident was whether a politician’s person life is relevant to their job performance.

c. **Learner Response:** A basic question for the media is whether a politician’s personal life is relevant to his or her performance in the job.

This example, which is representative of many of the questions and responses in the development set, illustrates an important property of the target and learner responses in the data collected, namely, the reuse of linguistic material from the question cue in the formation of a response.\(^{15}\) Target and/or learner responses borrow heavily from the text of the question. In pragmatic terms, this borrowed material can be thought of as *given* or *focus* material (Clark and Haviland, 1977; Sgall et al., 1986, among many others). This material restates the shared information and sets up the *new* information to follow.

\(^{15}\)The preponderance of such repetition in the data is likely due, in part, to the structure of the reading comprehension activity model, which encouraged learners to respond in complete sentences.
For processing purposes, this material is essentially extraneous. Learner responses that do not contain focus material represented in the target responses are not a priori incorrect. In the above example, (54c) is a correct response, even though it expresses the focus as *a basic question for the media*, rather than *the moral question raised by the Clinton incident*. Such learner responses should not be penalized for failing to contain identical focus material as the target.

Because responses should not be penalized or rewarded for the presence or absence of given text, this information is filtered out using the question text. To remove given information, the question text is tokenized, lemmatized, tagged and parsed along with the learner and target responses. Lemmas in the responses are searched for lemmas that appear in the question text. The positions corresponding to the lemmas in the question text that also appear in the learner response are removed from the list of positions that Alignment will try to map. The same removal process is applied to the target response to remove lemmas overlapping with the question.

While in the current implementation of CAM, given information is filtered out, it is worth noting that focus material should not necessarily be discarded without examination because the material could be an indicator that the learner has understood the question and has attempted a response. In other words, evaluation of focus material might be used as an indicator of whether the student was on the right track. Consider the hypothetical responses in (55).

(55) a. **Target Response:** The moral question raised by the Clinton incident was whether a politician’s person life is relevant to their job performance.

b. **Learner Response 1:** The moral question raised by the Clinton incident was whether Clinton had an affair with an intern.

c. **Learner Response 2:** The scandal was whether Clinton had an affair with an intern.
Both (55a) and (55b) are incorrect response, but differ only in focus material. But (55a) is a better answer because it suggests that the learner may have understood the question (and simply answered it incorrectly), while (55b) gives no indication that the learner even understood what was being asked. Future implementations of CAM may take this into consideration for more sophisticated question text processing.

5.4 Alignment

In the CAM design, alignment refers to the comparison of linguistic units in the learner and target responses in order to match or map concepts across responses. It is important to note that this mapping process is concerned with aligning equivalent concepts as well as compatible concepts. In other words, the semantic content need not be identical across learner and target responses as long as the learner response conveys the concepts expressed in the target response.

“Linguistic units” are the tokens, lemmas, synonym sets, etc. provided by the annotation phase described in Section 5.3. These annotations are critical to alignment because simple string matching and token overlap are insufficient for comparing learner and target responses. To underscore this fact, consider that of the 311 response pairs in the development set, only one pair was stem, token or string identical. In other words, 310 learner responses varied from their respective target responses by at least one token. After removing all the function words, the number of stem, token or string identical pairs only increased to 9 of the 311. While the low number of identical pairs was not unexpected given the less-restricted activity type used as a test case, this nicely show that the simplest approach – string matching – used by most existing ICALL systems for content assessment, is grossly inadequate for assessing the kind of responses addressed in the current study.
5.4.1 Exploring an Overlap Strategy for Content Assessment

Beyond string matching, one of the simplest approaches to comparing a learner response to a target response for alignment and content assessment is to measure word overlap. That is, to count the number of words (or tokens) the learner and target response share in common and calculate the precision (i.e., the overlap divided by the total tokens in the learner response) or the recall (i.e., overlap divided by the total number of tokens in the target response). This recall or precision score (or a combined f-measure score) is compared to an empirically determined threshold. If the score is above the threshold, then the learner response is assumed to be correct.

In developing the CAM design, such an overlap strategy for error detection was explored to determine how effective this strategy could be. A recall score was used because while 57% of all responses in the development set had extraneous words, most of those concepts were not wrong. That is, the extra words in the learner responses usually did not need to be present for the answer to be right but did not reflect incorrect information. Thus, the precision score, which penalizes for extra material in the learner response, was less useful in this context than recall, which penalizes the response only if the learner failed to include all the concepts in the target response. In addition, overlap of concepts is defined here in terms of lemmas, rather than tokens, to improve the overlap of concepts while maintaining the simplicity of the approach.

Tables 5.7 and 5.8 summarize the development data comparison results with respect to the token overlap recall at a threshold of .30.\textsuperscript{16} The term \textit{non-answer responses} is used to refer to learner responses in which the learner was not even close to conveying the expected (correct) concepts. Non-answer responses are relevant because the assumption is that these responses are likely to have lower token overlap.

\textsuperscript{16}A .30 threshold was experimentally determined by selecting the threshold that accounted for the maximum number of non-answer responses and the fewest number of correct responses.
<table>
<thead>
<tr>
<th>Responses</th>
<th>Average Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Responses</td>
<td>0.41</td>
</tr>
<tr>
<td>Correct Responses</td>
<td>0.51</td>
</tr>
<tr>
<td>Non-answer Responses</td>
<td>0.20</td>
</tr>
</tbody>
</table>

Table 5.7: Lemma Overlap Comparison Data: Average Recall

<table>
<thead>
<tr>
<th>Responses</th>
<th>Percent Responses with Recall ≤ .30</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-answer Responses</td>
<td>0.78</td>
</tr>
<tr>
<td>All Incorrect Responses</td>
<td>0.62</td>
</tr>
<tr>
<td>Correct Responses</td>
<td>0.17</td>
</tr>
</tbody>
</table>

Table 5.8: Lemma Overlap Comparison Data: Recall Below Threshold

scores than correct answers or even responses with other types of meaning errors. For instance, missing concept errors are those in which the learner was on the right track, but provided insufficient information for a correct answer. One would expect responses missing concepts would still have a high overlap score.

As expected, correct responses had higher overlap on average than non-answer responses, .51 and .20, respectively. And 78% of the non-answer responses had a recall score of less than or equal to .30, while only 17% of correct responses had an overlap score of less than or equal to .30. This suggests that there is a relationship between lemma overlap and non-answer responses. The less overlap, the more likely the response is a non-answer. However, nearly 20% of all correct responses also had a low overlap score. This suggests that lemma overlap alone is insufficient as the only approach to detecting errors because diagnosing errors based on lemma overlap is likely to produce too many false positives to be reliable. Looking at recall overlap of just content words does not change these findings.
5.4.2 Alignment Beyond Word Overlap

In the CAM design, concepts are aligned on several levels – at the level of token, chunk and relation – using a series of alignment modules. Token-level alignment compares learner and target responses using individual tokens, while chunk-level alignment groups tokens into noun phrase constituent chunks before comparing them. At the relation level, responses are aligned using dependency triples. These levels of comparison were selected after analyzing the development set to identify which modules (i.e., which types of alignment) might prove useful in comparing responses.

To simplify this development set analysis, concepts were defined in terms of all tokens. Thus, words that serve a grammatical function (e.g., that) were included along with content words (e.g., walking). The development pairs were aligned by matching tokens, lemmas and synonyms, as in the basic design of the METEOR system (Banerjee and Lavie, 2005) described in Chapter 4. The unmatched concepts were then compared manually to determine (i) why they did not match and (ii) what type of technology would improve the likelihood of a match.

Table 5.9 describes the different types of failure to match and how many responses in the development data contained at least one instance of each kind of match failure.\footnote{The counts refer to numbers of responses, but do not add to 311 (the total number of pairs) because any given pair may have had multiple match failures for multiple reasons.}

As Table 5.9 suggests, learner and target responses contain a lot of linguistic material that does not need to be present in the learner response in order for the learner response to convey the correct answer. A failure to match target concepts might be acceptable because those concepts are extraneous or unacceptable because they are critical to conveying the intended meaning. A failure to match concepts in the learner response might be i) acceptable because the unmatched are not needed but
<table>
<thead>
<tr>
<th>Failure Type</th>
<th>Description</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equivalence Recognition Failure</td>
<td>Failed to match equivalent units of legitimate variation Ex: <em>highest</em> should map to <em>most</em></td>
<td>114</td>
</tr>
<tr>
<td>Learner Error</td>
<td>Learner response used incorrect concepts that CAM should <strong>not</strong> match.</td>
<td>128</td>
</tr>
<tr>
<td>Unneeded Target Concepts</td>
<td>Unmatched because target contained extra concepts not required for a correct learner response.</td>
<td>148</td>
</tr>
<tr>
<td>Unneeded Learner Concepts</td>
<td>Unmatched because learner response contained extra concepts, that were not incorrect, but not needed.</td>
<td>176</td>
</tr>
</tbody>
</table>

Table 5.9: Match Failure Types and Counts

not wrong, ii) unacceptable because they are wrong, or iii) unacceptable because the failure is due to a system error (i.e., the system failed to detect a legitimate match). Distinguishing between unmatched concepts that have no impact on assessment (i.e., the unneeded concepts identified in Table 5.9) and unmatched concepts that do affect the assessment is a fundamental problem for implementing any CAM design. More will be said in subsequent sections about how this CAM implementation deals with this issue.

Equivalence recognition failures, on the other hand, might be addressed through a wider range of CAM alignment modules. A follow-up analysis of this class of alignment failures in the development data was used to identify types of mappings that might be explored to reduce the alignment failures. Table 5.10 describes these alignment suggestions and the corresponding count of response pairs that contained at least once instance of mapping failure that might have benefitted from that alignment
module. Note that in Table 5.10 and all subsequent uses, the symbol ⇔ indicates that a word or phrase is (or could have been aligned) aligned to another word or phrase.

<table>
<thead>
<tr>
<th>Suggestions</th>
<th>Description</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference Mapping</td>
<td>Equivalent NP expressions (e.g., Congress ⇔ The U.S. Congress)</td>
<td>35</td>
</tr>
<tr>
<td>Pronoun Resolution</td>
<td>TR or LR uses a pronoun for an NP (e.g., he ⇔ Al Gore)</td>
<td>20</td>
</tr>
<tr>
<td>Cross-POS Mapping</td>
<td>Compatible meaning expressions with different POS (e.g., initial ⇔ begin)</td>
<td>15</td>
</tr>
<tr>
<td>Ellipsis Mapping</td>
<td>TR/LR omits concepts based on question (e.g., ⊘ is . . . ⇔ Alliteration is . . .)</td>
<td>13</td>
</tr>
<tr>
<td>TAM Mapping</td>
<td>Tense, Aspect or Mood Mapping (e.g., may be ⇔ is)</td>
<td>12</td>
</tr>
<tr>
<td>Co-Hyponym Mapping</td>
<td>Acceptable sister/similar terms in TR and LR (e.g., good ⇔ favorable)</td>
<td>12</td>
</tr>
<tr>
<td>Passives Mapping</td>
<td>Concept difference due to passive construction (e.g., is chosen ⇔ choose)</td>
<td>7</td>
</tr>
<tr>
<td>Spelling Correction</td>
<td>LR contains non-word within 1 edit distance of TR word (e.g., Politicians ⇔ politicians)</td>
<td>5</td>
</tr>
<tr>
<td>Entailment Recognition</td>
<td>TR concepts are entailed by LR expression (e.g., highest number ⇔ most)</td>
<td>5</td>
</tr>
<tr>
<td>Definition Mapping</td>
<td>TR uses a term; LR uses a definition (e.g., a famous person ⇔ celebrity)</td>
<td>4</td>
</tr>
<tr>
<td>Coordination Mapping</td>
<td>In-context equivalent coordinating words (e.g., 8 and older ⇔ 8 or older)</td>
<td>2</td>
</tr>
<tr>
<td>Negation Mapping</td>
<td>Alternative ways to express negation (e.g., did not check ⇔ failing to check)</td>
<td>2</td>
</tr>
<tr>
<td>Modifier Mapping</td>
<td>Equivalent modification in TR and LR (e.g., mistakes were random ⇔ random mistakes)</td>
<td>1</td>
</tr>
<tr>
<td>Preposition Mapping</td>
<td>In-context equivalent preposition use (e.g., in his home ⇔ at his house)</td>
<td>1</td>
</tr>
<tr>
<td>Article Mapping</td>
<td>In-context interchangeable article use (e.g., a speaker ⇔ the speaker)</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 5.10: Suggestions for Improving Concept Matches
It is important to underscore the fact that, if implemented, any given suggestion in Table 5.10 would not necessarily directly improve CAM’s ability to diagnose meaning errors. Rather, these suggestions would help reduce the number of equivalence recognition failures, as defined in Table 5.9. This reduction of recognition failures would improve CAM’s ability to correctly identify compatible concepts, which should, in turn, improve CAM’s ability to detect and diagnose errors. With this in mind, the suggestions in Table 5.10 were taken as a starting point to develop a series of alignment and processing modules for the CAM design and implementation. These alignment modules are presented in Table 5.11 and described in the following sections.

<table>
<thead>
<tr>
<th>Level</th>
<th>Alignment Type</th>
<th>Example Match</th>
</tr>
</thead>
<tbody>
<tr>
<td>Token</td>
<td>Token-identical</td>
<td>advertising ⇔ advertising</td>
</tr>
<tr>
<td></td>
<td>Lemma-resolved</td>
<td>advertisement ⇔ advertising</td>
</tr>
<tr>
<td></td>
<td>Spelling-resolved</td>
<td>camping ⇔ campaign</td>
</tr>
<tr>
<td></td>
<td>Synonym-resolved</td>
<td>TV ⇔ television</td>
</tr>
<tr>
<td></td>
<td>Similarity-resolved</td>
<td>initial ⇔ begin</td>
</tr>
<tr>
<td></td>
<td>Type-resolved</td>
<td>Gore\text{\textsc{person}} ⇔ Al\text{\textsc{person}}</td>
</tr>
<tr>
<td>Chunk</td>
<td>NP-resolved</td>
<td>a skilled speaker ⇔ a very skilled speaker</td>
</tr>
<tr>
<td>Relation</td>
<td>Triple-resolved</td>
<td>nsubjpass(eaten, grain) ⇔ dobj(eats, grain)</td>
</tr>
</tbody>
</table>

Table 5.11: CAM Mapping Modules

5.4.3 Token-Level Alignment

As the name suggests, token-level alignment attempts to map individual tokens using a range of linguistic features assigned during annotation. Each comparison submodule tries to match tokens using a different feature. One design choice to consider here is whether to have each submodule run over all tokens or only over those unmapped
by the previous submodule. The latter case is exemplified by the original design of
the METEOR system (Banerjee and Lavie, 2005), which applies token, lemma and
synonym alignment submodules, in that order. There, the lemma-resolved aligner
only attempts alignment of unmapped tokens, and synonym alignment only applies to
unmapped lemmas. Because subsequent modules only apply to unmapped positions,
the order of alignment matters. This has the advantage that processing may be faster.

However, there are cases in which this is less than ideal. For example, consider
the hypothetical pair in (56).

(56) a. **Target Response:** The little boy shouted at his crying sister.

b. **Learner Response:** The little boy cried out to his weeping sister.

If a lemma aligner is applied first, then, *crying* is mapped to *cried*, but *shouted*
and *weeping* remain unaligned after synonym matching is applied. If synonym match-
ing is applied first, then *shouted* and *cried*¹⁸ are mapped, as are *crying* and *weeping*.
While in this example, synonym mapping should be applied before lemma mapping
for the desired result, in a different example, the reverse could easily be true.

Thus, rather than limit which token pairs are “visible” to each alignment
submodule, each aligner examines all token combinations and suggests candidate
alignment matches. The order of submodules is irrelevant, but user-defined settings
can be used to control which aligners are applied to the pairs, depending on the
activity. Token-level comparisons include

- **Token-Identical Alignment**

  The simplest type of token-level candidate alignments suggested in the CAM
  implementation are token-identical alignments. That is, if a token in the learner
  response is identical to a token in the target response (ignoring capitalization),

¹⁸Or *cried out*, depending on tokenization.
a candidate alignment is suggested. This is perhaps the most reliable (i.e., least error-prone) comparison technique because there is a direct comparison of surface form without any linguistic abstraction beyond identifying token boundaries and eliminating capitalization.

- **Lemma-Resolved Alignment**
  Lemma-resolved alignment proceeds just as token-resolved alignment does, except the lemmas provided by PC-KIMMO and MontyLingua are used to suggest candidate mappings between positions in the learner and target responses. Thus, `sleep ⇔ sleeping` might be suggested as an alignment pair.

- **Spelling-Resolved Alignment**
  Spelling-resolved alignment maps non-word spelling errors in the learner response to the (possibly) intended terms in the target response. Alignments of this type are based on suggestion proffered by the minimum edit distance approach described in Section 5.3.4. Recall that if fewer than three operations (letter insertion, deletion, or substitution) are required to transform the non-word into a word in the target, then candidate alignment is suggested for that learner-target token pair. For example, the non-word `boliticians ⇔ politicians` might be suggested as an alignment pair because their edit distance is 1 (substitution of `p` for `b`).

- **Synonym-Resolved Alignment**
  Synonym-resolved candidate matches are suggested alignments between learner and target tokens based on whether the token’s lemma appears in the synonym list of a corresponding token. For instance, token `televisions` in a learner response has the lemma `television`. From WordNet, the set of synonyms include `[telecasting, TV, video, television receiver, television set, tv, tv set, idiot`
box, boob tube, telly, goggle box, television system]. If TV appears in the target response, the synonym-resolved mapping module will suggest the candidate alignment televisions ⇔ TV.

**Similarity-Resolved Alignment**

For alignment using similarity scores, scores are considered as evidence for candidate alignments for word pairs with particular parts of speech. Only nouns, verbs, adjectives and adverbs are considered. And for those, given a pair of words from the learner and target response, alignment is only possible for word pairs that are noun-verb, noun-noun, verb-verb, adjective-adjective, adverb-adverb or adjective-adverb pairs.

Each pair in the learner and target responses that meet POS criteria have a similarity score. These pairs are the initial candidate pair list. But the list is filtered to remove scores below an empirically determined threshold. Once these possible candidate mappings are filtered, the remaining candidates are ranked by their PMI-IR scores. And the highest scoring pairs are selected. For instance, information ⇔ facts is suggested as a candidate alignment pair.

**Type-Resolved Alignment**

In the initial CAM implementation, only nouns are involved in type-resolved alignment. For each noun, the Annotation phase has identified a set of types. In Alignment, these sets are compared. For each comparison, if the intersection of sets is non-empty, the token positions are considered as candidate alignments. For example, if the lemmas cat and dog are both associated with type ANIMAL, then cat ⇔ dog would be a candidate alignment.

Note that in this approach to type-resolved alignment, there is no guarantee that the types associated with the learner and target tokens reflect the
intended meaning. For example, if the target response were *John won the lottery again. He’s such a lucky dog*, *dog* would be a slang term for PERSON. Yet, this token would be associated with the learner token *cat* in the unrelated sentence *My cat chases mice* through the type ANIMAL. Essentially, the learner is given the benefit of the doubt that the terms they use can be related to target terms through common types. But the same can be said of the other alignment strategies because there is an underlying assumption that the learner is trying to convey the meaning expressed by the target response. This assumption will be revisited in later sections.

After each token-level alignment module runs over the pair, all token-level candidate alignments are stored with the type of aligner that generated it. The idea is to collect as much evidence as possible from the different alignment techniques before selecting the final alignments. The resulting list of candidates is then passed to the alignment selection component.

5.4.3.1 Alignment Selection

Alignment selection in the CAM design involves ranking alignment candidates and selecting final matches. The prototype of the CAM design selects the best alignment using a variant of the traditional marriage algorithm (Gale and Shapley, 1962). The traditional marriage algorithm (TMA), also called the stable marriage problem or the Gale-Shapley algorithm, is a well-studied solution to the problem identifying a set of pairings of two mutually exclusive lists of items, given that each item on each list has a potentially different preference for the pairing. The most common example used to explain the matching algorithm – and the one used here – is that of traditional

\[ A \text{ variant of this algorithm is used each year by the National Resident Matching Program (NRMP) to match medical school students to residency programs across the country.} \]
marriages. Given a set of men and women, each possessing a ranked list of who from the opposite sex they would be willing to marry, how can the men and women be paired up? While there are multiple ways a set of pairings might be calculated, a fundamental assumption of the Gale-Shapley algorithm is that the best set of pairings is one that is stable. That is, the final set of pairings is one in which no man wants to be married to a woman who also prefers him to the man she is with. In other words, no man or woman is able to “trade-up” for someone higher on their preference list.

In the first round of the algorithm, each man proposes to the highest-ranked woman on his list. Each woman who is asked can either reject the proposal outright or say “maybe.” If a man is not on her list, a woman rejects him. Otherwise, she says “maybe” only to the highest-ranked man on her list (of those who have proposed). All others are rejected in the round. If a man proposes to a woman and she says “maybe,” they are engaged.

In the second round, all rejected men propose to the next woman on their preference lists who have yet to reject them. Again, each woman says “maybe” only to the proposal from the proposer who is ranked highest on her list. If the woman is already engaged and a higher-ranked man on her list proposes, she breaks off the original engagement (i.e., rejects her current fiancé) and says “maybe” to the new suitor. If a man has asked everyone on his list and is rejected by all, he stops asking.

These rounds continue until there is a round in which no man is rejected (either because each man is matched or because all unmatched men have been rejected by every woman each wanted to ask). At this point, each couple is married, and all unmarried men and women remain single. For a concrete example of how the algorithm works using the marriage metaphor, refer to Appendix D.
Roth and Sotomayor (1990) prove several properties of this algorithm:

- There is at least one stable matching situation as described above.
- The proposer has the advantage. When the man proposes, the final stable matching will be male-optimal, female-pessimal. That is, every man will be matched with the highest-ranked woman possible and every woman will be paired with the lowest-ranked acceptable partner.
- The algorithm allows for incomplete preference lists (i.e., each man does not have to rank each woman and vice versa). Thus, men and women can prefer to stay single. But if there is one stable matching in which a person remains single, that person will remain single in all possible stable matchings.
- The algorithm assumes that preference lists are strict, even if incomplete. That is, of those ranked on any given list, there are no ties. If preferences are not strict, there is no guarantee a stable matching exists.

Clear parallels can be drawn between the current matching problem and the stable marriage problem, as outlined in Table 5.12. These parallels suggest TMA has useful applications for CAM token alignment selection, assuming reasonable rankings can be generated. The goal is to select final alignments (i.e., marriages) between tokens using their ranking lists. The lists are strictly ranked but may be incomplete. Tokens may prefer to “stay single” (i.e., not align with another token) if they cannot be successfully paired with any choice on their list. A token that has no candidate alignments (i.e., an empty ranked list) prefers staying single over any choice available.

For the initial CAM implementation, ranking is based on the type of mapping module used. Each type of alignment adds a weight to a final ranking score for each candidate. The initial assumption adopted here is that the more abstract the linguistic analysis used in the mapping technique, the less reliable that technique will be. Thus, token-identical alignment adds more to the final score than lemma-resolved
<table>
<thead>
<tr>
<th>Marriage Problem</th>
<th>Matching Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mutually exclusive sets of men and women</td>
<td>Mutually exclusive sets of target and learner tokens</td>
</tr>
<tr>
<td>Different numbers of men and women in the population</td>
<td>Different numbers of target and learner tokens in responses</td>
</tr>
<tr>
<td>Men and women produce ranked lists of marriage</td>
<td>Candidate alignments can be converted to ranked lists of</td>
</tr>
<tr>
<td>preferences</td>
<td>preferred matches</td>
</tr>
<tr>
<td>Men and women may prefer to stay single</td>
<td>Tokens may not have any candidates to align with</td>
</tr>
</tbody>
</table>

Table 5.12: Parallels Between Concept Matching and Marriage Problems

alignment, for instance. But future versions of CAM may take a more sophisticated approach. One way to modify the ranking might be to allow the instructor to predefine the importance (i.e., weight) of a type of candidate match in the activity model of an exercise. Another approach might be to apply a machine learning algorithm to the problem in order to learn automatically how to rank candidates.

Also in this implementation, the learner response tokens “do the proposing.” Thus, the final matching will favor their rankings over those of the target response. But whether this makes any real difference in the final diagnosis is unclear, given that most tokens in the development set had at most two candidates (and usually only one). This suggests that a simpler algorithm than TMA might be effective for selecting matches in the development set. But the other data sets may include pairs with more matching candidates than found in the development set.

One important ramification to discuss with respect to the choice of TMA as the selection algorithm is that TMA enforces one-to-one mapping. That is, any token may be mapped to at most one other token. However, Table 5.13 offers hypothetical examples of one-to-one, one-to-many, many-to-one and many-to-many mappings.
<table>
<thead>
<tr>
<th>Mapping Type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-to-One</td>
<td>doctor ⇔ physician</td>
</tr>
<tr>
<td>Many-to-One</td>
<td>hair stylist ⇔ hairdresser</td>
</tr>
<tr>
<td>One-to-Many</td>
<td>rocker ⇔ rocking chair</td>
</tr>
<tr>
<td>Many-to-Many</td>
<td>one half ⇔ fifty percent</td>
</tr>
</tbody>
</table>

Table 5.13: Mapping Types and Examples

A clarifying distinction to make here is that the matching described thus far is between tokens, not concepts. A single concept may consist of multiple tokens. While, ideally, mapping concepts is the goal, in practice tokens are mapped because they are easier to recognize. One simple way to improve the mapping between concepts via tokens is to improve tokenization to recognize a wider range of multi-word tokens. For instance, WordNet treats *hair stylist* as a multi-word token. As such, it is synonymous with terms such as *barber, hairdresser, hairstylist*, etc. However, if tokenized as two separate tokens, *hair* and *stylist*, the connection to its synonyms will be lost. Grouping words together as multi-word tokens would improve mappings of this sort.

Of course, perfect tokenization of multi-word tokens does little to improve the chance of mapping the many-to-many example of *one half* and *fifty percent* from Table 5.13, or numerous other cases in which a single concept is expressed in multiple tokens. As a first step toward handling some of these cases in the CAM implementation, after concepts are aligned at the token level, they are aligned again as chunk-level segments, beginning with noun phrases. This is discussed in more detail in Section 5.4.4.

5.4.3.2 Alternative Approaches to Token Alignment

The annotation component of the CAM design is essentially modular. So, other NLP tools can be plugged in to tokenize, tag, chunk, etc. learner responses. The
same is true for the alignment component. For instance, once the candidates have been identified using the submodules described above, a different matching algorithm might be chosen or developed to select the best alignments, perhaps one that focuses on maximizing the overall number of matches between learner and target response.

Another alternative to the presented approach to token alignment comes from machine translation alignment. The token-level alignment discussed here is what is referred to as *word alignment* for monolingual texts in the machine translation (MT) literature. Approaches to word alignment in the MT focus on statistical methods of alignment. That is, given a parallel corpus of aligned sentences in two languages, the probability that a word in one language will translate as a word (or words) in the other language can be estimated. Perhaps one of the most recognized statistical word aligners is Giza++ (Och and Ney, 2003), a freely available implementation of the IBM MT models (Brown et al., 1993). Lexical translation probabilities are iteratively re-estimated using a training corpus. And the resulting set of probabilities can be used to produce one-to-one and one-to-many (though not many-to-many or many-to-one) alignments between words in unseen data.

Unfortunately, developing a training corpus is a prohibitive bottleneck to test this for the current study. In order to calculate reliable probabilities for word alignments, a sufficiently large corpus of aligned sentences is required. For machine translation, acquiring such aligned corpora can be nontrivial, depending on the pair of languages. In the current context, the aligned corpora would need to be learner and target sentences in the same language. Given enough available learner essays and their corresponding target/model essays, such a corpus would not be impossible to develop. Thus, such an approach might be worth exploring in the future. However, one potential complication is that probabilities of translating learner words and particular target words might depend on the L1 of the learner. For instance, the probability of
“translating” light as right or liver as river is likely higher for English learners whose L1 is Japanese, as compared to native Spanish speakers. Whether these or similar language-specific tendencies would make a significant difference is unclear. But if so, it would make collecting enough data for an aligned corpus more problematic.

5.4.4 Chunk-Level Alignment

As suggested above, the implementation of chunk-level alignment augments the conservative one-to-one mapping of token-level alignment by allowing what is essentially many-to-many, one-to-many and many-to-one mappings. The initial prototype of CAM focuses on aligning noun phrase chunks. The potential usefulness of aligning noun chunks, rather than individual tokens, is illustrated by the target-learner response pair from the development set in (57).

(57) a. **Target Response:** This happened in the U.S. Congress.
    
    b. **Learner Response:** This happened in Congress.

    In token-resolved mapping, the target tokens the and U.S. remain unmapped when compared against the correct learner response in (57b). Even ignoring unmatched function words, this correct response could be incorrectly labeled as incorrect because a content word (U.S.) is missing. However, if the target and learner responses are chunked, the comparison chunks are as follows in (58).

(58) a. [This], [happened], [in], [the U.S. Congress]

    b. [This], [happened], [in], [Congress]

    Ideally, in chunk-resolved mapping, the U.S. Congress would map to Congress, as desired. However, chunk-resolved mapping comes with its own challenges. Consider the pair in (59).
a. **Target Response:** They say that they have to edit if the speaker is not very skilled.

b. **Learner Response:** If they are quoting someone who is not a skilled speaker, they have to edit the talk.

A chunker will identify *a skilled speaker* in the learner response as a single unit. But *skilled* in the target response is not contained in a noun phrase. Thus, in mapping, *skilled* remains incorrectly unmatched if one-to-one matching is enforced.

The pair in (60) is another example from the development data where incorrect chunking has a negative impact. In this example, in chunking of the learner response, 8 is chunked with *years*, but in the target response, the incorrect chunk is *children 8*. Because of this processing error, the mapping proceeds incorrectly and the learner response is marked as incorrect, even though the learner and target response differ in only two tokens, *years, set*.

(60) a. **Target Response:** Two-thirds of children 8 and older have a TV in their bedroom.

b. **Learner Response:** Two-thirds of children 8 years and older have a TV set in their bedroom.

These examples suggest that aligning chunks is not always desirable. The question is how to preserve the benefits of chunking while avoiding alignment errors caused by (i) ignoring relevant material within a chunk or (ii) relying on incorrect chunks. The solution explored here is to perform token-resolved alignment over all tokens. The mapped tokens then are used to provide evidence for aligning chunks. The number and type of token mappings from one chunk to another are used to score, rank and select the best chunk alignments. Essentially, candidate chunks are ranked...
using relative token overlap and final chunk alignments are selected using TMA. As with token alignments, all chunk alignments are passed to the Diagnosis component.

In addition, the chunk alignments can be used as evidence to adjust token-resolved alignments as necessary. The heuristic condition for determining whether alignments between tokens should be broken is given with an example in (61). Essentially, if two chunks are not mapped, but there is a token mapping within those chunks, the token alignment is broken.

(61) a. If $token_x$ in $NP_A$ maps to $token_y$ in $NP_B$, but $NP_A$ does not map to $NP_B$, then break the token alignment.

b. Example: The token alignment for $good$ is broken if the tokens appear in unaligned chunks, as in [The $good$ politician] $\neq$ [A $good$ house]

5.4.5 Relation-Level Alignment

The final type of alignment implemented in the CAM prototype is at the relation level. As the name suggests, relation-level alignment involves examining pairs of tokens in one response and aligning them to a pair in another response based on the relationships they share. Relations between concepts are important for conveying meaning, as is apparent in the hypothetical example in (62). The meaning of the two sentences in (62) are clearly different, but without a way to compare argument structure, a concept comparison would not capture the difference in meaning.

(62) a. **Hypothetical Target Response:** John called Mary.

b. **Hypothetical Learner Response:** Mary called John.

Still, some analysis other than full syntactic comparison is appealing in the current context. First, language learner responses may often be ungrammatical. In the development set, independent of whether the correct meaning was conveyed, 115
of 311 responses (37%) contained some error in form (i.e., syntactic, spelling or morphological error) that might affect parser performance. Syntactic parsers, including the Stanford Parser used to recognize dependency relations for CAM, are robust enough to provide some analysis of a range of linguistic expressions, including those that are ill-formed. However, it is unclear how reliable the analyses are. Given enough learner data, the parser might be retrained on ill-formed input to better handle the types of (ill-formed) sentences learners produce. Short of that, the structure assigned to learner sentences by “off-the-shelf” parsers should be treated with some wariness.

Second, it is not the intent of the CAM design to evaluate the ill-formedness of learner responses. That task is left to another component of a complete ICALL system. Ideally, this grammatical analysis component should be tied into CAM to aid in content assessment, but is not implemented here.

Third, the comparison of complete syntactic structures is most important when the learner response can be expected to be syntactically similar to the target response. In the current context, learner responses may be semantically very similar, while syntactically very dissimilar. Short of mapping sentence types, comparing sentence structures for these cases is not particularly helpful. What is of more interest from the perspective of comparing meaning is that the argument relations are compatible.

Thus, the relation-level alignment is based on mapping dependency triples collected from the output of the Stanford Parser. Because grammatical error may result in incorrect analyses, partial matches of triples are accepted along with full matches for candidates. Recall that a triple consists of a grammatical relation and the two tokens (a head and a dependent) that participate in that relation. The triples from the target and learner response are compared. A full match exists if the relations are the same, the heads are aligned through token-level alignment and the dependents
are also aligned through token-level alignment. Partial matches are suggested if two of the three components align. There are three types of partial triple matches:

- **Dependent-Relation Match** – A matched dependent pair of tokens participate in the same relation, but have different/unmatched heads. For example, in the sentences *Horses eat hay* and *Horses drink water*, the triples \( nsubj(eat,Horses) \) and \( nsubj(drink, Horses) \) are a Dependent-Relation match because only the head differs. For such partial matches, the learner may not have produced a content error. For example, suppose the sentences are *Farmers feed hay to horses* and *Horses eat hay provided by farmers*. The sentences convey very similar information, but the triples \( dobj(eat,hay) \) and \( dobj(feed, hay) \) are only partial matches. This underscores the fact that the local match of triples is only part of the information necessary for determining whether there exists a global match of the sentences containing those triples.

- **Head-Relation Match** – A matched head pair participate in the same relation, but have different unmatched dependents. This is parallel to the Dependent-Relation match and carries the same issues.

- **Head-Dependent Match** – Two matched pairs of heads and dependents differ only in the relationship they participate in. For example, consider the sentences and the relevant triples in (63).

\[
(63) \quad \begin{align*}
\text{a. Hypothetical Target Response:} & \quad \text{Chickens eat seeds.} \quad (nsubj(eat,chickens)) \\
\text{b. Hypothetical Learner Response:} & \quad \text{Seeds eat chickens.} \quad (dobj(eat,chickens)) \\
\text{c. Hypothetical Learner Response:} & \quad \text{Seeds are eaten by chickens.} \quad (nsubjpass(eat,chickens))
\end{align*}
\]

Comparing (63a) and (63b), the mismatch of grammatical relation is indicative of an error in the learner response; the arguments are reversed. But for
63a) and 63c), the mismatch is not an error, but legitimate variation. Knowing which relations have equivalents is crucial for distinguishing between these cases. In the prototype of CAM, equivalent relations are recognized for mapping passives, as in the case of 63a) and 63c). Future implementations will explore additional mappings of relations for more thorough syntactic comparison.

5.4.6 Alignment Issues

Once the Alignment phase is completed, the token-, chunk- and relation-level alignment facts are passed to the Diagnosis phase of processing. However, before describing final stage of processing, two alignment issues to underscore relevant to each of the alignment modules described above are false positive alignments and aligning ill-formed input. Assuming the NLP tools provide accurate information to the alignment modules, incorrect mappings can still occur. For example, token matching may incorrectly relate tokens as the hypothetical example in (64) suggests. If the two instances of *can* in (64) are aligned, it is a false positive – a match that is incorrectly made.

(64) a. **Hypothetical Target Response:** He opened a [can] of tuna.
    
    b. **Hypothetical Learner Response:** You can go now.

Removing morphological endings can also remove relevant meaning distinctions to produce false positives as in (65). If *fruitless* and *fruitful* are stemmed to *fruit* in (65), then the two words are incorrectly matched.

(65) a. **Hypothetical Target Response:** a [fruitful] discussion
    
    b. **Hypothetical Learner Response:** a [fruitless] discussion

Even alignments based on spelling errors can be nontrivial, as in (66). Both of the underlined target concepts are one edit distance from the misspelled *raed*. Thus,
again there is potential for misalignment. In fact, for each of the alignment types, there is always the potential for aligning the wrong tokens, chunks or dependency triples.

(66) a. **Hypothetical Target Response:** He read the red book.

   b. **Hypothetical Learner Response:** He raed the book.

A second alignment issue is the need for alignment even in the presence of ill-formed responses. Consider the example learner response from the development set in (67).

(67) **Learner Response:** Walk to a bar and says “Mak mine a double.”

   This example is missing a subject, contains an agreement error and has a spelling mistake. If the tagger mistags mak as a noun, then the chunker will fail to identify the correct NPs. The dependency parser will also produce an incorrect analysis if the tagging is off or the sentences is incomplete. As a result, candidate alignments may not always be reliable if based on incorrect analysis of ill-formed text.

   Both issues have an impact on alignment and, thus, diagnosis. But the hope is that collecting as much information as possible to pass to the Diagnosis component will lead to a preponderance of useful evidence for making reasonable judgments about whether a learner response contains legitimate content variation or a meaning error.

5.5 Diagnosis

The Alignment phase reports both matches and types of matches to the Diagnosis component. Matches are the collection of mappings from the learner to target responses that represent concepts and relations that the learner likely “got right.” The *unmapped* relations and concepts in the learner and target data structures indicate possible learner errors. The types of errors are indicated by the differences in the
structures. The Diagnosis phase is actually a combination of detection and diagnosis of errors. The output of this module is a final report about the comparison once diagnosis is complete. But before delving into the details of how the Diagnosis phase proceeds, several preliminary topics warrant discussion, beginning with how diagnosis for content assessment relates to notions of assessment in teaching.

5.5.1 Content Assessment in the Context of Learner Assessment

Assessment of learner responses can be broadly divided into two categories – formative and summative assessment (Bachman and Palmer, 1996). Summative assessment typically takes the form of tests. That is, the goal of summative assessment is to evaluate how well students have learned something after the period of instruction has ended (Boston, 2002). Such tests are graded and tend to be associated with high stakes.

Formative assessment, on the other hand, is defined by Black and Wiliam (1998, p. 7) as “all those activities undertaken by teachers, and/or by their students, which provide information to be used as feedback to modify the teaching and learning activities in which they are engaged”. In other words, formative assessment is used to guide teachers during the instruction period by determining students’ strengths and weaknesses. Thus, it may be ungraded and low-stakes.

A third, possibly distinct, type of assessment mentioned in the literature is diagnostic assessment. Diagnostic assessment is sometimes considered a subtype of formative assessment in that it is intended to identify learner strengths and weaknesses, specifically for remediation purposes (McMillan, 2001). Like formative assessment, it is often ungraded and low-stakes, and it is used to guide the teacher to make choices about instruction. However, if diagnostic assessment is used prior to instruction in order to ascertain what level of knowledge and skills students have before teaching
begins, then diagnostic assessment might be considered a separate type of assessment. Whether or not diagnostic assessment is part of formative assessment is not a crucial issue here. Thus, for the current purposes, it will be included as a subtype of formative assessment.

Content assessment can be useful for formative or summative language assessment, depending on the purpose of the larger ICALL system it is embedded in. It should be left to ICALL system designers, working closely with language instructors, to determine how ICALL systems should be integrated into language instruction. Different approaches to instruction might lead to different uses for ICALL systems and for content assessment.

While the current work remains essentially agnostic on how content assessment in ICALL should be ultimately used in a comprehensive assessment program for a course, it is still relevant here because diagnosis is intimately tied to feedback, and the nature of feedback is determined, in no small part, by the type and goals of assessment. For instance, if summative assessment is intended, then a score that indicates right or wrong might be sufficient as feedback to the learner.

Still, because one of the goals of this thesis is to explore the extent to which meaning errors can be pinpointed using shallow linguistic information, providing holistic scores as feedback in summative assessment is less useful in the current context for illustrative purposes. Moreover, the assignments gathered in data collection were originally used formatively by the instructor to identify areas of progress and improvement for students’ reading comprehension skills. Thus, diagnosis for formative assessment is the focus here. But even within the realm of formative assessment, a range of feedback strategies might be explored. Bangert-Drowns et al. (1991) argue that the most effective types of feedback on homework (or tests) are those that point out specific errors, offer targeted suggestions for improvement, and encourage the
learner to think about the task rather than the right answer. And again, this very important question of what types of feedback are most effective – what information to provide and how best to present that information to the learner – is simply beyond the scope of this thesis.

5.5.1.1 More on Feedback

But while the focus of the current research is on the NLP underpinnings of diagnosis for ICALL content assessment and not on the evaluation of different types of feedback, the question of feedback is still relevant to diagnosis. Two considerations with respect to selecting feedback are the effectiveness of feedback types given i) the activity and ii) the ICALL setting, and the ability of the ICALL system to provide accurate feedback of a given type.

A number of feedback types are available in response to learner language. According to an interactionist view of language learning (Long, 1996), interaction through such feedback and negotiation is essential to language acquisition. This negotiation of meaning is defined broadly by Pica (1994) as any interaction between the learner and other participants in a communication exchange that is triggered by a breakdown in comprehension. Essentially, once someone in the exchange fails to understand the message being delivered, the participants in the exchange may enter into a negotiation of meaning where the message is clarified, reworded, etc. to improve comprehension.

From an analysis of learner speech, Oliver (1998) identified three kinds of negotiation. These included

- **Comprehension Checks** – The speaker asks the listener to confirm whether the listener understood.
• **Confirmation Checks** – The listener indicates they are unsure of the previous utterance. They ask for confirmation by repeating part or all of what they heard as part of the confirmation check.

• **Clarification Checks** – The listener indicates they did not understand the speaker and need repetition or rephrasing of the previous utterance.

The effectiveness of feedback types in second-language instruction has been studied (and debated) in the literature. Pica (1994) points to several studies discussing negotiation in second language learning, as does Mackey and Abbuhl (2005), who also show that interaction and corrective feedback have a positive effect of language learning.

In another study, Panova and Lyster (2002) examined corrective feedback (i.e., feedback types intended to prompt the learner to improve their response) in spoken classroom work and found specifically that recasts (i.e., repeating the learner’s expression, with form errors corrected) are used most often by teachers, but that clarification requests (e.g., *I don’t understand*) and elicitation (i.e., explicitly asking for a re-do of the learner’s utterance) show better correlations with the learner’s immediate response to the feedback (i.e., uptake) and their self-repair efforts.

They also found that the feedback type may depend on the type of error. For instance, clarifications, while only used 11% of the time, was used more often when the errors in a learner’s speech affected how comprehensible the meaning was. Intuitively, teachers were more likely to make clarification requests when they did not understand what the learner meant.

Panova and Lyster point out that clarifications trigger self-repair from the learner, and that self-repair is particularly desirable because, following Swain and Lapkin (1995), it shows how the learner has i) noticed the error and ii) deliberately modified their production accordingly in a negotiation of meaning.
The Panova and Lyster (2002) study, while of adult ESL students, focuses on feedback to grammatical errors in spoken production. In fact, much of the research on feedback focuses on feedback in the classroom to oral communication activities (e.g., Van den Branden, 2000; Murphy, 2007). But the current study relates to meaning errors in written production in response to a comprehension activity outside of a traditional classroom setting. It is unclear how the effectiveness of different types of feedback used for form-based errors translate in such a setting.

However, Pica’s definition of negotiation of meaning relates to oral or written communication. Thus, negotiation of meaning, and its benefit to the learner, is not inherently restricted to oral communication. In a study not focused on spoken production, Van den Branden (2000) looked at negotiation of meaning in reading comprehension activities. Primary school students were given the opportunity to orally discuss any difficult sentences or vocabulary passages in pairs or with the whole class prior to taking a multiple choice reading comprehension test. As compared to students who were given either an unmodified text or a simplified text followed by the test without opportunity for negotiation of meaning, the students who were allowed to negotiate meaning had better comprehension scores on the test. This suggests that negotiation of meaning is beneficial for comprehension activities.

In the domain of computer-assisted learning (not specific to language instruction), feedback has been divided into three categories – Knowledge of Response (KOR), Knowledge of Correct Response (KCR) and elaborative feedback (Clariana, 2000). KOR feedback informs the learner whether their answer was right or wrong. KCR feedback informs the learner whether their answers were right or wrong and provides the correct answer. And elaborative feedback informs the learner whether their answer was right or wrong and provides additional information about the answer.
Clariana (2000) divides elaborative feedback into three categories:

- **Explanatory Feedback** – Additional information on why the learner response is correct or incorrect.
- **Directive Feedback** – A prompt or hint that guides the learner to the right answer.
- **Monitoring Feedback** – A report of the learner’s overall progress in the task.

Clariana (2000) notes that while some evidence shows that elaborative feedback is better than KCR and KOR feedback (e.g., Clark and Dwyer, 1998), he argues that it is also possible that feedback types work equally well (i.e., any feedback is better than no feedback) as long as it is immediate.

Murphy (2007) builds on the work relating to computer-assisted instruction, feedback types and comprehension activities by studying negotiation in reading comprehension activities in an CALL domain. His study focused on feedback types and comprehension for students working in pairs and alone. Murphy found that the difference in student performance after KCR and elaborative feedback was insignificant. That is, both forms of feedback were equally effective. Still, students who worked in pairs and interacted with each other and with the system outperformed the other test groups. Thus, group work interacted with feedback type. But one of the implications is that if the system is likely to be used by students working alone, KCR feedback may work just as well as elaborative feedback and is far easier to implement.

In the current implementation of the CAM design, the output of the processing is a code, corresponding to a diagnosis. This code might be tied to whatever the particular feedback approach is selected in a larger ICALL system that incorporates content assessment. However, the brief overview above of feedback types suggests the following for how feedback might be selected for content assessment:
• While the findings from Murphy (2007) – that KCR and elaborative feedback are equally effective – are interesting, the current situation differs in that the activity involves a written component; the Murphy study used multiple choice. Given that there is a production aspect involved, there is more opportunity to enter into an interaction with the learner that targets specific meaning errors. Thus, tying diagnosis codes to specific types of elaborative feedback, with KCR as a fallback strategy, might be particularly effective.

• Currently, the design of CAM assumes that the interaction/exchange between the system and the learner is limited to a single turn each. That is, the learner provides an answer and CAM provides diagnosis for that response. Ideally, once CAM responds with an analysis to a learner response containing a meaning error, this would trigger a longer interaction between the learner and system. The learner would have additional opportunities to try to correct themselves and CAM would provide an analysis for each attempt, tracking previous effort. This is not currently implemented in the CAM design, but using elaborative feedback strategies would be a first step toward the goal of facilitating self-repair.

• Because the CAM design focuses on content (meaning) assessment, the feedback it provides is not related to ill-formedness. A separate ICALL module would be responsible for evaluating and providing feedback to grammatical errors using a possibly different set of feedback strategies. Given that clarification and confirmation checks described by Oliver (1998) focus attention of the breakdown of understanding rather than on the grammatical form used, these types seem useful for feedback related to meaning-based activities. As forms of directive feedback, clarification and confirmation attempt to guide the learner to self-repair while indicating that the system has not understood the meaning the learner attempted to convey.
• The information collected by CAM could be converted into elaborative feedback – either directive or explanatory. Explanatory feedback seems less interactive in the sense that once the system explains why the answer is right or wrong, the exchange is finished. However, it might be used in combination with directive feedback to give the learner information about the error, while prompting the learner to retry the question. That is, feedback based on the diagnosis given by the CAM could rely on a mix of directive and explanatory phrases.

5.5.2 Meaning Errors

To this point the discussion has centered on either how to collect information for use in error diagnosis or how diagnosis (through feedback) fits into a larger assessment picture. But the latter presumes that there is a diagnosis, based on a set of errors to choose from in diagnosis. Before the module can diagnose errors, the issue of what kinds of errors can and should be identified must be considered. To that end, there are several ways to conceptualize language learner errors. For instance, James (1998) broadly divides what he calls the learner’s ignorance of a language into four types:

• **Grammaticality**: Whether the form is ill-formed, independent of context
  • **Acceptability**: Whether a native speaker can use that form in a context
  • **Correctness**: Whether the form conforms to prescriptive rules of language
  • **Strangeness/Infelicity**: Whether the form is pragmatically acceptable

In the present context, the type of errors considered are those related to acceptability. If a native speaker could not use the expression to convey the same semantic meaning in the given context, then the expression is unacceptable. However, the only way to know whether a learner response reflects some degree of unacceptability is to compare it to a target response that is acceptable.
So, to describe errors in terms of such a comparison, James (1998) presents a set of error types that reflect incorrect “target modification.” Given that the current approach identifies similarities and differences between the target and learner response in order to diagnose errors, this descriptive approach is useful. The five categories James identifies include the following:

1. **Omission** – Material is left out of the learner response (e.g., *He went prison*)
2. **Overinclusion** – Addition of incorrect material, such as over-application of rules (e.g., *He goed to prison*)
3. **Misselection** – Use of the wrong form or alternation (e.g., *He get a prison sentence*)
4. **Misordering** – Misplacing material (e.g., *Got he a prison sentence*)
5. **Blends** – Combining two forms to produce an incorrect form (e.g., *He got a sentence to prison*)

A second way to describe learner errors is by their linguistic characteristics. James (1998) presents another descriptive error taxonomy organized by linguistic unit of the error (e.g., morphological errors, lexical errors, syntactic errors, etc.)\(^{20}\) This way of classifying errors identifies many semantic errors as word choice errors.

Of the word choice errors, James (1998) divides the error classes into formal and semantic, though all involve incorrect word selection. Formal errors include synform, misformation and distortion categories. **Synform errors** result in words that look or sound similar to the target word.\(^{21}\) These errors are real-word substitution errors. Synform errors can be further classified into suffix-type, prefix-type, vowel-based and consonant-based mistakes.

---

\(^{20}\)Note that descriptive taxonomies based on linguistic units or target modification are not mutually exclusive.

\(^{21}\)These errors are sometimes referred to as malpropisms. In addition, if there is a resemblance in form between the L1 word and the L2 target, then such terms are called *false cognates.*
Misformations result in non-existent word errors and are often a product of using the L1 term without translation, translating an L1 term literally (but inappropriately) or by word coinage.

And finally, distortions are also non-word errors. But unlike misformations, distortions have no connection to the L1. Distortions may include non-words produced by omissions, overinclusions, misselections, misordering and blending.

Although these form-based distinctions are useful for conceptualizing what errors learners make, it is not at all clear how to distinguish between many of the categories in any practical way. For instance, without a possibly extensive understanding of the L1, distinguishing many misformations from distortions seems rather difficult.

Non-word errors can often be detected through the use of a spell checker, but such spell checkers do not provide an analysis that would prove useful in distinguishing between overinclusions and misselection, misselection and misordering, etc.

In addition to form-based selection errors, James also defines semantic word choice errors, which broadly cover confusion of sense relations and collocation errors. In confusing sense relations, the learner may use a hypernym when a hyponym is called for, use a hyponym for a hypernym, use the wrong co-hyponym or use the wrong near synonym. For collocations, learners may make word selection errors, preference errors and incorrectly combine words. To be able to detect the sense relation errors, there needs to be an available taxonomy relating the words in the target language. For English, WordNet might be sufficient for this purpose, for nouns at least.

The detection of collocation errors relies on the availability of reliable frequencies of different word combinations and collocations for the target language.

A different classification is provided by Schmidt et al. (2004) for categorizing meaning errors in learner responses in the context of an ICALL setting. They divide
the task of semantic error checking into *semantic correctness* and speech act evaluation. For semantic correctness, they further divide the task into evaluating i) whether the appropriate words are used, ii) whether words are combined into well-formed sentences, and iii) whether the response is appropriate given the dialogue.

Still, even though Schmidt et al. (2004) give a description of an error checking architecture, it is unclear whether (and if so, how) the system was actually implemented to check these features. Without any operationalization of what it means to have appropriate words and responses, it is not a sufficient breakdown of errors.

Looking outside ICALL and second language research, work in other CL tasks on text comparison may provide useful insights into meaning error classification. For instance, in her work in machine translation, Dorr (1994) defines divergences and mismatches in translation. Divergences are cross-linguistic differences in translations. A categorical divergence occurs from English to Spanish in the translation of *I am hungry*, for example. The Spanish equivalent is *Yo tengo hambre*, which literally translates as *I have hunger*. Thus, *hungry* categorically diverges in translation by changing category from adjective to noun. Dorr (1994) identifies seven types of divergent translations – thematic, promotional, demotional, structural, conflational, categorical and lexical. All refer to grammatical and lexical changes. But with divergences, the same information is conveyed across translations. Translation mismatches, on the other hand, occur when the translations convey slightly different information. For instance, the word *fish* translates as *pez* or *pescado*, depending on whether it has been caught.\textsuperscript{22}

In an ICALL context, a learner response may be thought of as an attempt at a monolingual translation of a target response. If a learner response and a target response convey the same meaning, but use different linguistic expressions to do so, \textsuperscript{22}Example from Dorr (1994).
then the responses may be said to diverge. If they convey slightly different concepts, they may be mismatches. However, divergences capture legitimate linguistic variation and mismatches are not necessarily errors. Thus, Dorr’s categorization of divergences and mismatches are not an error classification but a difference classification. Thus, it may be useful for identifying how a learner response may be allowed to vary from the target, rather than in describing how the learner response is not allowed to vary from the target.

Another analysis of differences in semantic content is provided in the description of the Microsoft Paraphrase Corpus (Dolan et al., 2005). To develop the corpus, human judges read pairs of sentences and labeled them as equivalent or not equivalent. The annotation guideline for the human judges of candidate paraphrases notes that minor differences in content are acceptable for equivalent pairs. For instance, differences in anaphora (i.e., use of pronouns) were ignored.

Differences leading to “Not equivalent” labels included when
- one sentence lacked details present in the other sentence such that the concepts in the less informative sentence were a subset of the concepts in the more informative sentence,
- one sentence did not necessarily refer to the same event as the other sentence,
- the sentences being compared used different rhetorical structures, and
- the sentences described the same event, but with a different emphasis.

Of course, the authors note that the guidelines were deliberately vague and required judgment calls by the annotators. As a result, the average agreement between judges was 83.5%.

The types of differences described for the paraphrase corpus annotation are relevant to CAM because the task of the human judgments of paraphrases is similar to what CAM must do in an ICALL context. CAM is essentially determining whether
the differences between target and learner responses can be ignored. However, CAM clearly needs more precise description of which differences can and cannot be ignored than what is provided to human judges. Otherwise, CAM cannot link observable phenomena in the learner responses to any particular error.

5.5.2.1 Additional Issues with Error Taxonomy Development

In addition to needing a more precise way to characterize meaning errors, there are several important issues associated with placing a given learner error into any error category. In their work on transfer errors from American Sign Language (ASL) to English, Suri and McCoy (1993) develop an error taxonomy using essay data from signers learning written English. The relevant concerns they raise include the following:

1. The intent of the learner is not always (if ever) discernible. This issue is also nicely summed up by Garrett (1995, p. 353), who says,

   “... no matter how detailed and accurate a parser is, parsing is basically an analysis of language form (i.e., what the error is), not an analysis of language processing (i.e., why the learner made the error) or an analysis of language acquisition (i.e., how and why the processing changes over time).”

   While Garrett’s comments were clearly about syntactic analysis and error detection, the same can be said of meaning error detection. Just like a human instructor, the computer cannot be sure what the learner intended. So (how) can the computer identify what kind of error the learner made if it cannot know what the learner meant?
2. The underlying reason for an error may not be clear. Learners may truly not know a form (a competence issue) or may know the form but inadvertently misuse it (a performance issue). Should these errors be treated identically?

3. Errors may be ambiguous. For example, subject-verb agreement errors may result because the learner made an error in the ending of the verb or the noun. If there are multiple errors that apply, how can the system/human choose between them?

4. Errors are not independent of one another. If one error triggers another, are they counted as a single or multiple errors?

5. Multiple instances of the same type of error are possible in a single learner’s response. Should each instance count as a separate error?

These issues are relevant to both taxonomy development and human/automatic error diagnosis. And for CAM, the design choices for the error taxonomy do take these issues into consideration. The approach adopted for the CAM design focuses more on how learner responses are different than why the responses are different from a target response. In this way, the (likely indiscernible) intent of the learner is minimized as a factor in error analysis, beyond the assumption that learners actually intend to reproduce the target content in their responses. The reason the learner made the error is also ignored here in the sense that the CAM design makes no effort to determine whether the learner error is a simple typo or reflects an underlying lack of knowledge.\(^{23}\)

But focusing on observable error characteristics does not avoid the possibility that errors may be ambiguous or that multiple errors may be present in the learner response. And errors may be dependent on other errors. For instance, if the learner

\(^{23}\)A larger ICALL system might attempt this through the use of learner models that track performance over time for a range of language skills.
uses the wrong verb, then arguments of that verb may also be wrong, triggering other errors. To simplify diagnosis, at least in the initial implementation, only one type of error is reported at a time and multiple wrong or missing concepts are classified as a single type of error.

5.5.2.2 Target Modification and the CAM Design

An underlying assumption in the CAM design is that for any activity in the viable processing ground, there is a target response (or set of target responses) provided in the activity model that represents the acceptable answer to the activity. Another fundamental assumption is that the learner is cooperative in his/her response to an activity. That is, the learner attempts to provide an acceptable answer by trying to “hit the target.” Given these assumptions, an error taxonomy that is based on ways the learner could have missed the target is a reasonable starting point. Thus, the target modification approach described in James (1998) warrants additional consideration. While James’ error taxonomy was developed to describe grammatical errors, Table 5.14 shows a possible modification of that taxonomy to reflect meaning errors. Other than misorderings, these meaning error types would result in unaligned learner and target concepts.24

Error classification in the initial CAM implementation focuses on three of the error types described in Table 5.14 – omissions, overinclusions and blends. Omission errors can occur when the learner fails to include one or more concepts in his or her response. Presumably, if the instructor provides a list of keywords for a response and the learner fails to include one or more of the keywords, an omission error diagnosis should be considered (along with the possibility of a misselection error). However,

24Note that in Table 5.9, several types of alignment failures are presented. Learner errors are included there as a single category. The remaining alignment failures in that table are cases in which the learner did not make an error.
<table>
<thead>
<tr>
<th>Meaning Error</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Omission</td>
<td>Necessary concepts are left out of the learner response</td>
</tr>
<tr>
<td>Overinclusion</td>
<td>Learner response contains extra, incorrect concepts</td>
</tr>
<tr>
<td>Misselection</td>
<td>An incorrect substitution (wrong concept) is present</td>
</tr>
<tr>
<td>Misordering</td>
<td>Concepts are in the wrong argument position</td>
</tr>
<tr>
<td>Blends</td>
<td>Incorrect combinations/use of multiple concepts</td>
</tr>
</tbody>
</table>

Table 5.14: Meaning Errors Based on Target Modification

if the instructor does not explicitly identify which concepts from the target response are critical, the diagnosis is not clear. For instance, consider the question cue, target response, and set of learner responses in (68). The learner response in (68c) matches all the concepts and should be labeled as correct. Response (68d) is missing the concept home and should be marked as incorrect.

(68) a. **Question Cue:** Where was Bob Hope when he heard about the news?
    b. **Target Response:** He was eating breakfast at his home.
    c. **Learner Response 1:** He was in his home at that moment happily eating his breakfast.
    d. **Learner Response 2:** Bob Hope was at that moment happily eating his breakfast when he heard about the news.
    e. **Learner Response 3:** He was in his house.

But if all the concepts in the target response in (68b) are required, then the response in (68e) would also be marked as incorrect because it fails to include eating and breakfast. This is unacceptable since the question asks only where Bob Hope was (i.e., at his home), not what he was doing (i.e., eating breakfast). Unlike (68d), the missing concepts in (68e) are not crucial for correctly answering the question. This suggests a need for clear, precise target responses and careful selection of keywords.
In contrast to omission errors, overinclusion errors include material in the learner response that is extra (i.e., not in the target response) and incorrect. But when are extra words and phrases wrong? Can diagnosis ignore the extra material in the learner responses? Analysis of the development set would suggest that diagnosis should often ignore extra material. Of the 311 learner responses in the development set, 177 had extra material in the learner response. For instance, the learner responses in (68c) and (68d) contain extra concepts. (68c) conveys Bob Hope’s emotive state at eating breakfast, and (68d) unnecessarily repeats some of the question material. But these inclusions are not incorrect.

In fact, of the 177 responses with extra material, only 4 (2%) contained overinclusion errors, such as the one presented in (69). While the learner response in (69c) contains the essential concept *cartoons*, it also includes other television programming that is incorrect as a response to the question cue.

(69) a. **Question Cue:** Which form of programming on TV shows the highest level of violence?

   b. **Target Response:** Cartoons show the highest level of violence.

   c. **Learner Response:** Television drama, children’s programs and cartoons.

This suggests that extra material in the learner response is likely acceptable in diagnosis, but not always. In a discussion of issues related to automated scoring and diagnosis using a regular-expression-based approach, Carr (2007) argues that whether to penalize extraneous material is one of the most difficult issues to contend with. He identifies three approaches to dealing with extraneous information, which address the issue from different perspectives. The first is to provide clear instructions to the learner in the activity model about avoiding irrelevant information in their responses. This approach tries to eliminate the extraneous information that comes in as input to the grading system through carefully constructed activity models.
The second approach tackles the problem through target responses. Note that this goes hand in hand with the approach to avoiding false diagnosis of omission errors mentioned above, which requires more precisely constructed target responses and keyword lists to eliminate unneeded concepts. Given such construction of targets, the target responses would typically include words, phrases and sentences that are required in the learner response. When some or all of the target material is found in the learner response, it is taken as positive evidence that the learner response is correct. Carr (2007) proposes including negative evidence – specific words or phrases that count as incorrect answers. However, this negative evidence must be pre-envisioned, and Carr argues they are best for handling common, known mistakes the learner might make. This use of negative evidence may also be a partial solution to the issue of how to handle incorrect responses that are otherwise very similar to a target response.

The final approach is to set a length limit for learner responses. Carr argues this avoids “kitchen sink” answers in which the learner writes (or copies) a lot of text in hopes that some part of it is the correct answer. One way to set the length limit is to make it proportional to the target response and to restrict the maximum length for any particular answer. However, Carr suggests that this approach should be mixed with the second approach that identifies particular negative phrases. But whatever the strategy chosen, the development data suggest that overinclusion errors are relatively infrequent, and additional processing may be required to recognize when extra material in the learner response is incorrect.

The final class of errors considered are blend errors. These cover multiple concept differences between the learner and target responses, as in the examples from the development set in (70). In (70c), the learner response partially overlaps with the target response. The learner correctly named some of the features, and incorrectly named others. In (70d), none of the features are correct. In fact, the
learner did not answer the question. Rather than naming features used in the design of advertisements, the learner listed the images created by those advertisements.

(70) a. **Question Cue:** Name the features that are used in the design of advertisements.

b. **Target Response:** The features that are used are eye contact, color, famous or glamorous people, language and cultural references.

c. **Learner Response 1:** jewelry, watches and clothes. Eye contact and color
d. **Learner Response 2:** beauty, wealth, an idyllic location, romance or popularity

In the case of (70d), the learner response is so far off from addressing what the question was asking for that it could be considered a non-answer. The learner entirely fails to answer the question. This is an extreme form of blend errors and is given special consideration in diagnosis here.

The two remaining error types in Table 5.14 are misselection and misordering but are not addressed in the present study. Misselection, a substitution error, can be thought of in edit distance terms as a omission followed by an overinclusion error. In this way, it is a combination of errors, and therefore a type of blend error. Thus, while it may also deserve special treatment, that is left to future research.

On the other hand, misordering errors do not readily fall under the blend category. All the concepts are present, just in the wrong position. Diagnosing these errors requires clear understanding of the relationships between concepts. But in the small corpus collected, there were no instances of misordering errors. Thus, this type of error, which would require individual treatment, is also not included in the diagnosis described in the next section.
5.5.3 CAM Response Analysis

Intuitively, the three basic error types considered – omission, overinclusion and blends – correspond to the cells in a grid comparing high and low overlap of target and learner concepts, as depicted in Figure 5.3. This potential connection between unaligned concepts and error diagnosis is a simple starting point for exploring diagnosis.

However, the one cell of the grid not yet discussed is that containing correct answers. The CAM design splits response analysis into detection (whether or not there is an error) and diagnosis (the type of error, if any). In principle, correct answers should be those that are similar enough to a target response as to convey the required meaning. But, in analysis of the development set, it became clear that a different category of correct answer was possible. Consider the pair in (71).
(71) a. **Question Cue:** What was the major moral question raised by the Clinton incident?

b. **Target Response:** The moral question raised by the Clinton incident was whether a politician’s personal life is relevant to their job performance.

c. **Learner Response:** If a person is not honest and faithful to his or her spouse, that person will not be honest and faithful to his or her country.

While the learner response in (71c) in very dissimilar to the target response, it is an entirely acceptable alternate answer, given the source text material. In some sense, alternate answers are the opposite of non-answers. That is, non-answers do not try to hit the target response and are wrong, while alternate answers also do not try to hit the target but are right. Such responses are highly problematic for the current approach. The CAM design assumes there will be some overlap of concepts for correct learner responses. With alternate answers, this may or may not be the case. Thus, the CAM implementation may be unreliable in determining whether alternate answers are actually correct. The best solution for this issue, though not implemented here, is to expand the set of target responses to include a wider range of acceptable answers. This will be discussed further in Chapter 7.

5.5.3.1 Relative Overlap for Detection and Diagnosis

The number and kinds of matches collected in the Alignment phase might be combined in several ways to detect and diagnose errors. As mentioned above, perhaps one of the simplest is to focus on the relative numbers of matches. To implement a diagnosis module following this approach, thresholds for relative learner and target concept overlap (only at the token level) were empirically determined and diagnosis proceeded using the following conditions:
1. If the percent of unaligned target concepts is below the threshold and the percent of unmapped learner concepts is also below the threshold, then label the response Correct.

2. If the percent of unaligned target concepts is below the threshold but the percent of unmapped learner concepts is above the acceptable threshold, then label the response as an Overinclusion error.

3. If the percent of unaligned learner concepts is below the threshold but the percent of unmapped target concepts is above the acceptable threshold, then label the response as an Omission error.

4. If the percentages of unaligned learner and target concepts are above the allowable thresholds, label the response a Blend error.

5. If the percentages of unaligned learner and target concepts are above the allowable thresholds and are, in fact, above an addition maximum unaligned threshold, label the response a Non-answer.

The development set results and evaluation will be discussed in some detail in Chapter 6. For now, it is enough to say that this approach obtained as much as 81% accuracy on the development set in detecting whether or not an error is present and less than 70% overall accuracy in diagnosing errors. And while the results were good for the development set, this approach is less than satisfying because it ignores chunk- and triple-level information, as well as the type of match at the token level. And it is not immediately clear how to best combine all the collected information into a single, manually tuned diagnosis component. However, the number of features and potential for combination do lend themselves nicely to a solution using machine learning.
5.5.3.2 Machine Learning for Detection and Diagnosis

Several previous works in related fields have applied machine learning to the task of equivalence recognition. For instance, the work of Hatzivassiloglou et al. (1999), described in Chapter 4, trained a classifier for paraphrase detection, though their performance only reached roughly 37% recall and 61% precision. In a different approach, Finch et al. (2005) found that MT evaluation techniques combined with machine learning improves equivalence recognition. They used the output of several MT evaluation approaches based on matching concepts (e.g., BLEU) as features/values for training a support vector machine (SVM) classifier. Matched concepts and unmatched concepts alike were used as features for training the classifier. Tested against the Microsoft Research Paraphrase (MSRP) Corpus, the SVM classifier obtained 75% accuracy on identifying paraphrases. But at the time of this writing, it does not appear that machine learning techniques have been applied to or even discussed with respect to meaning error diagnosis in ICALL systems.

To begin to address the application of machine learning to meaning error diagnosis, the data collected from the implementation described in Section 5.4 were converted into features. These features are listed in Table 5.15. Features 1-7 reflect relative numbers of matches (relative to length of either the target or learner response). Features 2, 4, and 6 are related to the target response overlap. Features 3, 5, and 7 are related to overlap in the learner response. Features 8-13 reflect the types of matches.

The values for the 13 features in Table 5.15 were used to train the detection classifier. For diagnosis, a fourteenth feature – a detection feature (1 or 0 depending on whether the detection classifier detected an error) – was added to the development data to train the diagnosis classifier. Given that token-level alignments are used in
<table>
<thead>
<tr>
<th>Features</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Keyword Overlap</td>
<td>Percent of keywords aligned (relative to target)</td>
</tr>
<tr>
<td>2. Target Overlap</td>
<td>Percent of aligned target tokens</td>
</tr>
<tr>
<td>3. Learner Overlap</td>
<td>Percent of aligned learner tokens</td>
</tr>
<tr>
<td>4. T-Chunk</td>
<td>Percent of aligned target chunks</td>
</tr>
<tr>
<td>5. L-Chunk</td>
<td>Percent of aligned learner chunks</td>
</tr>
<tr>
<td>6. T-Triple</td>
<td>Percent of aligned target triples</td>
</tr>
<tr>
<td>7. L-Triple</td>
<td>Percent of aligned learner triples</td>
</tr>
<tr>
<td>8. Token Match</td>
<td>Percent of token alignments that were token-identical</td>
</tr>
<tr>
<td>9. Similarity Match</td>
<td>Percent of token alignments that were similarity-resolved</td>
</tr>
<tr>
<td>10. Type Match</td>
<td>Percent of token alignments that were type-resolved</td>
</tr>
<tr>
<td>11. Lemma Match</td>
<td>Percent of token alignments that were lemma-resolved</td>
</tr>
<tr>
<td>12. Synonym Match</td>
<td>Percent of token alignments that were synonym-resolved</td>
</tr>
<tr>
<td>13. Variety of Match</td>
<td>Number of kinds of token-level alignments</td>
</tr>
</tbody>
</table>

Table 5.15: Features for Machine Learning of Diagnosis

identifying chunk- and triple-level alignments, that kinds of alignments are related to variety of matches, etc., there is clear redundancy and interdependence among features. But each feature adds some new information to the overall diagnosis picture.

The machine learning algorithm suite used in all the development and testing runs is TiMBL (Daelemans et al., 2007), the Tilburg Memory-Based Learner. As with all the tools discussed to this point, TiMBL was chosen mainly to illustrate the approach. It was not evaluated against several learning algorithms to determine the best performing algorithm for the task, although this is certainly an avenue for future research.
As a set of memory-based learning algorithms, TiMBL uses the training phase of learning to store instances in memory. In testing, a new instance is compared to the stored instances and classified based on its similarity to one or more of the stored instances. TiMBL comes with many options, and experiments with these options included varying how similarity between instances was measured, how importance (i.e., weight) was assigned to features and how many neighbors (i.e., instances) were examined in classifying new instances. However, because the tweaking of the classifiers is tied closely to the evaluation of the development set data, further explanation of the configuration selected is deferred to Chapter 6. For the moment, the classifiers will remain a black box in which the features are fed as input and the output is a code corresponding to diagnosis.

5.5.3.3 Error Reporting

As perhaps suggested by the brief overview of feedback in Section 5.5.1.1, how errors are reported in ICALL could be the subject of an entire thesis. How many of which errors and in which order are a few of the concerns that depend on factors such as the level of the student, the nature of the error, the goals of the activity, the pedagogical foundations for both the exercise and feedback type, etc. The full range of considerations will not be discussed because CAM is intended as a component of a larger ICALL system that would manage error reporting independently. CAM only identifies meaning errors and currently reports all it finds. A separate feedback manager would be responsible for selecting what to present to the learner and how to present it that best fits with the goals of the ICALL system it is a component of. Debating the merits and tradeoffs of such decisions are outside the scope of the current study. For instance, in the absence of an ICALL system with a working learner model, there is no way to practically implement feedback tailored to the individual student.
Rather, a diagnosis code is delivered as the final output of the system, which might be passed to a larger ICALL system or used to generate canned text in feedback.

5.6 Conclusion

This chapter has covered a lot of ground in describing the design and implementation of a content assessment module. The CAM design is illustrated through an implementation that draws on the linguistic analysis provided by a range of NLP tools. These linguistic analyses are used to identify token-, chunk- and triple-based alignments across target and learner responses. And the alignments are, in turn, used to classify the learner response as correct or incorrect. In the latter case, a diagnosis based on target modification is provided.

At each phase of processing, the development data were used to generate ideas for what linguistic units to compare, how to compare them and how they relate to a final analysis of a learner response. Chapter 6 takes a closer look at the data itself to describe how the learner corpus was collected, annotated and split into training and development sets. From there, the discussion turns to evaluating the design against a set of the test pairs form the corpus.
CHAPTER 6

EVALUATION

Research in ICALL has suffered from a lack of standardized evaluation. The absence of freely available, error-annotated learner corpora, the diversity of languages ICALL systems are designed for, and the lack of clear reporting of system function and performance have arguably hampered progress in the field. This is not a problem unique to ICALL systems. Phillips (2007, p. 41) notes that for automatic essay scoring, it is difficult to ensure “transparent scientifically based comparison” because there are different scoring mechanisms, focuses, and technology uses to contend with. All this makes evaluation of automatic grading a challenge. The same is true for ICALL system evaluation. If there are no variables held constant, performing a scientific comparison across ICALL systems is simply not possible. And it is difficult to evaluate progress in the field without such measurable comparisons.

In an effort to begin to redress this weakness in the field, this chapter provides a qualitative and quantitative evaluation of the CAM design implementation described in Chapter 5. However, because the design and evaluation of the implementation relied heavily on the small learner corpus collected for that purpose, the first section centers on describing the corpus data.
6.1 Building a Corpus

To develop and evaluate CAM, a set of written responses to reading comprehension questions were collected from English language learners from intermediate reading/writing courses of the American Language Program (ALP) at The Ohio State University. The ALP program offers intensive English instruction to undergraduate- and graduate-level students. Classes are kept small, with less than 20 students per course. And at least one Level 3 Reading/Writing course is offered each quarter.

Data collection began in Autumn 2006 to gather responses to the reading comprehension questions. From the outset, the collection goals were to minimize the impact on instructors and students in order to collect data typical of this type of activity. The questions themselves were assigned as part of the existing course curriculum rather than artificially imposed on the instructors and students. The ALP Level 3 Reading/Writing course curriculum guidelines outline several skills targeted in the reading comprehension component of the class (ALP, 2005). These skills are listed in Table 6.1.

Each ALP instructor had some flexibility in implementing the curriculum guidelines and in choosing which activities to assign to students and how to evaluate students. Thus, appropriate question sets/activities were identified at the start of each collection quarter. However, due to instructor variation in implementing the curriculum guidelines, many activities were not suitable for data collection in the study. Indeed, whole courses were planned without the use of materials appropriate for the current study. And the instructors themselves varied in how they evaluated the students’ work, but the student populations were given the same instructions for participating in the collection project.
Reading Comprehension Skills

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Pre-reading (skimming, noting headers, relating reading to personal knowledge)</td>
</tr>
<tr>
<td>2.</td>
<td>Reading for the main idea</td>
</tr>
<tr>
<td>3.</td>
<td>Identifying meaning from context</td>
</tr>
<tr>
<td>4.</td>
<td>Analyzing word morphology</td>
</tr>
<tr>
<td>5.</td>
<td>Analyzing supporting details</td>
</tr>
<tr>
<td>6.</td>
<td>Recognizing inference</td>
</tr>
<tr>
<td>7.</td>
<td>Recognizing discourse structure</td>
</tr>
<tr>
<td>8.</td>
<td>Reading for specific purposes</td>
</tr>
<tr>
<td>9.</td>
<td>Restating the reading content (note taking, paraphrase, explaining the reading, recognizing intent/opinion, synthesizing information, etc.)</td>
</tr>
<tr>
<td>10.</td>
<td>Recognizing tone</td>
</tr>
<tr>
<td>11.</td>
<td>Recognizing literal meaning and metaphor</td>
</tr>
</tbody>
</table>

Table 6.1: Targeted Reading Skills for Level 3 ALP Course

All prompts used to collect responses were those distributed or assigned to students as post-reading comprehension questions, either as a separate instructor-designed worksheet or as activities from the course textbook, *Academic Encounters: Life in Society* (Brown and Hood, 2002). These comprehension questions required single-word, phrasal or sentential answers, although students were encouraged to respond in complete sentences. In all cases, students had access to their textbooks when completing activities.

Several examples of cues, target responses and learner responses have been presented already in explaining the CAM design in Chapter 5. However, additional examples of question cues and response pairs are listed in Figure 6.1.

The resulting corpus was split into two sets – a development set for building CAM and a test set for evaluating the system on unseen data. The development data came from the Autumn 2006 collection of short-answer responses and originally
<table>
<thead>
<tr>
<th>Question Cue:</th>
<th>Target Response:</th>
<th>Learner Response:</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is alliteration?</td>
<td>Alliteration is repetition of the initial sound in sequential words. The worlds are often chosen to make some pattern or play on works. Sequential works begins with the same letter or sound.</td>
<td></td>
</tr>
<tr>
<td>Which form of programming on TV shows the highest level of violence?</td>
<td>Cartoons show the most violent acts.</td>
<td>Television drama, children's programs and cartoons.</td>
</tr>
<tr>
<td>Where was Bob Hope when he heard about the news?</td>
<td>He was eating his breakfast at home.</td>
<td>He was at home and eating his breakfast.</td>
</tr>
</tbody>
</table>

Figure 6.1: Example Corpus Data

consisted of 327 responses to 50 questions from 11 students. The test data came from responses collected in Spring 2007. Fifteen students participated in the Spring 2007 data collection, providing 270 responses to 30 questions.

Of the development data, three questions and their 16 responses were discarded. In addition, two questions and their 15 responses were discarded from the test set. These discarded questions are listed in Table 6.2. The first two questions in Table 6.2 were discarded because the responses were graphs, not text, and there is no mechanism for evaluating images in the CAM design. The second two questions in the table were discarded because it was not possible to specify a well-defined target response for these questions. Rather, the questions required the learner to draw on their background knowledge, and any number of responses would have been suitable.

For both development and test sets, not all students answered every question or turned in every activity sheet. The final sets used to develop and evaluate CAM

---

1The last question in Table 6.2 was found in both the development and test sets.
Discarded Question | Responses
---|---
Make a PIE CHART of the 67 studies. | 3
Make a BAR GRAPH of the information given in paragraph 6. | 3
Give me another, not listed in the book, example of positive modeling and negative modeling. | 6
How do you think the statistics in your country compare to those in the story on employees use of the internet at work? Why? | 19

Table 6.2: Discarded Questions from Development and Test Sets

consisted of 311 responses to 47 questions (development) and 255 responses to 28 questions (test). The breakdown of data by question and student submission for each activity sheet are given in Tables 6.3 and 6.4, for the development and test sets, respectively.

<table>
<thead>
<tr>
<th>Development Question Sets</th>
<th>Question Count</th>
<th>Student Submissions</th>
<th>Total Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit 3, Ch. 5, Reading 3</td>
<td>11</td>
<td>5</td>
<td>54</td>
</tr>
<tr>
<td>Unit 3, Ch. 5, Reading 4</td>
<td>9</td>
<td>11</td>
<td>92</td>
</tr>
<tr>
<td>Unit 3, Ch. 6, Reading 1</td>
<td>6</td>
<td>7</td>
<td>40</td>
</tr>
<tr>
<td>Unit 3, Ch. 6, Reading 2</td>
<td>6</td>
<td>10</td>
<td>60</td>
</tr>
<tr>
<td>Unit 3, Ch. 6, Reading 3</td>
<td>7</td>
<td>6</td>
<td>41</td>
</tr>
<tr>
<td>Unit 3, Ch. 6, Reading 4</td>
<td>8</td>
<td>3</td>
<td>24</td>
</tr>
</tbody>
</table>

Table 6.3: Student Submissions for Development Data Questions

Students submitted written responses to their instructors. These responses were later manually typed at have machine-readable responses, but all errors in grammar, spelling, punctuation and capitalization were preserved. As mentioned above, the low-impact approach to collecting data, while producing interesting responses typical for the activity given, resulted in inconsistencies in the data sets in spite of efforts to hold constant the collection procedures across courses. Specifically, for the
Table 6.4: Student Submissions for Test Data Questions

<table>
<thead>
<tr>
<th>Test Question Sets</th>
<th>Question Count</th>
<th>Student Submissions</th>
<th>Total Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit 1, Ch. 1, Reading 3</td>
<td>6</td>
<td>10</td>
<td>60</td>
</tr>
<tr>
<td>Unit 1, Ch. 1, Reading 4</td>
<td>3</td>
<td>15</td>
<td>45</td>
</tr>
<tr>
<td>Unit 3, Ch. 5, Reading 3</td>
<td>6</td>
<td>10</td>
<td>60</td>
</tr>
<tr>
<td>Unit 3, Ch. 6, Reading 2</td>
<td>6</td>
<td>7</td>
<td>42</td>
</tr>
<tr>
<td>Unit 3, Ch. 6, Reading 3</td>
<td>7</td>
<td>7</td>
<td>48</td>
</tr>
</tbody>
</table>

deviation set, the instructor provided an answer key for each activity, but did not grade learner responses. For the test set, no answer key was provided, but the instructor (a different instructor than for the development data) graded learner responses, marking answers as correct or incorrect.

6.1.1 Language Learners

The majority of ALP students are preparing to apply to colleges and universities at the graduate or undergraduate level. A few students are business professionals polishing their English skills. Respondent ages ranged from 22 to 30 for contributors to the development set and 18-31 for contributors to the test set. Seven men and four women contributed to the development set; nine men and six women contributed to the test set.2

The native languages (L1) of learners in Level 3 varies. The breakdown of L1’s reported by participants is listed in Table 6.5. The most common L1 for learners in the development and test sets was Arabic. Those students who reported Chinese as their L1 were not specific as to whether they spoke Mandarin, Cantonese, Taiwanese or some other language/dialect of that region.

2Age, and gender, and native language were self-reported.
<table>
<thead>
<tr>
<th>Language</th>
<th>Development</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arabic</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Japanese</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Chinese</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Turkish</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Korean</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>11</strong></td>
<td><strong>15</strong></td>
</tr>
</tbody>
</table>

Table 6.5: Learner Native Languages

Intermediate/high intermediate students were targeted because i) they know enough English for variation to be possible in their responses and ii) their knowledge was at a level where pattern practice and drills are less common (i.e., the task complexity is such that responses can show variation).

6.1.2 Annotating the Corpus

To create a consistent set of model answers, an independent grader was hired to i) develop a set of target responses for the corpus and ii) use those target responses to score and diagnose errors in the learner responses. Guidelines for developing target responses included in Figure 6.2.

In grading, the grader assigned a score of 0 (no credit), 1 (partial credit) or 2 (full credit), along with a diagnosis code. Responses that received no credit provided insufficient information or contained enough meaning errors to be considered incorrect. Partial-credit answers were “on the right track” but contained meaning errors that prevented the responses from being considered completely correct. Full-credit answers conveyed the expected meaning of the target response and all necessary concepts. The grader also assigned a binary judgment – *Y* or *N* – that eliminated partial credit. The binary judgment essentially collapsed the 0 or 1 score to *N*.
Guidelines for Target Responses

1. For each question, use the textbook to create a model answer.
2. A model answer a sentence (or sentences) that sufficiently answers the question.
3. Be brief and include only what is needed for a correct answer.
4. Avoid using pronouns whenever possible to refer to something not mentioned elsewhere in the question or model answer.
5. Using phrases from the question text (or textbook) is acceptable to create model answers, but avoid copying whole sentences from the text (again, if possible).
6. If the question has more than one acceptable model answer given the textbook, list all model answers.

Figure 6.2: Target Response Construction Guidelines

Beyond scoring, the diagnosis codes assigned to learner responses provide a richer description of why the learner response was correct or incorrect. The set of diagnosis codes used in grading are given in Table 6.6, along with the description of the code given to the grader and how many instances of each were ultimately labeled in the data.

Most of the codes correspond directly to the meaning error diagnosis types and terminology used in Chapter 5. CA is used for correct answers that are similar to a target response, MC is for omission errors, EC is for overinclusion errors, and MD is for blend errors. NA was used to label extreme blend errors, the non-answers. For those correct learner responses that were not similar to the target response, the grader was instructed to assign an alternate answer (AA) diagnosis code. The two codes provided to the grader, but ultimately not used for any of the pairs, were NM (No Model) for any cases in which the grader could not construct a response and OA (Other Answer) in cases where a response could not be assigned any other category. Given that the last two were not used, they will not be discussed further.
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA</td>
<td>Correct Answer. The students answer conveys the correct meaning. That is, their answer is sufficiently similar to one (or more) of the model answers specified for a given question to be given “full” credit. Note that a correct answer by this definition may contain grammatical errors.</td>
</tr>
<tr>
<td>AA</td>
<td>Alternate Answer. If the student response is significantly different than the model answer, but still acceptable as a correct answer (given the question and the textbook material), the AA code is appropriate.</td>
</tr>
<tr>
<td>MC</td>
<td>Missing Concept. The students answer does not include one or more of the concepts required for a complete answer, but is otherwise similar to one of the model answers for that question. MC answers would get partial or no credit, depending on the severity of the omission(s).</td>
</tr>
<tr>
<td>EC</td>
<td>Extra Concept. The students answer is similar to a model answer but contains extra concepts that are incorrect. This code is NOT intended for student answers that contain extra material that is not necessary (but also not wrong). EC answers would get partial or no credit.</td>
</tr>
<tr>
<td>SE</td>
<td>Substitution Error. The students answer is similar to a model answer, but it contains an incorrect concept “substituted” for the correct one. SE answers receive partial credit or no credit, depending on the severity.</td>
</tr>
<tr>
<td>MD</td>
<td>Multiple Differences. The students answer differs from a model answer and contains multiple Missing Concept, Extra Concept or Substitution errors. These answers would receive no credit.</td>
</tr>
<tr>
<td>NA</td>
<td>Non-Answer. The students response completely failed to answer the question or convey the correct meaning. This is different from MD in that the students answer is significantly different from any model answers for a given question and fails to answer the question, given the information provided in the text. These responses receive no credit.</td>
</tr>
<tr>
<td>NM</td>
<td>No Model. No model answer could be constructed to grade the response.</td>
</tr>
<tr>
<td>OA</td>
<td>Other Answer. The student’s answer fails to fit into any of the above categories.</td>
</tr>
</tbody>
</table>

Table 6.6: Diagnosis Codes and Descriptions
The final code to discuss is SE, for misselection errors. An example of such an error in the corpus is given in (72), with the substitution error underlined. Essentially, the learner used the wrong number.

(72) a. **Question Cue**: Does all of the research point to the bad influence of violence on TV?

b. **Target Response**: No, 20 percent didn’t show a clear-cut connection, and 3 percent showed a decrease in aggression.

c. **Learner Response**: No. In 20% of cases there were no clear-cut results and in 30% of studies indicated that watching TV violence decrease aggression.

In order to determine how often misselection errors appeared in the corpus, a separate label was used for such misselection errors. There were only two instances in the development set and none in the test set. Thus, responses labeled SE were mapped to MD (blend) errors, as discussed in Chapter 5.

6.1.3 Inter-rater Reliability

One concern with respect to the corpus is the reliability of the judgments provided. The reliability of human graders impacts the evaluation of the CAM design in two ways. First, it is an issue for ‘gold standard’ judgments and corpus creation. If the human judgments provided by instructors are potentially wrong, then using human judgments to evaluate system performance is questionable. But for a quantitative evaluation of performance, some human judgments must be assumed to be correct. Second, it has an impact on the level of performance expected and targeted by the system. If humans do not agree on the appropriate diagnosis 100% of the time, then should a CAM implementation be expected to agree with the human judgments 100% of the time?
Carr et al. (2002) points out that typical human instructors may be unreliable because they

- can be inconsistently lenient or strict,
- may mistakenly accept incorrect answers given a target, and
- may construct incorrect target responses.

It is not reasonable to assume that human graders will never make mistakes. But even when multiple graders propose target responses and evaluate responses, agreement between graders on answers cannot be guaranteed for all types of questions. Voorhees and Tice (2000) discuss this issue in their work describing the process of developing an evaluation set for the question answering track of the TREC conferences. They point out that even seemingly obvious questions may have unanticipated answers. One example Voorhees and Tice describe comes from the to TREC question *Where is the Taj Mahal?*. The expected answer is *India*. However, if *Taj Mahal* is interpreted not as the famous mausoleum, but as the U.S. casino of the same name, an equally acceptable (though perhaps unexpected and less likely) answer would be *Atlantic City*. Thus, predicting the range of correct answers is not trivial, even for simple questions.

In their work, Voorhees and Tice found that on average, 6% of the answers were disagreed on by graders. Furthermore, they argue that even when two graders are trained and their agreement reaches 100%, once a third grader is introduced, more disagreements are likely because “different people have different definitions of correct” (Voorhees and Tice, 2000, pp. 205).

These observations are directly relevant to the current task. Different teachers may grade in different ways, with variation possible in their views on what is or is not a correct answer. If there is no universal definition of “correct,” then how can a system be evaluated? Voorhees and Tice argue that their research suggests one
solution may be *comparative* evaluations between systems. They point out that even if underlying human judgments were not stable, comparisons between systems were. That is, correlation between system rankings were very high, as much as 0.95 (where 1 is perfect correlation). While a comparative evaluation with other systems is beyond the scope of the current work, it is an interesting line of research to pursue.

Closer to the present task, inter-rater reliability has also been discussed in the automatic grading literature. Lee et al. (2007) compared human judgments in an essay scoring task. Eight coders (three native speakers and five bilingual speakers) scored a set of essays written by ESL learners for the TOEFL test. Each essay was graded twice, once by a native speaker and once by a bilingual speaker. To test inter-rater reliability, 50 of the essays were re-scored by two native speakers. Exact agreement on the assigned score (a number between 0 and 6) ranged between 81 and 89 percent, depending on the coders being compared. Kappa scores ranged between 0.51 and 0.66.3

Leacock (2004) also reported an evaluation of inter-rater agreement in the two pilot studies of the short-answer grader C-rater. The pilot studies were for automatic grading of short-answer questions on reading comprehension and algebra tests. One hundred responses for each item on the tests were scored by two human judges, who had been trained together on scoring but who independently scored each item. Their agreement rates were much better than those reported for the essay scoring: On the reading comprehension items, the average agreement between the human judges was 93% (kappa: 0.90). On the algebra test, it was slightly lower with 91.5% agreement (kappa: 0.86). One factor to consider in comparing agreement rates in essay and short-answer grading is that the short answers were given a score of 0 (no credit),

---

3 The kappa values refer to a statistic for measuring inter-rater reliability that takes into consideration not only whether the raters agree, but what the agreement rate would be by chance (Carletta, 1996).
1 (partial credit) or 2 (full credit), while essays were given a score between 0 and 6. Agreement may be higher on the short-answer task, in part, because there are fewer choices to make, in addition to the difference between the kind of activity being scored. Even within a particular activity, the specific test item can make a difference. For instance, Leacock and Chodorow (2003) note that for the C-rater evaluation of math questions, inter-rater exact agreement ranged from 87% to 94%, depending on the question, even though the graders were trained.

In the current study, to get a sense of the reliability of human judgments in grading such tasks during the development phase, the development set was graded by the author with three sets of judgments – a binary score (Y/N), a score from 0-2 and a diagnosis code, as the initial independent grader did. As held-out data, the test set was not graded by the author. However, the ESL instructor who provided the test data set also supplied binary judgments for the task, and these were compared to the independent grader for an evaluation of the reliability of detection.

On the development set, the agreement between raters is generally lower than in the C-rater study, as presented in Table 6.7. The count refers to the number of development set pairs that the raters agreed on (of 311). Exact refers to the proportion of exact agreement between raters. Given that kappa is often used when the rating task is both difficult and potentially subjective (Jurafsky and Martin, 2000), it is also included. Kappa is calculated as in (73).

\[
(73) \quad \kappa = \frac{P(A) - P(E)}{1 - P(E)},
\]

where \( P(A) \) is the proportion of actual agreement, and \( P(E) \) is the proportion of expected agreement given chance.

Agreement is presented in Table 6.7 for the three scales. Recall that the Y/N scale refers to agreement between the graders when the only decision was whether the answer was completely correct or not. The 0-2 scale agreement refers to when the
graders awarded the same number of points to a response: 0 (no credit), 1 (partial credit) or 2 (full credit). The diagnosis agreement reflects how often the graders agreed on the diagnosis code (of the six codes possible). It is not surprising that the agreement decreases as the number of choices for labels increases. It is far easier to agree on the appropriate label if the choice is between 2 labels (i.e., Yes or No) than six labels (i.e., diagnosis codes).

<table>
<thead>
<tr>
<th>Scale</th>
<th>Count</th>
<th>Exact</th>
<th>Kappa</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y/N Scale</td>
<td>263</td>
<td>0.85</td>
<td>0.66</td>
</tr>
<tr>
<td>0-2 Scale</td>
<td>252</td>
<td>0.81</td>
<td>0.61</td>
</tr>
<tr>
<td>Diagnosis Codes</td>
<td>240</td>
<td>0.78</td>
<td>0.56</td>
</tr>
</tbody>
</table>

Table 6.7: Inter-rater Agreement and Reliability on the Development Set

Perhaps also not surprisingly, it appears that the questions make a difference to agreement over the answers. For instance, looking at agreement between human judges for individual questions on a 0-2 scale, for 51% of the questions (24 of 47 questions), there is perfect agreement on how to grade all the responses. There is at least 80% agreement for 33 questions. For two questions, there is perfect disagreement.

Further analysis of disagreements reinforces the idea that question type is relevant to agreement. One main source of disagreement was how to treat questions that require a list answer (49% of the disagreements). Most of these questions asked for unspecified number of concepts. This raises the question of how many concepts must be included for a correct answer. For instance, one of those questions was What are the images that the article says an advertisement can make? If there were five possible images included in the textbook, but the student only lists 2, do they get full, partial credit or no credit? If they list 4 of 5? Different instructors might make different choices with respect to how to grade such activities.
Another main source of disagreement (about 30%) relates to the interpretation of some of the questions. For example, one of the questions is a Yes/No question, and there is no explicit requirement that students justify their answers. The instructor’s target response, however, is a justified no response. Should a grader count off for answers that are unjustified? Again, different instructors might make different choices for reasonably grading such questions.

Turning toward the test set, there was only one set of diagnosis judgments provided by the independent grader. However, the original instructor marked the 255 responses in the set as right or wrong. The agreement on a Y/N scale is presented in Table 6.8.

<table>
<thead>
<tr>
<th>Scale</th>
<th>Agreement</th>
<th>Exact</th>
<th>Kappa</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y/N Scale</td>
<td>224</td>
<td>0.88</td>
<td>0.63</td>
</tr>
</tbody>
</table>

Table 6.8: Inter-grader Agreement and Reliability in the Test Set

While the kappa value is lower than ideal, there are two points to be made about this statistic. First, even though kappa values are often seen in the literature on inter-rater reliability, there is no universally accepted interpretation of the kappa statistic. As Di Eugenio and Glass (2004) point out, some argue that any value below .67 is unreliable (Krippendorff, 1980), while others say $0.40 \leq \kappa \leq 0.60$ actually indicates moderate agreement (Rietveld and van Hout, 1993).

The second point is that Di Eugenio and Glass (2004) show that the kappa statistic is flawed when the underlying distribution of categories is skewed. That is, if the number of judgments in one category is significantly higher than the other categories, the value of the expected agreement will also be higher when compared to another more evenly distributed data set of the same size with the same total number of agreements between raters. The higher the expected agreement value,
the lower the kappa value. Thus, with skewed distributions, the kappa is low. And that is exactly the scenario in the current development and test sets. In both cases, the number of responses labeled correct (“Y”, 2, or “CA”, depending on the scale) far outnumbers the responses labeled as incorrect. This negatively affects the kappa value, even though the exact agreement is fairly high. Interpretation of any kappa score reported for this data set should take this into consideration. However, note that in a balanced subset of the test data, consisting of 72 responses (half labeled Y), the kappa value is 0.84.

6.1.4 Development and Test Data Overview

The level of agreement with respect to correctness judgments in the corpus had a clear impact on the development and evaluation of the CAM design. As discussed above, many of the disagreements could be attributed to issues related to the questions themselves. It is not the purpose of the current study to debate the design and interpretation of individual questions. Therefore, looking only at the responses the humans agree on avoids the need to determine which questions were “good ones” or which grader’s interpretation was better. However, because the corpus is small, eliminating too many pairs would not be useful either. To balance these concerns, the disagreements in Y/N judgments were used to eliminate several pairs, and the diagnosis disagreements in the development set were resolved in favor of the independent grader except in cases of clear grader error.\(^4\)

Thus, the final sets of target and learner response pairs used in developing and testing the CAM design are described by diagnosis code in Table 6.9.

\(^4\)Resolving diagnosis disagreements in the test set was not an issue because there was only one set of diagnosis judgments.
<table>
<thead>
<tr>
<th>Task</th>
<th>Judgment</th>
<th>Description</th>
<th>Development</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detection</td>
<td>Y</td>
<td>Correct</td>
<td>187</td>
<td>188</td>
</tr>
<tr>
<td></td>
<td>N</td>
<td>Incorrect</td>
<td>76</td>
<td>36</td>
</tr>
<tr>
<td>Diagnosis</td>
<td>CA</td>
<td>Correct Answer</td>
<td>172</td>
<td>173</td>
</tr>
<tr>
<td></td>
<td>AA</td>
<td>Alternate Answer</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>MC</td>
<td>Omission Error</td>
<td>43</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>EC</td>
<td>Overinclusion Error</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>MD</td>
<td>Blend Error</td>
<td>15</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>NA</td>
<td>Non-Answer</td>
<td>14</td>
<td>5</td>
</tr>
</tbody>
</table>

| Total Number of Response Pairs per Set | 263 | 224 |

Table 6.9: Breakdown of Data Sets by Judgment Labels

One final note to make is with respect to Alternate Answers. The approach adopted in CAM assumes that the learner responses will converge on the concepts represented by one of the target responses given for that question. But a learner response may be converging on a target not supplied for a question. An instructor might give such responses full credit, even though it was not the intended answer. Such alternate answers may not consistently be evaluated by CAM as correct. In developing the system, the 15 AA development set pairs were both included and excluded to determine whether inclusion helped or hindered performance. The findings will be discussed in Section 6.3, after an overview of evaluation metrics.

6.2 Evaluation Metrics

Several metrics have been used in ICALL, automatic grading and other similar classification tasks to evaluate system performance relative to a gold standard (i.e., a set of correct judgments). These common measures include precision, recall and F-measure; accuracy, correlation, and kappa.
To illustrate the evaluation metrics, consider the evaluation of detection – whether the learner response is acceptable/correct (Yes) or unacceptable/incorrect (No). Suppose CAM analyzed 100 responses that have gold standard judgments and that a contingency table showing the breakdown of Yes and No answers was constructed as in Table 6.10.

<table>
<thead>
<tr>
<th>CAM says ...</th>
<th>Gold answer was ...</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Yes</td>
<td>56</td>
</tr>
<tr>
<td>No</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 6.10: Hypothetical Performance Metric Example Data

*Precision* captures how often a system provides a correct judgment (e.g., correct Yes answers), relative to the total number of times the system made that judgment, whether correct or incorrect. In this example, precision would be the number of times CAM correctly said Yes, relative to the total number of times CAM said Yes, even if the correct answer was No. Thus, precision is 56/(56 + 16), or 0.78.

*Recall* is the total number of correct judgments, relative to the total number of times the system *should have* made that judgment. Intuitively, recall accounts for the judgments a system misses. In this example, CAM correctly said Yes 56 times, but should have said Yes 64 times (i.e., 56 + 8). Thus, recall is 56/64 (0.88).

Precision and recall are often inversely related. That is, when precision increases, recall decreases and vice versa. To account for the importance of both, *F-measure* combines precision and recall into a single score, calculated with the formula in (74) from Jurafsky and Martin (2000), which allows for weighting precision over recall or recall over precision. Assuming precision and recall are given equal weight, the F-measure of the hypothetical example is 0.82.

\[
(74) \quad \text{F-measure} = \frac{(\beta^2+1)(\text{Precision} \times \text{Recall})}{(\beta^2 \times \text{Precision}) + \text{Recall}}, \quad \text{where} \ \beta \ \text{is a weighting factor}
\]
Another metric to consider is accuracy. **Accuracy** is the number of times the system correctly answered Yes or No relative to the total number of times the system answers. Thus, accuracy is \((56 + 20)/100\), or 0.76, in the example. Accuracy is sometimes referred to as exact agreement (Marín, 2004). Adjacent agreement is sometimes reported in evaluation (e.g., Attali and Burstein (2006)), but is used when judgments are on a scale (e.g., 0-6 points) and agreement can take into consideration whether the judgment given by the system is within one scale value of the gold standard judgment. This metric is not applicable here.

Yet another frequently used metric is **correlation**, a measure of the relationship between two variables. One of the more common measures of linear correlation is Pearson Product Moment Correlation (also known as Pearson’s r). The Pearson’s r score is a value between +1 and -1. A positive correlation means the system’s judgments are in line with the human judgments in the gold standard. That is, the closer the correlation is to 1, the stronger the positive relation and the more likely that if the human gives a high score, the system will also give a high score. The formula for calculating Pearson’s r is given in (75), where X and Y are observed values for the variables, and N is the number of instances (Weiss, 1993). For the hypothetical data (where Yes is given a value of 1 and No a value of 0), the Pearson’s r is 0.46, only a weak positive correlation. Because the values for judgments are binary, the correlation is a special case of Pearson’s r known as the Phi Coefficient.

\[
(75) \quad r = \frac{\sum XY - \frac{\sum X \sum Y}{N}}{\sqrt{(\sum X^2 - \frac{(\sum X)^2}{N})(\sum Y^2 - \frac{(\sum Y)^2}{N})}}
\]

The final metric, kappa, has already been defined in (73) during the discussion of inter-rater reliability in Section 6.1.3. For the hypothetical example given above, the kappa score would be 0.45.
Of these metrics – all commonly used in classification evaluation tasks – precision and recall (and, thus, F-measure) are less ideal for the current evaluation. In this case, reporting accuracy is preferable at least for detection, because it takes into consideration all the correct judgments, not just the correct Yes judgments or correct No judgments. Moreover, Pearson’s r would be suitable as a measure of agreement for detection but not diagnosis because in the latter case, the data are nominal, invalidating the use of Pearson’s r. Kappa, while useable with nominal data, suffers from the problem of the underlying distribution of data in the classes, as discussed in Section 6.1.3. Thus, to be consistent across evaluations, the accuracy score is used for reporting in the detection and diagnosis experiments.

While evaluation metrics are useful for getting a sense of performance, it is also important to note that it is only one piece of the puzzle. That is, the CAM design should eventually be evaluated with respect to the application it is embedded in or the benefit to the end user. For instance, the diagnosis codes are intended to be linked to feedback messages. One external evaluation of the design might be to measure how effective such messages are in guiding the learner to recognize or correct errors. Given that integrating the CAM design into a larger ICALL system is beyond the scope of the current study, this is left to future research.

Regardless of the evaluation metric selected, there is still an issue of what to compare the results to. Given that other ICALL or automatic grading systems have been applied to different data sets using different evaluation standards with different goals in mind, no direct comparison with other systems is currently possible. Thus, there are no previous results to aspire to and no well-established baseline to outperform. To address the latter, two possible ways to construct baseline results might be to i) simply randomly assign detection/diagnoses values to learner responses or ii) assign the same detection/diagnosis value to all responses.
For detection, where the classification is binary – Y or N, randomly assigning a detection code establishes a baseline for detection performance at 50%. Alternatively, a second baseline might be constructed by always labeling a pair Y. This would be equivalent to a hypothetical system that always gives the student the benefit of the doubt and marks every answer as correct. However, given that the underlying distribution of Y labels in the development set is skewed, this would produce a baseline of 71% accuracy. That is, 187 of 263 pairs (71%) were labeled Y by humans. Using this baseline as a measure of success is questionable because the underlying assumption that humans will (and, therefore, the system should) always grade lightly is shaky. Thus, the system should not rely on the presence of more positive than negative examples in learning to detect and diagnose errors.

For diagnosis, as will be discussed below, there are five diagnosis codes – CA (Correct), EC (Overinclusion), MC (Omission), MD (Blend), and NA (Non-answer). Random assignment of diagnosis codes would produce a baseline of 20%.

6.3 Revisiting the Diagnosis Component

Chapter 5 left the diagnosis component as a black box for detection and diagnosis. This section opens the box to discuss how the classifiers were trained, beginning with motivating the need for a machine learning approach to detection and diagnosis.

6.3.1 Manually Tuned Detection and Diagnosis

Recall that in development, the first CAM implementation used the relative overlap of concepts in the learner and target responses with manually tuned thresholds to detect and diagnose errors. Using this strategy, henceforth referred to as manual CAM, the approach obtains 81% accuracy on the development set.
Note that if the detection algorithm were tuned to answer $Y$ most of the time, it would be expected that on a balanced corpus, the performance would suffer. However, when a 152-pair balanced subset of the data is selected, the performance of CAM drops only slightly to 78%. Thus, manual CAM was not overly tuned to respond with a $Y$ label.

However, it was overly tuned to the development data. Performance on the test set illustrates this. On the 224 test set pairs, the manual CAM only correctly labeled the pair as correct or incorrect in 141 instances. The performance, while still better than the baseline of 50%, drops significantly to 63%. It was this drop in performance that led to a rethinking of how to detect (and ultimately diagnose) errors. In the manually tuned CAM, the system output was a diagnosis code, which was used to generate a $Y/N$ label. The results of the manual CAM suggest that it overdiagnosed the presence of overinclusion (EC) errors, causing the detection and diagnosis performance drop. In turn, this overdiagnosis suggests that the learner responses in the test set contain more unaligned (but not necessarily incorrect) concepts than the development set data.

One possible conclusion from the manual CAM results for detection is that relative overlap of concepts alone is an insufficient indicator of appropriate diagnosis. However, this is not the only feature that is identified by CAM. Other features, such as overlap of chunks and triples, as well as the types of alignments, may be more relevant indicators. And given the range of features collected, the nature of the problem lends itself naturally to exploring a machine learning solution to determine more reliably how the different pieces of evidence should be combined. While there the data set is not large enough for extensive evaluation of a machine learning approach, it is sufficient to explore the feasibility of such a solution.
6.3.2 Machine Learning for Detection and Diagnosis

There are a few preliminaries to address regarding the development of classifiers for detection and diagnosis. First, as discussed in Chapter 5, the TiMBL suite includes numerous options that alter how features are weighted, how many nearest neighbors are considered in measure similarity, and how similarity between examples are measured. Various combinations were tried to maximize performance on the development set.

Second, performance on the development set was measured using a leave-one-out strategy. That is, rather than split the development into even smaller sets for training and testing, the development experiments held out one pair for testing and trained on the remaining pairs. This process was repeated for every pair in the set. The overall score was a then measure of the average accuracy for all the development pair instances.

Even so, given the small size of the training sample, it was expected at the outset that performance on detection would be higher than for diagnosis, not only because choosing between two labels is a simpler task than choosing between 5 or 6, but also because the training set for diagnosis consist of fewer instances labeled with each diagnosis code.

Independent of the TiMBL options available, experiments with training classifiers explored the following:

- voting to select the most likely detection/diagnosis value,
- eliminating alternative answer pairs in the training set,
- using the detection value in diagnosis training, and
- reducing the diagnosis classifiers to binary decisions.
6.3.2.1 Detection/Diagnosis Voting Using Different Distance Metrics

There are seven options for the distance metric TiMBL to measure similarity between instances. These are given in Table 6.11.\textsuperscript{5} While Daelemans et al. (2007) point out that data sparsity is a problem for some of the options, such as modified value difference and Jeffrey divergence, it is not clear a priori which of these will perform the best on the development set or a much larger training/test set. Thus, the top-performing TiMBL configuration for the development set is not necessarily going to be the top performing configuration for the test set.

Rather than selecting a single, best-performing combination from the development set to use on the test set, a simple alternative is to train several classifiers and let them vote on the most likely detection or diagnosis class. In this way, the differences in performance of metrics, which may reflect the small instance base, is somewhat mitigated. Thus, the top scoring run from each of seven similarity metrics offered by TiMBL were used to evaluate the performance of a voting scheme. For binary detection, there is an odd number of votes, and therefore no ties. For the sake of simplicity, the majority vote rules. For diagnosis, again the top performing run for each of the seven similarity metrics was selected for final voting. Because there were several diagnosis codes to choose from, the possibility of ties existed. To break ties, the value of the detection feature was used to label the pair either \textit{CA} if the detection value was \textit{Y} or \textit{MC} (the most common error label in the training set) if the detection feature had the value \textit{N}. All the results reported in the remainder of the chapter used this approach to selecting the final detection/diagnosis value for any given pair.

\textsuperscript{5}Descriptions are summarized from Daelemans et al. (2007). Refer to the TiMBL manual for complete metric descriptions.
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>O (Overlap)</td>
<td>The sum of feature value differences, where difference is binary (1 or 0) for nominal data and a ratio if the data are numeric.</td>
</tr>
<tr>
<td>L (Levenshtein)</td>
<td>The sum of feature value differences, where difference is measured in terms of edit distance between feature values.</td>
</tr>
<tr>
<td>N (Numeric Overlap)</td>
<td>The sum of feature value differences, where difference is defined as in Overlap, but all feature values are required to be numeric.</td>
</tr>
<tr>
<td>M (Modified value difference)</td>
<td>The sum of feature value differences, where difference between values is defined as the difference between the conditional class probabilities for the given values.</td>
</tr>
<tr>
<td>J (Jeffrey divergence)</td>
<td>The sum of feature value differences, where difference is measured similar to option M, except the formula for calculating the conditional class probabilities includes a logarithm term.</td>
</tr>
<tr>
<td>D (Dot product)</td>
<td>The difference between the maximum dot product for an exact match and the sum of dot product values for feature value pairs.</td>
</tr>
<tr>
<td>C (Cosine)</td>
<td>The difference between the maximum cosine value and the ratio of the dot product metric and the product of vector lengths.</td>
</tr>
</tbody>
</table>

Table 6.11: TiMBL Distance Metrics

6.3.2.2 Eliminating Alternate Answers in Training

Analysis of the development set suggested that eliminating AA pairs (i.e., pairs with alternate answers diagnosis) from training might improve performance. Consider the alternate answer pair in (76). Given the reading, the learner response is perfectly acceptable as an answer to the given cue.
a. **Question Cue:** What is the excuse that the paparazzi use to justify their actions?

b. **Target Response:** Paparazzi say the people are public figures.

c. **Learner Response:** The news value of a picture is more important than the person’s right to peace and privacy.

The motivation for eliminating such pairs is that while these pairs are correct, they tend to be very dissimilar to any of the target responses provided. Thus, the feature data collected could be unreliable in detecting correctness or diagnosing alternate answers as such. Eliminating these pairs reduced the development set to 248 pairs. However, as the results in Table 6.12 suggest, performance does improve, if only slightly for detection. Cells of the table contain the overall accuracy scores, which show that training on alternate answers does negatively affect performance.

<table>
<thead>
<tr>
<th></th>
<th>Detection</th>
<th>Diagnosis</th>
</tr>
</thead>
<tbody>
<tr>
<td>With AA Pairs</td>
<td>86%</td>
<td>75%</td>
</tr>
<tr>
<td>Without AA Pairs</td>
<td>87%</td>
<td>79%</td>
</tr>
</tbody>
</table>

Table 6.12: Development Set Performance with(out) AA Pairs

One solution to the problem of alternate answers, to be discussed further in Chapter 7, is to expand the set of target responses considered as references. But given these results, the remaining tests reported reflect training on the development set without the AA pairs included.

6.3.2.3 Using Detection in Diagnosis

Detection and diagnosis have been separated to this point based on the assumption that detection is a “simpler” task. But knowing whether a pair is labeled correct or incorrect is also highly relevant for diagnosis. To explore the effectiveness of feeding
the output of the detection classifier to the diagnosis classifier, a detection code was added as a fourteenth feature value for each pair before training the diagnosis classifier. Table 6.13 shows a comparison of performance on the development set, with and without the fourteenth feature.

<table>
<thead>
<tr>
<th></th>
<th>Overall Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without Detection Feature</td>
<td>79%</td>
</tr>
<tr>
<td>With Detection</td>
<td>83%</td>
</tr>
<tr>
<td>With Perfect Detection</td>
<td>87%</td>
</tr>
</tbody>
</table>

Table 6.13: Development Set Diagnosis Performance with(out) Detection Feature

The last row of the table requires additional explanation. For any pair in the training set, the value of the detection feature can be either the output of the detection classifier or the human judgment of correctness. In the former case, the detection classifier could make a mistake. For instance, a pair might incorrectly be labeled with a $Y$ in detection, and this misinformation would then be used in training the diagnosis classifier. The alternative is to use the human judgments (i.e., “perfect” detection) in training. The results of exploring both possibilities from Table 6.13 show that, as expected, performance drops with imperfect detection. However, perfect detection judgments on unseen data would presumably not be available to the diagnosis classifier. Thus, it is perhaps more realistic evaluation of diagnosis performance on the development set to train the diagnosis classifier on the output of the detection classifier, rather than on human judgments. Regardless, adequate detection seems to improve diagnosis, and the remaining accuracy scores reported for the development and test set use separate detection and diagnosis classifiers. The output of detection feeds into the input of diagnosis.
6.3.2.4 Reducing Diagnosis to Binary Decisions

Given that the detection and diagnosis classifiers were split, resulting in an increase in performance, the possibility of dividing the diagnosis classifier into multiple classifiers was also explored. Specifically, a separate diagnosis classifier was trained for each of the four error classifications. Each classifier labeled a pair as a specific error or as correct, reducing diagnosis to a series of binary classifications. These classification were combined to produce a final diagnosis. However, the results on the development set for binary classifiers were identical to that of the single diagnosis classifier. Thus, this line of exploration was not extended to the test set evaluation.

6.3.2.5 Summary of Classifier Development

The final configuration for training the classifiers for the implementation of the CAM design took into account the following assumptions based on experiments with the development set.

- Voting on labels with classifiers using different distance metrics gives a better sense of the possible performance of the classifiers. With a larger instance base, the combination of distance and weighting options to use might be clearer. In the absence of that, voting provides a good measure of potential performance.

- Detection can facilitate diagnosis classification. Using the output of detection as a feature value in diagnosis provides the system additional information relevant to effective diagnosis, and should be exploited for more accurate diagnosis. Splitting off detection may be useful in providing feedback as well. If the system is unsure of the accuracy of diagnosis (e.g., there is a tie between two error diagnoses), then the feedback manager could always fall back to a less-specific response to the student based only on detection.
• The cleaner the training data, the better the results. Removing the alternate answer pairs and using perfect detection information improved performance. In the latter case, the performance obviously would improve with perfect detection information because incorrect diagnoses based on bad detection information would be eliminated.

Given these observations, the top performing configuration for the development set was used to train classifiers for the CAM implementation. Results on evaluating the test set are described in the next section.

6.4 Evaluation Against the Test Set

Before discussing the performance of the CAM implementation, the dependence of performance on the underlying test set should be underscored. Pérez et al. (2005b) note that overall performance of automatic grading systems varies widely from 30% to 93% agreement between systems and human judges because the underlying corpora used to test the systems vary widely in difficulty. Thus, the performance of the CAM implementation may vary widely, depending on the kinds of test pairs used as an evaluation set. However, given that the tasks used to generate the development and test sets were very similar, the results are comparable.

Also note that there is simply not enough labeled pairs for conclusive quantitative evaluation of the diagnosis classifier. However, the results are encouraging. First, consider the results for detection presented in Table 6.14.

As previously discussed, the performance of the manually tuned CAM drops significantly from development to test set. In contrast, the performance of the final CAM system with the detection and diagnosis classifiers improved slightly from development to test set, suggesting that the classifiers learned how to diagnose without
over-fitting the training data. The final accuracy score for meaning error detection is 88%, significantly higher than both the baseline of 50% and the manual CAM performance of 63%.\(^6\) These results suggest that detection using the CAM design is viable, though more extensive testing with a larger corpus is needed. Refer to Chapter 7 for a discussion of this issue.

Turning to diagnosis, the overall performance of the diagnosis classifier is presented in Table 6.15. Given that the number of labels increases from 2 to 5, the slight drop in overall performance in diagnosis as compared to the detection of semantic errors (from 88% to 87%) is both unsurprising in the decline and encouraging in the smallness of the decline. However, given the sample size and few numbers of instances of any given error in the test (and development) set, additional quantitative analysis of the results would not be particularly meaningful.

\begin{table}[h]
\centering
\begin{tabular}{|l|c|}
\hline
\textbf{Development} & \textbf{Accuracy} \\
\hline
Development & 87\% \\
Test & 87\% \\
\hline
\end{tabular}
\caption{Diagnosis Performance}
\end{table}

\(^6\)As with the development set, removing the alternate answer (AA) pairs from the evaluation set improves results, though only slightly to 89% accuracy. However, there is no way to determine beforehand which pairs are AA pairs. Thus, the final evaluation does not exclude these pairs.
It is important to keep in mind that the higher the accuracy scores, the better, but looking solely at the numbers misses the point. In order to assess the strengths and weaknesses of the approach, it is just as important to get a sense of how the diagnosis judgments are made (whether right or wrong) and the nature of the underlying data that leads to those decisions. The remainder of this section focuses on such concerns.

6.4.1 Correct Diagnosis Examples

To launch the qualitative discussion of semantic error diagnosis, consider the pairs in (77) and (78), both correctly diagnosed by the CAM implementation as correct responses.

(77) a. **Question Cue**: What is the definition of socialization?
    
    b. **Target Response**: Socialization is learning how to behave in society.
    
    c. **Learner Response**: The process of learning what to expect and how to behave in the society the individual lives in.
    
    d. **Diagnosis**: Correct (CA)

(78) a. **Question Cue**: What is the major difference in TV and radio versus the internet concerning censorship?
    
    b. **Target Response**: On TV or radio there are editors to check for accuracy or appropriateness and restrictions on when programs can be broadcasted.
    
    c. **Learner Response**: TV and radio can check the programs and there are restrictions on what kinds of programs can broadcast, but Interenet cannot check.
    
    d. **Diagnosis**: Correct (CA)

In both examples, the system correctly recognizes the similarities between the learner and target responses. In the learner responses in (77), there are several
unnecessary concepts, including *expect, individual, lives,* and *process,* but the learner is not penalized. In (78), the correct diagnosis is made in spite of grammatical and spelling errors. The misspelled *Internet* is correctly aligned to *internet* in the question cue and eliminated as given information before alignment.

(79) a. **Question Cue:** What is the definition of sanctions?
   b. **Target Response:** Sanctions are consequences that influence whether the behavior will be repeated.
   c. **Learner Response:** Sanctions are results from rewards
   d. **Diagnosis:** Blend (MD)

(80) a. **Question Cue:** What are the concerns (about the Internet)?
   b. **Target Response:** The concerns of the internet include the lack of censorship, privacy, control of communication, misuse of the internet in the workplace and internet addiction.
   c. **Learner Response:** One concern relates to a lack of censorship or control over what appears on the Internet.
   d. **Diagnosis:** Omission (MC)

(81) a. **Question Cue:** Describe the difference found between the Kenyan and Mexican communities studied AND the communities in the United States.
   b. **Target Response:** Kenyan and Mexican children learned quickly to be responsible and caring through household chores and caring for younger siblings. Children in the United States were less likely to develop those traits early because they only had chores such as cleaning rooms.
   c. **Learner Response:** The children of K and M quickly learn to take care of their younger siblings and help the whole household.
   d. **Diagnosis:** Omission (MC)
The pairs in (79), (80) and (81) are examples of correctly diagnosed errors. Example (79) conveys different (and incorrect) concepts given the target response and is diagnosed as a blend error. Examples (80) and (81) are omission errors. Note that in the test set, there were no instances of overinclusion (EC) errors and none of the five non-answer (NA) instances were correctly diagnosed.

In both omission examples, the learners seem to be aiming for the target responses but fall short of conveying all the necessary concepts. The two examples differ in the types of question cue presented to the learner. In (80), the question asks for a list of concerns people have about Internet use. The text includes five concerns, but the learner response only conveys two of the five. Rather than a list, the question in (81) asks for a response contrasting the Kenyan and Mexican communities with American societies. In addition to inappropriately abbreviating Kenyan and Mexican societies as K and M, respectively, the learner response fails to explicitly mention how American society differs from them.

6.4.2 Incorrect Diagnosis Examples

Of the 30 errors in diagnosis, only 4 (13%) are mix-ups between different error diagnoses. Most of the diagnosis errors (53%) occurred when the system incorrectly labeled a pair as correct when should have had an error diagnosis instead. Given that detection is imperfect, it makes sense that the system might under-diagnose errors. A summary of all the diagnosis results is presented as a contingency table, Table 6.16.

An example of a misdiagnosis in which the answer was labeled as correct, but was in fact an omission error, is given in (82). The question has two parts and the learner failed to answer both parts by mentioning socializing agents as the targeted term. The learner response also contains extraneous material, but that material is not incorrect.
Table 6.16: Contingency Table of Diagnosis Results

<table>
<thead>
<tr>
<th>CAM</th>
<th></th>
<th>Correct Diagnosis</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CA</td>
<td>EC</td>
</tr>
<tr>
<td>CA</td>
<td>178</td>
<td>0</td>
</tr>
<tr>
<td>EC</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MC</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>MD</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>NA</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

(82) a. **Question Cue:** What is the term that is used to describe the people and groups that try to influence us and socialize us? List them.

b. **Target Response:** Socializing agents are family, school and peer groups.

c. **Learner Response:** Every society tries to socialize its members. The family, the school, and the peer group are the most important socializing.

d. **Diagnosis:** Correct (CA)

e. **Actual:** Omission (MC)

Another example of an incorrect response that was mislabeled as correct is in (83). As with (82), the concept overlap is high, but in this case, the learner response is wrong because, perhaps through ungrammaticality, it conveys the incorrect meaning.

(83) a. **Question Cue:** Who uses peer pressure in advertising and why?

b. **Target Response:** Soft-drink companies use peer pressure to persuade people to buy a product because it is popular.

c. **Learner Response:** Propagandist who use soft-drink companies on young audiences because it is seen as popular.

d. **Diagnosis:** Correct (CA)

e. **Actual:** Blend (MD)
The remaining examples are cases in which the system found connections between responses where none existed. Of these, two are examples in which the CAM implementation recognized an error but gave the wrong diagnosis. Both (84) and (85) should be labeled as non-answers. In (84), *people* is aligned to *everyone* through similarity scoring and *because* aligns though a token match. Given that all the concepts are mapped in the learner response, but there unmapped target concepts, an omission error diagnosis is not surprising.

(84) a. **Question Cue**: Who uses peer pressure in advertising and why?
   b. **Target Response**: Soft-drink companies use peer pressure to persuade people to buy a product because it is popular.
   c. **Learner Response**: everyone because
   d. **Diagnosis**: Omission (MC)
   e. **Actual**: Non-answer (NA)

(85) a. **Question Cue**: In your own words, explain the purpose of the three bulleted statements at the beginning of the reading. What is their purpose? Why did the writer put them there?
   b. **Target Response**: To give examples of socialized behavior.
   c. **Learner Response**: Being polite, take care of themselves and learning of responsibility
   d. **Diagnosis**: Correct (CA)
   e. **Actual**: Non-answer (NA)

While the learner response in (85) is also a non-answer as in (84), it differs in that the misalignments are further afield. That is, while the learner response is seemingly unrelated to the target response, unconnected concepts are nevertheless incorrectly related through synonym and similarity matching (e.g., *give* and *take* are aligned as synonyms). These misalignments result in the incorrect CA diagnosis.
The final example diagnosis error, in (86), is a case in which the grader accepted the answer as correct, but the system labeled it an omission error. The overlap is high, but the concepts missing from the learner response were not enough for the human to label it as an error, though there is less information conveyed by the learner response. More will be said in Chapter 7 about seemingly non-critical concepts in target responses.

(86) 

a. **Question Cue**: What are the three factors that limit the influence of media on public opinion?

b. **Target Response**: The factors are independent organizations promote other points of influence or cancel out another influence. A second is profit, which often leads media owners to give their customers what they want to see and hear. The third is a two-step process involving hearing the news and accepting or rejecting it based on opinions of leaders.

c. **Learner Response**: One is the fact that independent organizations can present us with different points of view. A second is that media owner are interested in making a profit. The third is that there is often a two-step process of influence.

d. **Diagnosis**: Omission (MC)

e. **Actual**: Correct (CA)
6.4.3 Form Errors and Semantic Error Diagnosis

Given the examples of correct and incorrect diagnoses presented in previous sections, one consideration is the relationship between form and meaning errors. That is, to what extent do form and meaning errors go hand-in-hand? To explore this question, form errors were counted in the 224 learner responses of the evaluation set.\(^7\) Form errors were defined as

- **Spelling errors**: Non-word spelling mistakes (e.g., *creat* instead of *create*).
- **Lexical errors**: In-context word choice errors (e.g., *put* instead of *post, by* instead of *buy*). In order to avoid deciding between when a real word is a spelling error or a word choice error, all real-word errors were counted as lexical errors, except cases in which it is clear that the error was morphological.
- **Morphological errors**: Mistakes in inflection and derivation (e.g, *behave* instead of *behavior, child* instead of *children*). A mistake in inflection might result in an ungrammatical sentence, but each error is only counted once, as morphological errors.
- **Syntactic errors**: Grammatical mistakes, such as omitting needed words (e.g., articles), inserting unneeded function words (e.g., wrong prepositions), and making word order mistakes.

In all cases, a linguistic expression was counted as a type of error if the shortest number of changes to fix the mistake involved that type of error. For instance, consider the response in (87).

(87) *Since six years, the children will begin to learn respectful for the parnts.*

\(^7\)The 224-pair subset of the 255-pair test data were used because the judgments for these pairs were agreed upon by humans, which is relevant to the discussion that follows.
There are several errors in the responses and a number of ways to fix them. For instance, the expression *respectful for* could be replaced by *to respect*, but this would require syntactic and morphological change. A “shorter” fix would be to replace *respectful* with *respect*, a single morphological change. Thus, in one sense, error counting was based on a kind of edit distance. In addition, more than one instance of an error might be found in a single response. But the numbers reported below reflect types, not instances. That is, if a response contains four spelling mistakes, those mistakes only count once. Table 6.17 shows the breakdown of errors by type for the response in the test set, as compared to the 311-pair development set.

<table>
<thead>
<tr>
<th>Errors</th>
<th>Development Set</th>
<th>Test Set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Count</td>
<td>Percent</td>
</tr>
<tr>
<td>Spelling</td>
<td>33</td>
<td>11%</td>
</tr>
<tr>
<td>Lexical</td>
<td>35</td>
<td>11%</td>
</tr>
<tr>
<td>Morphological</td>
<td>28</td>
<td>9%</td>
</tr>
<tr>
<td>Syntactic</td>
<td>74</td>
<td>24%</td>
</tr>
<tr>
<td>Any Form Error</td>
<td>115</td>
<td>37%</td>
</tr>
</tbody>
</table>

Table 6.17: Errors by Type and Data Set

The percentages reported reflect the percent of responses in the given set (development or test) containing at least one instance of that type of error. For instance, there were 28 learner responses of the 311 in the development set (or 9%) that had at least one morphological error. Overall, for the entire corpus, there was at least one form error of any kind in 262 learner responses, or 46% of the corpus. Across data sets, there was an increase in form errors in learner responses from development to test set, but different students produced the responses for the development and test sets and students’ English skills may have varied across the sets. Focusing in on the test set numbers, it is interesting to see that the number of responses that contained some kind of error is high. Nearly half of all the learner responses contained syntactic
errors. And when combined into a single form error count, it turns out that 66% (147 responses) of the test set responses contained at least one form error, regardless of whether that response contained a meaning error. Of these, there are often several instances of form errors within responses. There were an average of 2.7 instances of form errors per response.

The breakdown of form errors by the presence of meaning errors (i.e., whether the meaning conveyed was correct or incorrect as judged by humans) is presented in Tables 6.18 and 6.19. Given that the number of correct responses far outnumbered the incorrect responses, the percents are perhaps more meaningful than absolute counts. For instance, 51% of all correct responses contained at least one syntactic error, as compared to only 39% of the incorrect responses. For correct and incorrect responses, the percentage of syntactic error was greater than morphological errors, which, in turn, was greater than lexical and spelling errors. Looking at overall numbers of errors, Table 6.19 suggests that correct responses had proportionally more form errors than incorrect responses, with 68% of all correct responses containing at least one form error, as compared to 53% of the incorrect responses. In other words, correct responses had more form errors than incorrect responses. Note that in a balanced subset of correct and incorrect responses, these findings also hold.

<table>
<thead>
<tr>
<th>Form Errors</th>
<th>Correct Count</th>
<th>Correct Percent</th>
<th>Incorrect Count</th>
<th>Incorrect Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spelling</td>
<td>44</td>
<td>23%</td>
<td>5</td>
<td>14%</td>
</tr>
<tr>
<td>Lexical</td>
<td>48</td>
<td>26%</td>
<td>4</td>
<td>11%</td>
</tr>
<tr>
<td>Morphological</td>
<td>62</td>
<td>33%</td>
<td>9</td>
<td>25%</td>
</tr>
<tr>
<td>Syntactic</td>
<td>95</td>
<td>51%</td>
<td>14</td>
<td>39%</td>
</tr>
</tbody>
</table>

Table 6.18: Breakdown of Form Errors by Meaning Correctness
Table 6.19: Breakdown of Presence of Form Errors by Meaning Correctness

<table>
<thead>
<tr>
<th>Form Errors</th>
<th>Correct</th>
<th></th>
<th>Incorrect</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Count</td>
<td>Percent</td>
<td>Count</td>
<td>Percent</td>
</tr>
<tr>
<td>None</td>
<td>60</td>
<td>32%</td>
<td>17</td>
<td>47%</td>
</tr>
<tr>
<td>At least one</td>
<td>128</td>
<td>68%</td>
<td>19</td>
<td>53%</td>
</tr>
<tr>
<td>Total</td>
<td>188</td>
<td>100%</td>
<td>36</td>
<td>100%</td>
</tr>
</tbody>
</table>

While in the test data, the mere presence of a form error is not indicative of a likely meaning error, is there a connection between the number of instances or combinations of form errors and the presence of a meaning error? In the case of error instances, the answer is No. There were 35 responses (16% of the data set) that had 4 or more instances of errors in each response. Of these, none were incorrect. Looking at combinations of errors, such as a response with grammatical and lexical errors, no clear pattern emerges that would suggest that form errors are linked to meaning errors in a clear way.

Thus, one conclusion to draw based on these data is that form and content assessment can be treated as distinct in the evaluation of learner responses. Even in the presence of a range of form-based errors, human graders can clearly extract the intended meaning to be able to evaluate semantic correctness.

However, the extent to which there is a relationship between form errors and misdiagnosis by the system should be considered. Table 6.20 shows numbers and percentages of correct and incorrect diagnoses by the CAM implementation for responses with and without form errors. The implementation misdiagnosed 13% of the responses. Of these, 47% had no form errors, 53% had at least one form error. The relatively small difference between these percentages suggests that there is not a direct relationship between the presence of form errors and the inability of the system
to produce an accurate diagnosis. In other words, the presence of form errors did not directly inhibit the ability of the NLP tools to process learner language for diagnosis.

<table>
<thead>
<tr>
<th>Form Errors</th>
<th>Correct Diagnosis</th>
<th>Misdiagnosis</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Count</td>
<td>Percent</td>
</tr>
<tr>
<td>None</td>
<td>66</td>
<td>34%</td>
</tr>
<tr>
<td>At least one</td>
<td>128</td>
<td>66%</td>
</tr>
<tr>
<td>Total</td>
<td>194</td>
<td>100%</td>
</tr>
</tbody>
</table>

Table 6.20: Overview of Form Errors and System Diagnosis

6.4.4 Alignment Properties and Diagnosis

Given the small number of diagnoses for any given error type in the test set, it is difficult to analyze data patterns in order to produce conclusive results. However, Table 6.21 presents the average values for the features of the test pairs used in training the detection and diagnosis classifiers.

<table>
<thead>
<tr>
<th>Features</th>
<th>Actual Diagnosis</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Correct</td>
</tr>
<tr>
<td>1. Keyword overlap</td>
<td>0.96</td>
</tr>
<tr>
<td>2. Aligned target tokens</td>
<td>0.79</td>
</tr>
<tr>
<td>3. Aligned learner tokens</td>
<td>0.47</td>
</tr>
<tr>
<td>4. Aligned target chunks</td>
<td>0.78</td>
</tr>
<tr>
<td>5. Aligned learner chunks</td>
<td>0.48</td>
</tr>
<tr>
<td>6. Aligned target triples</td>
<td>0.49</td>
</tr>
<tr>
<td>7. Aligned learner triples</td>
<td>0.31</td>
</tr>
<tr>
<td>8. Token matches</td>
<td>0.63</td>
</tr>
<tr>
<td>9. Similarity matches</td>
<td>0.27</td>
</tr>
<tr>
<td>10. Type matches</td>
<td>0.01</td>
</tr>
<tr>
<td>11. Lemma matches</td>
<td>0.04</td>
</tr>
<tr>
<td>12. Synonym matches</td>
<td>0.02</td>
</tr>
<tr>
<td>13. Variety of matches</td>
<td>2.18</td>
</tr>
</tbody>
</table>

Table 6.21: Average Values for Machine Learning Features for Diagnosis
As the table suggests, the higher the relative keyword overlap, the more likely that the diagnosis should have been correct (i.e., Y for detection or CA or AA for diagnosis). The same is true, though to a lesser degree, for the alignment of target response tokens, chunks, and triples. In other words, the more token positions and noun phrase chunks in the target response that were aligned to positions/chunks in the learner response, the more likely the response was to be labeled correct.

However, by a small margin, the reverse appears to be true of the data set for token or chunk alignments of the learner response. Responses that should have received an incorrect diagnosis tended to have a higher relative alignment for these learner response feature values. This might indicate that there were false alignments of learner response features as was the case in the example in (85). But it also may indicate that incorrect learner responses were very similar to the target as in the examples in (82) and (83).

The trends appear less clear for most of the types of matches, except for token and similarity. For token matches, correct responses had, on average, a higher percentage of token-identical alignments. Correct learner responses often used many of the same words as the targets. For similarity matches, the average relative similarity-based alignments were higher for incorrect responses. Given that similarity alignments may link terms that are less semantically related than the other types of matches, this is not surprising. Chapter 7 will discuss possible improvements to the alignment modules.

6.5 Summary

The combination of features discussed in the previous section were able to obtain 88% accuracy on detecting meaning errors in the test set. While there are no existing systems to compare these results to, the accuracy is reasonably high. The closest
related system that does a similar kind of detection is the C-rater system (Leacock, 2004), as described in Chapter 2. That system obtains 85% accuracy, but the test set and scoring system were different, so the results are not directly comparable. In addition, their work focused on detection of errors rather than diagnosis. However, the detection results are very competitive. Still, as stated at the outset, the ability of the features outlined in Chapter 5 to effectively diagnose particular errors is unclear. Analysis of the test set data suggests that the approach does make reasonable decisions about what to align and when, but the next chapter will discuss how translate the observations from this chapter into future approaches to diagnosis.
Language learners often find unexpected ways to express intended meaning. This variation in linguistic form is readily apparent when looking at the extent to which learner responses varied from target responses in the corpus described in the previous chapter. In the development set, only one of the 311 pairs were string identical. In the test set, there were no instances of string-identical pairs. When pairs are treated as bags of tokens, the number of identical pairs increases to only nine in the development set and remains zero for the test set. In other words, the most common approach to meaning comparison in ICALL systems – character or token-based string matching – would be almost entirely ineffective for these learner responses. Thus, the corpus illustrates nicely that loosely restricted responses to language exercises need more sophisticated approaches to content assessment.

A few ICALL systems (c.f. DeSmedt, 1995; Dorr et al., 1995) have approached this issue of content assessment for unconstrained responses by providing formal representations of learner responses through deep semantic analysis. Such systems have the disadvantage of being restricted to a small domain and (usually) a single activity type. They are time-intensive to create and equally time and labor intensive to port to new domains and languages. On the other hand, systems that rely on so-called “shallow” representations of semantic information have the advantage of being easier to transfer to different domains and less difficult to create for new languages, if
the basic language processing tools are available. Thus, this thesis has explored an approach to content assessment in ICALL that relies on shallow natural language processing and techniques from other computational linguistics domains.

The primary source of material for this exploration into ICALL content assessment was a new corpus of language learner data. At the time of writing, this corpus is one of the few publicly available language corpora and, as far as the author knows, the first to be annotated with semantic error diagnoses.\footnote{Contact the author at s.bailey@ling.osu.edu for details on access to the corpus.} As described in Chapter 6, the diagnoses were developed from analyzing part of the corpus and adapting an annotation scheme based on target modification to annotate the full corpus. This corpus provided invaluable insight into the considerations necessary for developing a content assessment module for semantic error diagnosis. However, the corpus itself is too small for extensive testing and evaluation of the machine learning components of content assessment module and in particular for training machine learning classifiers for content diagnosis. Future work to provide additional evidence in support of the CAM design will require a larger corpus.

The lack of available corpora points to a larger issue in the field of ICALL. The lack of common tasks and test sets hampers progress in the field. This lack is understandable, given the fact that ICALL systems are developed with different exercises, languages and users in mind. Because the applications are so varied, comparison across systems is difficult. However, the effectiveness of one approach, compared to another, cannot be evaluated scientifically without some common testing framework. This thesis begins to address this concern by providing the annotation scheme and seed corpus for evaluating system design.
This seed corpus is comprised exclusively of responses to short-answer reading comprehension questions. Loosely restricted reading comprehension questions fall in the viable processing ground of the language exercise spectrum (c.f. Chapter 3). This exercise spectrum describes language exercises with respect to the restrictions placed on learner answers in response to activities. For instance, consider the multiple-choice reading comprehension question in (88).

(88) When was Mozart born? a) 1756 b) 1796 c) 1812 d) 1917

This example is at one end of the spectrum extreme. Learner responses are highly restricted to one of four possible choices. Any other response is automatically wrong. Therefore evaluating responses for correctness of meaning is trivial. At the other end of the spectrum are activities such as in (89).

(89) How do the statistics in your country compare to those in the text?

Responses to questions such as in (89) are far less constrained and the evaluation of semantic correctness far more difficult. Essentially, there is no single right answer. While such questions may be evaluated for form errors, content assessment is beyond current technology.

The short-answer reading comprehension questions used to prompt the responses in the seed corpus represent activities that fall firmly between these two spectrum extremes. There is a right answer to such questions, but it may be expressed in a number of ways. These responses are ideal for developing and testing an approach to semantic error diagnosis because they exhibit linguistic variation on lexical, morphological, syntactic and semantic levels, but they have definable target responses that capture what it means to be correct.

In the CAM design, evaluating a learner response in the corpus involves comparing it to one or more target responses. This task is similar to tasks found in other
CL domains, including paraphrase recognition, natural language question answering and machine translation evaluation, among others. Chapter 4 elaborated on the parallels between fields and described the relevance of shallow processing techniques in those domains to ICALL content assessment.

Work from those fields inspired the CAM design, which performs a series of increasing complex linguistic comparisons across learner and target responses. This approach was effective in detecting errors in almost 90% of test cases. However, the analysis of the corpus and of the CAM results in Chapter 6 suggest that there are several issues to consider in evaluating the effectiveness of this shallow processing approach. These issues revolve around the question of which learner responses the approach should be able to evaluate. Crucial to answering this question are the following subquestions:

1. Are there reliable human judgments for a sufficient number of corpus pairs?
2. Are the target responses representative of correct answers?
3. For what kinds of questions and answers are the adopted strategies insufficient?

With respect to the first two subquestions, any implementation of the CAM design presented in Chapters 5 and 6 cannot reasonably be expected to consistently or reliably provide the correct diagnosis for pairs i) that humans cannot agree on the appropriate evaluation or ii) for which the learner is not targeting the given target response. Of the 566 pairs of responses in the corpus (both development and test sets), almost 20% fall into one of these two categories. The third question refers to examining both questions and answers to identify when responses may require more sophisticated analysis. The sections that follow take a closer look at these questions and tie them to avenues for future research.
7.1 Corpus Considerations

Voorhees and Tice (2000) suggest a certain futility in trying to identify a single set of target responses that everyone can agree on since even though rigorous training of graders may improve agreement, this does not guarantee the final set used to train and evaluate the system will conform to every instructor’s idea of right and wrong responses. Still, given that only 10% of the disagreements in the development set were due to grader error, rather than grader interpretation, and that the remaining disagreements have clear motivations, resolving the differences in grading to improve reliability seems possible. One obvious recommendation for developing a larger corpus of language learner data would be to provide explicit grading rubrics. While language instructors may or may not use clear, explicit grading rubrics for evaluating their students’ work, human graders judging responses for a learner corpus need detailed guidelines for use in grading and in training on how to use the guidelines in order to increase the likelihood they will agree in their judgments. Otherwise, graders – like many teachers – will differ in how they interpret questions and how harshly or leniently they grade responses.

The rubrics defined in Chapter 6 are a first step toward defining guidelines for semantic error diagnosis. This approach assumes meaning errors in learner responses are based on target modification – omissions, overinclusions, blends, etc. – and the error codes presented reflect this assumption. But the kinds of errors identified are not specific to the type of activity presented. Thus, the rubric would be applicable to grading responses to a wide range of activity types.

And of course, obtaining more data has a two-fold benefit. First, it increases the overall number of pairs humans agree on. And second, it would allow for a more extensive evaluation of the machine learning approach described in Chapter 6. Those
results showed that semantic error detection is effective and that machine learning of semantic error diagnosis is feasible. But the corpus must be expanded to have an instance base large enough for training and testing.

7.2 On Representative Target Responses

While one of the goals of this work was to explore the use of natural language processing strategies to compare two texts for diagnosing meaning errors, it is clear that in many cases, a single text (i.e., target response) is insufficient for effective processing. That is, for some learner responses, referred to in this work as alternate answers, there is no way to recognize the response as a correct answer because it is too dissimilar from the given target.

In such cases, expanding the set of target responses may be the only way to improve the chance for a correct diagnosis. Carr et al. (2002) mentions something similar with respect to their work with paraphrases. They point out that paraphrases pose an interesting challenge to automated scoring because every reasonable paraphrase cannot be pre-envisioned. They recommend pilot testing to build models of target responses based on typical learner responses, which will likely contain unanticipated paraphrases.

Caching is an important related technique for adding acceptable target responses for given responses. Rather than (or in addition to) pilot testing, correct learner responses may be stored as targets for later re-use. As the system diagnoses more and more responses to a question, it obtains more and more examples of target responses. Of course, this approach requires human oversight to ensure that only truly correct answers are added to the target response list.
In addition to positive evidence (i.e., additional target responses), this idea might be further expanded to include negative evidence (i.e., known incorrect responses). In the latter case, the set would no longer be a set of target responses. Rather, it would be a set of reference responses. Caching positive and negative responses to add to such a reference set would make system processing more efficient because analysis would not need to be duplicated for some learner responses. But an expanded reference set also potentially improves performance by including additional references that are different from the initial target response.

7.3 Improving the Comparisons

Along with expanding the corpus, there are several simultaneous fronts for future research in the area of improving the comparison between learner and target response texts. These include

- exploring the relationship between question complexity and detection,
- improving the alignment of response pairs, and
- experimenting with machine learning for diagnosis.

7.3.1 Question Complexity and Classification

As a test case for semantic error detection and diagnosis, loosely restricted reading comprehension (RC) questions illustrate many of the important issues in automatic meaning error diagnosis related to response variation and system processing. But because loosely restricted RC questions are similar in terms of the level of expected variation and explicitness of their activity models, classifying RC question by other features may prove useful for making better predictions about the ability of a content assessment module to analyze question responses for this type of activity. Several
taxonomies have been proposed in the literature and applied to RC questions, and categorization strategies include distinguishing by learning goals, knowledge sources, text types, question formats, and answer types.

But in order to make better predictions about how likely it is that a content assessment module will be able to reliably process a learner response to different RC questions, it is necessary to first understand what RC questions ask learners to do in their responses. To that end, a brief investigation of RC question classification is warranted.

7.3.1.1 Classifying by Learning Goals

Classification along the first dimension groups questions according to the learning goals – knowledge and cognitive skills – an instructor may have for students. One such classification is the Bloom taxonomy (Bloom, 1956), which identifies six categories of cognitive skills – knowledge, comprehension, application, analysis, synthesis, and evaluation. But note that Bloom’s taxonomy refers to general learning goals, not those specific to reading comprehension.

On the other hand, Smith and Barrett (1974) introduce a similar skill-based taxonomy specifically targeting reading comprehension. The Barrett taxonomy reduces the learning categories to recall, inference, evaluation, and appreciation. These four categories are subdivided into additional learning goals such as the ability to recall specific character traits or have an emotional response to the narrative plot. The Barrett taxonomy was specifically designed for teaching reading in the first language to children in grades 4-8.

Yet another skill-based taxonomy is from Nuttall (1982), who divides questions into literal comprehension, reorganization/reinterpretation, inference, evaluation and
personal response. These categories are similar to those in the Barrett taxonomy, but Nuttall’s approach is geared specifically toward second-language reading skills.

Finally, a more recent approach to classification of questions by learning goals is a revision of Bloom’s taxonomy (Anderson and Krathwohl, 2001). Anderson and Krathwohl’s revised taxonomy includes an expanded knowledge component. Under the revision, questions are classified by both knowledge type and cognitive skill. Knowledge types include factual (e.g., terms), conceptual (e.g., principles), procedural (e.g., algorithms), and metacognitive (e.g., strategies). This is in addition to the six cognitive processing categories: Remember, Understand, Apply, Analyze, Evaluate and Create. These categories correspond to activities such as recognizing terms, classifying information, implementing ideas, organizing data, critiquing texts and producing hypotheses, respectively. The taxonomy is given in Table 7.1.

<table>
<thead>
<tr>
<th>Knowledge Type</th>
<th>Cognitive Processes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Remember</td>
</tr>
<tr>
<td>Factual</td>
<td></td>
</tr>
<tr>
<td>Conceptual</td>
<td></td>
</tr>
<tr>
<td>Procedural</td>
<td></td>
</tr>
<tr>
<td>Metacognitive</td>
<td></td>
</tr>
</tbody>
</table>

Table 7.1: Anderson and Krathwohl (2001) Revision of Bloom’s Taxonomy

There are many similar question taxonomies. Inherent to each is the idea that classification should be based on the skills or knowledge the student must possess to be able to answer questions. Skills vary slightly by taxonomy, but taxonomies are highly overlapping. For instance, literal recall is a fundamental component of all, as is the ability to make inferences about the material present in texts.
From a processing perspective for ICALL content assessment, classification of questions by learning goals is useful because questions that target some skills may be more difficult to process than others. For instance, it seems reasonable to think that processing responses to recall questions would be relatively simpler for a content assessment module than processing responses to inference questions. And it is also reasonable to assume that questions with certain learning goals cannot be automatically processed for content at all. For example, the Create category in the Anderson and Krathwohl taxonomy might involve questions asking students to produce imaginative sentences that cannot be predicted or evaluated for correctness of meaning.

Ultimately, the question of whether the learning goals underlying RC question are linked to the difficulty of automatically evaluating answers to those questions is an empirical one. But any one of the available taxonomies mentioned above offers a useful way to group questions in order to test these assumptions.

7.3.1.2 Classifying by Knowledge Sources

A second dimension for question classification – the knowledge source dimension – groups questions not by what knowledge or skills a student must possess, but where that knowledge comes from. Irwin (1986) proposes a taxonomy that combines knowledge types with the source of information contained in an answer. Her taxonomy takes as a starting point the “question-answer relationship” (QAR) taxonomy of Pearson and Johnson (1978). The QAR taxonomy groups questions into one of three simple categories, based on whether the answer comes explicitly from the text, implicitly from the text or from the reader’s background knowledge. Irwin’s expanded QAR taxonomy (Ex-QAR) combines the implicit/explicit knowledge in answers with the type of information requested, such as metacognitive information.
As with learning goals, incorporating knowledge sources into a classification scheme in the present context would be appropriate in order to help identify questions that a content assessment module likely cannot effectively evaluate. Specifically, questions that have answers requiring background (or world) knowledge from the student are beyond the capabilities of ICALL content assessment. Such questions require information that is not available to systems on an unrestricted scale.

7.3.1.3 Classifying by Text Type

A third dimension for classification is the context of the question. Champeau de Lopez et al. (1997) propose a taxonomy that takes this dimension into account by classifying not only question stems, but also the source texts that the questions are written for.

In the Champeau de Lopez, et. al. taxonomy, the classification for stems is divided into categories that can be traced back to skill categories described for other taxonomies in Section 7.3.1.1. It is the classification by text type that makes their taxonomy of interest. Underlying texts are classified by subject (e.g., psychology, architecture, chemical engineering), rhetorical pattern (e.g., static description, comparison-contrast, cause-effect, argumentation), source (e.g., newspaper, textbook), and form (e.g., summary, letter, table, sentence list).

While each of the four classification components provide relevant information about text type, the specific distinctions proposed in Champeau de Lopez et al. (1997) are perhaps not appropriate for more general use. The taxonomy was designed specifically for multiple-choice questions on scientific and technology reading. Thus, their subject classification, for instance, is not appropriate for a wide range of reading materials. Also, the source texts for the multiple-choice questions are typically very
short passages. As a result, the source texts likely tend to have a single, clear rhetorical structure. It may not be possible to classify longer texts into a single rhetorical structure category. However, different sections of a longer text may exhibit a particular rhetorical structure. Thus, it may be possible to classify a question by the rhetorical structure of the section of the source text that contains the answer to that particular question. Such a classification would need to be explicitly included in the activity model of an exercise since it cannot currently be detected automatically.

In addition, Meyer (1985) proposed a simpler rhetorical structure classification for texts with only five categories – collection, causation, problem-solution, description and comparison – but the issue of classifying a text with a single rhetorical structure remains.

Nevertheless, classifying questions by the rhetorical structure of underlying source text passages may facilitate question processing. For example, the linguistic choices that identify a text as having a “comparison-contrast” rhetorical structure will be different from those that identify a text as a “process” text. This is not to say that processing questions based on texts of one rhetorical structure is inherently more difficult than processing questions based on texts of another structure. But the linguistic choices realized in a text will often influence the structure of responses to questions about that text. In processing, knowing the rhetorical structure of a text may prove useful in identifying the structure(s) of expected responses.

7.3.1.4 Classifying by Question Format

Yet another dimension to consider for question classification is the format of the question. Nuttall (1982) proposes a format-based taxonomy that divides questions into four types:
1. Yes/No questions (e.g., Is it X?)
2. Alternative questions (e.g., Is it X or Y?)
3. Wh-questions (e.g., Who is X?)
4. How/Why Questions (e.g., Why is X?)

The format of a question can impact the difficulty of processing because the format (i) guides the learner to answer in a particular way and (ii) restricts the range of suitable target answers. Yes/No questions are easier to process than Wh-questions, for instance. But the Nuttall (1982) format taxonomy is insufficient for distinguishing questions in the current study since most of the questions have a wh-question format.

Moreover, the effectiveness of using a format-based classification to distinguish questions for purposes of processing is debatable. Format-based classification of questions is used in another CL domain – natural language question answering for information retrieval (QA-IR), where the focus is on using question format to guide the search for answers to fact-based questions. QA-IR research has shown that some fact-based questions (e.g., how and why) are harder to answer automatically than others, but that the format of questions is ultimately insufficient for guiding processing (Hirschman and Gaizauskas, 2001). For example, just knowing that a wh-question starts with what does little to indicate the difficulty of answering the question as the examples in (90) suggest.

(90) 1. What year did the Civil War begin?
2. What are three states that joined the Confederacy?
3. What caused the Civil War?

Thus, pursuing a classification of questions by question format may have limited usefulness in distinguishing questions that can be effectively processed in a content assessment module.
7.3.1.5 Classifying by Answer Type

While QA-IR suggests that classifying by question type has dubious merit (at least in terms of using that classification to identify processing difficulties), the QA-IR literature offers an alternative classification, that of answer typing, relevant to classifying RC questions in an ICALL domain. Recall from Chapter 4 that answer typing refers to labeling the category of answer a question is “looking for.” For example, consider the hypothetical target and learner responses in (91).

(91) a. **Target Response**: The man is dancing with his wife.

   b. **Learner Response 1**: The man is dancing with a lady.

   c. **Learner Response 2**: The man is dancing with a lake.

The underlined portion of (91a) is the concept in variation. If the task is a picture description activity in which the relationship of the man and woman dancing is unclear, then (91b) is a reasonable answer because lady is of the same type, PERSON, as wife. (91c), on the other hand, is not of type PERSON.

For question classification, it might be expected that factoid answer types (e.g., PERSON, LOCATION, etc.) are likely easier to process than DEFINITION questions, which in turn are possibly easier to assess than REASON questions. Future testing would be required to determine whether classifying RC question by answer types will shed any light on which questions are easier to process reliably for content assessment.

7.3.1.6 A Final Note on RC Question Taxonomies

One last point to make with respect to taxonomies for classifying RC questions relates to the background of anticipated learners. Specifically, does the taxonomy take
into consideration whether the student is learning to read in a first or second language? Presumably, if a learner already knows how to read in one language, they have mastered many of the reading skills established in a taxonomy such as Barrett’s taxonomy, which targets skills for children learning to read for the first time.

Second language acquisition (SLA) does not necessarily follow the same pattern as first language acquisition, and taxonomies for RC questions in an SLA domain should ideally take this into consideration. The Champeau de Lopez et al. (1997) taxonomy mentioned above is one SLA-specific approach to question classification. But while it targets reading in a second language, it is very tailored for use in a particular situation. Another SLA-specific approach (not mentioned above) is a taxonomy suggested by Day and Park (2005). They present a classification system that includes six types of comprehension (literal, reorganization, inference, prediction, evaluation and personal response) and five question formats (yes/no, alternative, true/false, wh-/how, and multiple choice). It draws heavily from Nuttall’s approach, which is a third taxonomy designed for SLA reading.

At the same time, an approach such as Anderson and Krathwohl’s is still very useful because it classifies learning goals and knowledge types that are independent of subject or level of the student. It is not specific to language learning but applies to all learning. Thus, it is general enough to be applicable in a second language-learning (and reading) context. Furthermore, it is not specific to reading comprehension, so it can be used for a wider array of activity types.

7.3.1.7 RC Question Taxonomy

As the discussion of question classification indicates, none of the outlined taxonomies directly account for question processing difficulty. This is not surprising since none of
the taxonomies was designed specifically with ICALL in mind. Yet such knowledge would be useful either for pre-screening the types of questions that are added to an ICALL activity set or for associating a confidence level to the diagnosis produced by the ICALL system that includes such activities.

The preceding discussion does suggest that other dimensions of question classification may relate indirectly to processing difficulty. In order to test this connection, future research should look at the various dimensions summarized in (92).

(92)  
• **Learning Goals.** The targeted cognitive skills and knowledge.

• **Knowledge Sources.** The source of the targeted answer to a question.

• **Text Type.** The nature of the text a question is based on.

• **Answer Type.** The kind of answer a question is looking for.

This classification system combines the knowledge and cognitive skills dimensions defined in Anderson and Krathwohl (2001) with the knowledge sources suggested by Pearson and Johnson (1978), and the rhetorical structure text classification of Champeau de Lopez et al. (1997). A classification taxonomy based on these approaches is presented in Table 7.2, abbreviated only for space considerations.

As suggested in Section 7.3.1.6, Anderson and Krathwohl’s general learning goals can be suitably applied to a SLA reading context, and their categories are sufficiently detailed to provide relevant subclassification of skills. For instance, the *Understand* skill, though abbreviated in the table, is subdivided into seven categories – *Interpret, Exemplify, Classify, Summarize, Infer, Compare and Explain.*

The six cognitive skill categories from Anderson and Krathwohl are combined with their four knowledge categories, each with 2-3 subclassifications. These subclassification distinguish between knowledge types such as knowledge of terminology (factual) or strategic knowledge (metacognitive) or knowledge of skills and algorithms.
| Source             | Terminology | Specific Details | Classification | Generalizations | Theories | Skills | Methods | Selection | Strategic | Cognitive Tasks | Self-Knowledge | Explicit in text | Implicit in text | World Know. | Definition | Description | Classification | Comparison | Chronology | Process | Cause-Effect | Hypothesis | Argumentation | Exemplification | Other Structure | Person | Organization | Location | Reason | … | Other |
|--------------------|-------------|------------------|-----------------|-----------------|------------|--------|---------|----------|-----------|----------------|----------------|-----------------|----------------|-------------|------------|------------|-------------|-------------|-----------|----------|----------|----------------|------------|--------------|----------------|---------------|--------|-------------|----------|--------|----|-------|
| Text Type          |             |                  |                 |                 |            |        |         |          |           |                |                |                 |              |            |            |            |            |           |          |          |                |            |              |                |               |        |             |          |        |   |      |
| Answer Type        |             |                  |                 |                 |            |        |         |          |           |                |                |                 |              |            |            |            |            |           |          |          |                |            |              |                |               |        |             |          |        |   |      |

Table 7.2: Dimensions of RC Question Classification
(procedural). In addition, the classification includes the three categories related to the source of the answer. Following Pearson and Johnson (1978), answers may be explicit in the text or implicit in the text, or answers may come from world knowledge. Third, the rhetorical structures identified in Champeau de Lopez et al. (1997) are adopted in the taxonomy with eleven subcategories.

And, finally, an initial set of answer types selected for the RC classification might be compiled from those used in various QA-IR. Bailey (2001) discusses various type hierarchies used in QA-IR. However, most QA-IR answer type hierarchies focus on answer types for fact-based questions. Thus, any hierarchy borrowed from the QA-IR domain would need to be supplemented with more types to be generally applicable in an ICALL setting.

7.3.1.8 Question Classification and Performance

Analyzing system performance by question type is not revolutionary. For instance, Marín (2004) notes that in her work on automatic scoring of short answers with the Willow/Atenea system, the system performed best on description or definition questions. She argued that some questions, such as those requiring numerical analysis or a comparison, were outside the limits of the technology employed for the system. While semantic error diagnosis for ICALL is a different task than the one intended for the Willow system, a similar issue of question classification and performance is an interesting line of future research. The taxonomy presented in Table 7.2 brings together several dimensions of question complexity, and a combination of these factors may prove to be best correlated with system performance.
7.3.2 Improvement of Alignment Between Responses

Beyond future research into question classification and system performance, an additional line of continuing research should consider improvements to the processing approach itself. While this research establishes a clear research agenda for semantic error diagnosis, it reflects initial experiments in using surface-level linguistic cues for semantic error diagnosis for ICALL. Thus, there are a number of avenues to explore in improving the alignment between target and learner responses. These include

- improving the quality of target responses,
- recognizing answer types in the learner response, and
- improving the alignments between responses.

7.3.2.1 Improving Target Responses

The target responses used in this thesis were created by experts familiar with second language teaching, but unfamiliar with natural language processing. Thus, the target responses reflect the types of answers that a typical instructor might create. But the analysis presented here suggests that target responses need to be carefully crafted for successful language processing. Consider the pair in (93).

(93) a. **Question Cue**: What is the definition of socialization?

   b. **Target Response 1**: The definition of socialization is the process of learning the mannerisms, expectations and behavior of a society.

   c. **Target Response 2**: Socialization is learning how to behave in society.

One key point to make is that to extend ICALL systems to process activities in the viable processing ground, such as the loosely restricted reading comprehension activities explored here, target responses need to be concise, containing all but only
the necessary concepts and relations. Given the source material, both targets in (93) are acceptable responses, but (93b) conveys the minimal necessary information for an acceptable answer.

As seen in Chapter 5, strategies for removing given material can be effective for removing some concepts in target responses that are unnecessary in learner responses. For example, in (93a), the definition of socialization would be removed from the list of concepts to align.

Once given information is removed, the remaining keywords in (93a) are process, learning, mannerisms, expectations, behavior and society. Are all these concepts equally important? Based on the acceptable alternate target response in (93b), the answer is no. Not all the terms are equally important. Only learning, behavior and society are critical for a response.

However, because the computer cannot tell the difference between critical and unnecessary terms in target responses, the distinction needs to be made in the activity model. Manual (or possibly automatic) term weighting is one way to guide the system on how to treat individual concepts. But this adds an extra layer of effort required to set up an activity in an ICALL system. Expanding the target response sets as discussed in Section 7.2 can mitigate the problem as both (93a) and (93b) might be included in the response set.

7.3.2.2 Recognizing Answer Types

Chapter 4 provided a brief overview of answer types and Section 7.3.1.5 suggested how they might be used for exploring the relationship between question classification and system performance. They also have a potentially useful function in processing. Consider the pairs in (94) and (95), an omission and an overinclusion error, respectively.
(94) a. **Question Cue:** Where was Bob Hope when he heard about the news?

    b. **Target Response:** He was eating breakfast at his home.

    c. **Learner Response:** Bob Hope was at that moment happily eating his breakfast when he heard about the news.

(95) a. **Question Cue:** Which form of programming on TV shows the highest level of violence?

    b. **Target Response:** Cartoons show the highest level of violence.

    c. **Learner Response:** Television drama, children’s programs and cartoons.

While (94c) has high overlap with the target in (94b), the crucial concept – the location of Bob Hope – was missing. In (95c), the crucial concept is present, but so are other extra concepts that are incorrect. In both cases, answer types maybe be of use in distinguishing between correct and incorrect answers. In the first case, the answer type is LOCATION; in the second, it might be COMMUNICATION or CREATION. In processing of the pair in (94), if no instance of the correct answer type is found, then it would correctly be labeled as an omission error. In (95), too many concepts of the correct answer type would be found. In such cases, the system could label it as overinclusion.

Using some form of answer typing in ICALL-related research is not unprecedented. Gerbault (1999) describes five expected answer types, based on the type of questions presented to the learner. Her classification is as follows:

1. **Type 1:** Short answers to situational questions about a text. These are equivalent to the factoid kinds of questions in the QA task (*who, what, when, where*).

2. **Type 2:** Statement extracted from a text and modified to be syntactically or pragmatically appropriate.
3. **Type 3:** Reformulations of material from the text with additional context added.

4. **Type 4:** Synthesis of material from the text.

5. **Type 5:** Reformulations of reasoning processes in the text. Such answers involve inferences and making logical connections between parts of the text.

There are two points to make about this set of answer types. First, the answer types do not make reference to the kind of content required in an appropriate answer. Rather, the focus is in the kind of mental processing a learner must do to be able to produce an acceptable answer. However, there is a connection from these types to the level of form or content variation that might be expected in a answer (very little in Type 1, but possibly quite a lot in Type 5).

A second, and more problematic, point is that from the brief description in Gerbault (1999), it is unclear exactly how these answer types can lead to improved content assessment. While Gerbault outlines the answer types and data collection for gathering information about each answer type, there is no follow-up to address the details of how her answer typing can be put to practical use. A set of content-focused answer types (as in the QA systems) seems more directly useable in content assessment.

However, use of answer types in ICALL can have some of the same limitations as in QA systems. Just because an answer is of the correct type does not mean it is appropriate.

(96) The man is dancing with his brother.

For example, if (96) is a learner response to the example in (91), then the system would not detect the error because *brother* is of type PERSON. Thus, answer types are only a partial solution. They must be combined with accurate alignment.
7.3.2.3 Improving Alignments

While the existing alignment modules make useful comparisons between learner and target responses, analysis of the test set pairs showed there is clear room for improvement. For instance, synonym alignment suggested useful alignments in the corpus pairs such as almost ⇔ about, grown ⇔ raised, and promote ⇔ campaign. However, many of the synonym alignments were tenuous at best including

- think ⇔ see,
- take ⇔ give,
- try ⇔ show, and
- make ⇔ campaign.

Spelling alignment also leaves room for improvement. For instance, while most of the suggested spelling corrections were actually correct, one of the more interesting alignment failures was American ⇔ Mexican. The pair has a case-insensitive edit distance of 2 (substitute r for x and insert an a) and are not on common noun word lists. Clearly, the current word lists could be expanded to include proper nouns to handle such cases, but other alternatives to spell checking might also be explored.

And finally, similarity scoring is problematic because similarity alignments may link terms that are less semantically related than the other types of matches. This can lead to a lot of spurious alignments along with useful ones. Modifying the scoring algorithm for similarity scoring or exploring alternate approaches to measuring similarity are areas of future research.

In addition, future research could explore expanding the range of alignment types to include aligning idiomatic expressions, multi-word expressions, equivalent discourse cues, and negation terms, among others.
7.3.3 Developing Machine Learning for Diagnosis

The review of work on content assessment in Chapter 2 mentioned the work of Pulman and Sukkarieh (2005) on using regular expressions to grade short-answer questions. Their experiments with machine learning were an effort to avoid hand-crafting answer patterns by learning the patterns automatically from marked up learner responses. But after comparing several machine learning algorithms, they concluded that the accuracy of a machine learned approach was not as accurate as they needed for automatic grading.

In contrast, the current work suggests that machine learning for error diagnosis in ICALL is a fruitful avenue of research. If the intended use of the ICALL system is more formative than summative, providing low-stakes language practice to the learner, such an ICALL system can rely on imperfect technology.

This is not to say that performance of the diagnosis classifier cannot be improved. Much to the contrary. Given enough training data, machine learning might be explored for both alignment and detection/diagnosis. Detection and diagnosis has already been discussed in preceding chapters, but for alignment, the machine learner would learn how to rank candidates for aligning the most likely equivalent concepts. However, in a sample of 50 development pairs, there were 1,218 total candidates suggested, but only 193 of which were unique. This means that candidates usually have no competition. In the sample, only 17 (8.8%) had more than one candidate. Examples of tokens in this position include our, or, person, the, they, from, percent and larger. And no token position had more than 2 candidate alignments suggested for it. Thus, if most positions have at most 1 candidate (and no more than 2), then there may be little need for sophisticated machine learning to learn the rankings.

\[2\]A single position might have been suggested more than once, by different alignment modules.
For machine-learned error detection and diagnosis, and indeed for alignment, the biggest bottleneck is training data. But while there is not enough learner corpus data for training and testing, there are other sources of data that might prove useful. For example the Microsoft Research Paraphrase (MSRP) Corpus (Dolan et al., 2004). This corpus consists of roughly 5,800 pairs of sentences extracted from newspaper data and labeled with Yes/No judgments as to whether the texts are paraphrases of each other. For purposes of diagnosis, this corpus would need to be relabeled with diagnosis codes. And to do this, one half of each pair would need to be selected as the target response, and the other half the learner response. This approach has several advantages including that the corpus already exists and that the pairs have been labeled with “detection” values. However, one important drawback of using this data is that it is not language learner data. It is edited newspaper text, so it has few (if any) grammatical errors or mistakes of the kind language learners would make. Thus, training on it may not help the system learn to diagnose errors in learner text.

Another possibility for approaching the problem of acquiring more learner corpus data is discussed in Brockett and Dolan (2005) with respect to the development of a paraphrase recognition system used to construct an aligned monolingual corpus. Again, their approach uses sentences extracted from online newspaper resources. These sentences were often on the same event, just taken from two different sources. Thus, they could be sure that they were collecting a set of pairs that contained some paraphrases. For collecting a set of learner and target responses, the question of where the learner and target responses come from is not clear. However, sentences might be extracted/aligned from learner/model essays to create pairs of short texts to add to the corpus. Because each pair includes actual learner language, it would potentially be more relevant for training than text pairs from newspaper data, as in the MSRP corpus.
7.4 Summary

Current pedagogical theories have argued that meaningful interaction in the second language is critical for language learning. Based on the spectrum defined in Chapter 3, this thesis argues that while some meaningful activities are too unrestricted for ICALL systems to provide effective content assessment, where the line should be drawn on the spectrum is an open question. In her work on second-language acquisition and CALL, Chapelle (1998) argues that learners not only must have the opportunity to practice the second language (L2) but also need interaction that requires modification and negotiation of meaning. Thus, one important concern for ICALL system designers is how to create interesting, meaningful activities within the limits of the technology.

There is then a delicate balance between the kinds of activities instructors may want and what they can get. For instance, recall the example from Chapter 1 from Cox’s study of open-ended tasks (Cox, 2005), repeated in (97).

(97) Discuss what you think life will be like in fifty years. List three aspects you agree on.

Open-ended questions such as this may be very meaningful and require the kind of negotiation Chapelle argues for, but they cannot be evaluated for the correctness of content in the same way that the corpus question in (98) can be evaluated.

(98) What are the three factors that limit the influence of media on public opinion?

The less predictable the set of correct answers for any given question, the less likely automatic analysis will produce adequate diagnosis. The middle ground of the spectrum, in which target responses have some predictability, is where the current work has focused in exploring where the limit of effective content assessment resides.
Because variation is possible across learner responses in activities in the middle ground of the spectrum, any degree of content assessment must be flexible and support the comparison of target and learner responses on several levels including token, chunk and relation levels. The architecture for a content assessment module (CAM) presented in Chapter 5 provides this flexibility using multiple surface-based matching strategies and existing NLP tools, several of which have been used in other CL applications.

The CAM design focuses on general-purpose techniques for effective semantic error diagnosis beyond the limited-domain functioning of content assessment in existing ICALL systems or the response scoring provided through automatic grading systems such as C-rater (Leacock, 2004). Thus, by developing this model, as well as exploring the middle ground of activities, this work begins to bridge the gap between what is practical and feasible from a processing perspective and what is desirable from the perspective of current theories of language instruction.

The seed corpus described in Chapter 6 represents the first step in a larger research agenda to provide an effective evaluation framework for ICALL technologies. The corpus provides detection and diagnosis judgments for pairs of learner and target responses. This work has argued for an evaluation that examines i) whether the system accurately determines if a learner response is correct and ii) how reliable the system is in diagnosing particular semantic errors. This type of systematic evaluation of the strengths and weaknesses of an ICALL system component is missing from much of the existing ICALL literature. But such evaluation is essential for making progress in the field.

And progress is needed. Content assessment is essential for better integration of ICALL systems. But existing ICALL systems emphasize form assessment, limiting their usefulness in real-life language teaching. To push the boundary of what ICALL
systems can offer language instructors, ICALL systems must be able to process learner responses from less-restricted activities. And this means ICALL systems need effective content assessment.

This work addresses that need by providing a testable model of content assessment designed with a range of language-learning activities in mind. To explore properties and processing requirements of activities, the CAM builds on the inspiration from other CL subfields to allow for content assessment of a wide range of concepts and relations between concepts. This is a new application for NLP technology originally designed for other CL domains and tasks, but adapted and modified for the current ICALL setting.

This work also contributes to the understanding of which language-activities can and should be targeted for ICALL. Explicitly defining the viable processing ground is a critical step in determining the feasibility of incorporating a wider range of activities into effective ICALL systems. Systematically exploring activities and content processing of those activities will make it possible to identify the different levels of content assessment possible within the boundaries of current technology, the types that push those boundaries in interesting directions, and those that lie beyond the capabilities of current ICALL systems. Such exploration will result in better understanding of what NLP technology offers to language teaching practice and the extent to which it can be integrated into real-life practice.
APPENDIX A

GLOSSARY OF TERMS

The definitions below are for terms used throughout the thesis.

1. **Antonymy** – A semantic relationship such that one word has the opposite meaning as another word (e.g., hot and cold are antonyms).

2. **CALL** – Computer-Aided Language Learning. Approaches to language learning and instruction that incorporate the use of technology.

3. **Closed-class words** – The category of words that is “closed” to the addition of new words. That is, there are relatively few closed-class words and it is relatively difficult to introduce a new word to the class. These words include determiners, conjunctions, pronouns, and prepositions.

4. **Content** – In the language-teaching literature, content is what is presented to the learner in language-learning materials. But for the purposes of this thesis, content is the meaning expressed by the learner response which the system attempts to analyze.

5. **Content analysis** – Analysis of semantic or pragmatic information in order to assess the meaning of a learner response.

6. **Content assessment** – Evaluation that focuses on the meaning, rather than the grammatical form, expressed by a learner response. This assessment involves evaluating the compatibility of the ostensible content of a learner response and the expected content of a target response.
7. **Content word** – Non-function words that convey meaning, such as nouns, adverbs, adjectives and verbs.

8. **Correct response** – A learner response that conveys appropriate meaning given the activity. Note that given this definition, a correct response may contain grammatical errors.

9. **Form analysis** – Analysis of grammatical form in order to assess either the form or content of a learner response.

10. **Form assessment** – Evaluation that focuses on grammatical form and syntactic structure of learner responses.

11. **Function word** – Words that serve a largely grammatical function, and convey relatively little meaning. These include

    - possessive pronouns (e.g., *my, your*),
    - auxiliary verbs (e.g., *be, has*),
    - modal verbs (e.g., *might, should*),
    - determiners (e.g., *the, another, whichever*),
    - coordinating conjunctions (e.g., *and, or*),
    - subordinating conjunctions (e.g., *while, because*), and
    - prepositions (e.g., *in, at, by*).

12. **Hyponymy/hypernymy** – A semantic relationship between two words such that one word is a kind of the other. For instance, *tulip* is a kind of *flower*. Thus, a *tulip* is a hypernym of *flower*; *flower* is a hypernym of *tulip*.

13. **Ill-formed response** – Responses that contain one or more errors in grammatical form.

14. **Incorrect response** – A learner response that fails to convey the appropriate meaning given the activity. In other words, such responses contain one or meaning errors, though the response may be perfectly grammatical.
15. **Input** – In the language-teaching literature, input often refers to stimulus that the learner receives. For the purposes of this thesis, input refers to the input to the ICALL system. As such, the input is the learner response.

16. **ICALL** – Intelligent Computer-Aided Language Learning. A subset of CALL approaches that incorporates natural language processing (NLP) technology to provide language analysis.

17. **Learner response** – An answer to a language-learning exercise provided by the language learner.

18. **Lemma** – The canonical form of a set of forms that are all variants of the same word (e.g., *dog* is the lemma of *dogs*, *dog*, *dogged*, etc.). For English, *lemma*, *stem* and *stemmed word* may be used interchangeably.

19. **Lemmatizer** – A computer program that automatically identifies the lemma for each input word.

20. **Machine learning** – A field of study that focuses on algorithms for training computers to learn from data.

21. **Machine translation** – A subfield of computational linguistics that deals with problems of automatically translating input text from one language to another.

22. **Meronymy** – A semantic relationship between two words such that the concept represented by one word is a part of the concept represented by the other word. For example, *knee* is a part of (i.e., a meronym of) *leg*; *leg* is a holonym of *knee*.

23. **Natural language question answering** – A subfield of information retrieval that, instead of returning documents based on a keyword query, returns a text extract that contains an answer to a natural language query.

24. **NLP** – Natural Language Processing. A subfield of computational linguistics that studies problems related to both automatic understanding and automatic generation of human languages by a computer system.
25. **(Domain) ontology** – A collection (e.g., a database) of the entities and relationships between entities in a particular domain or field. For instance, an ontology for the domain of botany would include the concepts of *plant, tree, flower, stem, root, stamen*, etc. and how these entities are related (e.g., hierarchically, part-of).

26. **Open-class words** – The class of words that is “open” to new additions such as nouns, verbs, adjectives and adverbs.

27. **Open-ended task** – A language-learning activity for which there is no predefined target response. That is, the instructor does not know in advance what the answer to the activity is.

28. **Output** - In the language-teaching literature, output refers to the learner response. In contrast, for this thesis output is the analysis the system produces.

29. **Part-of-Speech (POS) tagger** – A computer program that associates part of speech class labels (or tags) with words. For example, as a common noun, *commencement* might be labeled with the tag *NNS*.

30. **Stemmed word/stem** – The part of a word common to all inflected variants of the word (e.g., *dog* is the stem of *dogs, dog, dogged*, etc.).

31. **Stop word** – While no definitive list of stop words exists, a stop word is any word removed before processing. Typically, these include frequently used function words (e.g., *the, of*) that carry little semantic content.

32. **Summarization** – A subfield of computational linguistics that deals with problems related to automatically extracting the main ideas of a text and presenting that material in a summary.

33. **Synonymy** – A semantic relationship such that one word has the same meaning (or nearly so) as another word (e.g., *quick* and *fast* are synonyms).

34. **Synset** – In the WordNet database, a synset is a set of words that may all be used interchangeably in some context.
35. **Target response** – An answer provided in the activity model of an ICALL system as an expected response to an exercise. The target response is typically provided by the instructor or domain expert.

36. **Tokenizer** – A computer program that automatically identifies individual tokens from a string of input characters.

37. **Tryponymy** – A semantic relationship between two words such that one word is a way of doing the action represented by another word. For instance, *whisper* is a way to *speak*. Thus, *whisper* is a tryponym of *speak*. Tryponymy is parallel to hyponymy, but is applied to verbs.

38. **Well-formed response** – Responses that are grammatically correct, regardless of whether they are semantically appropriate for a given exercise.
APPENDIX B

THE BLEU METRIC

Chapters 2 and 4 refer to work with the BLEU metric for evaluating machine translations. The following is a more detail explanation of how BLEU works. To start, the BLEU scoring formula is given in (99), where $c$ is the length of the candidate translation, $r$ is the reference corpus length, $N$ is the length of n-grams, and $w_n$ is a weighting factor.

\[
(99) \quad \log(\text{BLEU}) = \min(1 - \frac{r}{c}, 0) + \sum_{n=1}^{N} w_n \log(p_n)
\]

This formula basically counts the weighted overlap of n-grams in the machine and human translations and modifies that sum by a brevity penalty that penalizes machine translations that differ significantly in length from the reference translations. Thus it relies on the idea that the closer the machine output is to a string match (calculated in terms of the number of substring matches), the better the translation.

The most important fact about the formula in (99) is that BLEU relies on the modified precision $p_n$ as the basis for evaluating translations. Standard unigram precision is calculated as in (100).

\[
(100) \quad \text{Precision} = \frac{\text{Count(word overlap in candidate and reference translations)}}{\text{Count(words in candidate translation)}}
\]

In contrast, the modified n-gram precision of a candidate machine translation is calculated as in (101).

\[
(101) \quad p_n = \frac{\text{Count(clipped n-gram overlap in candidate and reference translations)}}{\text{Count(n-grams in candidate translation)}}
\]

293
The reference translation is not limited to one translation. This accounts for the fact that there may be multiple acceptable reference translations available. In the event of multiple reference translations, the overlap is counted as n-gram matches with n-grams in *any* of the reference translations. But to avoid allowing multiple candidate n-grams to match the same reference n-gram, the total number of matches is limited to the maximum number of instances that n-gram appears in any single reference translation. For instance, if the bigram *of the* appears 4 times in the candidate, 3 times in one reference, and 2 times in a second reference, then the total overlap count for *of the* is 3. The overlap count for *of the* has been effectively “clipped.”

Modified n-gram precision is actually calculated for an entire test corpus in the BLEU system, rather than for individual sentences, but the n-gram counts are sentence-by-sentence. In other words, the final precision score ($p_n$) for a test corpus is the sum of the total clipped n-gram overlap counts for each sentence divided by the sum of n-grams in all sentences.

Second, in theory, n-grams may be any length $n$, but Papineni et al. (2001) found that n-grams of length 4 or less worked best. Furthermore, in comparing precision scores for n-grams between lengths 1 and 4, they found that although as $n$ increases, precision decreases exponentially, any of the n-gram lengths can distinguish between good and bad translations. To combine the n-gram precisions into an average, while still taking into account the drop in precision as $n$ increases, Papineni, et al. multiply the sum of precision scores by the positive weight, $w_n = \frac{1}{N}$, where $N$ is the n-gram length, and take the geometric mean of the precision scores.\(^1\)

And finally, using precision penalizes candidate translations that are longer than their reference translations because the overlap is lower. But candidate translations that are shorter than reference translations are not penalized by the measure.

---

\(^1\) Geometric mean is equivalent to $\sqrt[N]{\prod_n p_n}$. 294
To account for this, a brevity penalty, $\min(1 - \frac{c}{r}, 0)$, is imposed on the overall precision score. For candidate sentences with more than one reference translation, the reference translation that is the best overlap match to the candidate is selected as the representative length to add to the reference length sum. The brevity penalty only penalizes the system for candidates shorter than their reference translations.
APPENDIX C

NLP TAGS FOR AND LABELS

The following tables outline the tagsets used by the NLP tools in CAM for part-of-speech tagging, semantic class labeling, and dependency relations.

<table>
<thead>
<tr>
<th>TreeTagger Tag</th>
<th>Penn Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>PP</td>
<td>PRP</td>
</tr>
<tr>
<td>PP$</td>
<td>PRP$</td>
</tr>
<tr>
<td>SENT</td>
<td>.</td>
</tr>
<tr>
<td>VH</td>
<td>VB</td>
</tr>
<tr>
<td>VHD</td>
<td>VBD</td>
</tr>
<tr>
<td>VHG</td>
<td>VBG</td>
</tr>
<tr>
<td>VHN</td>
<td>VBN</td>
</tr>
<tr>
<td>VHP</td>
<td>VBP</td>
</tr>
<tr>
<td>VHZ</td>
<td>VBZ</td>
</tr>
<tr>
<td>VV</td>
<td>VB</td>
</tr>
<tr>
<td>VVD</td>
<td>VBD</td>
</tr>
<tr>
<td>VVG</td>
<td>VBG</td>
</tr>
<tr>
<td>VVN</td>
<td>VBN</td>
</tr>
<tr>
<td>VVP</td>
<td>VBP</td>
</tr>
<tr>
<td>VVZ</td>
<td>VBZ</td>
</tr>
<tr>
<td>NP</td>
<td>NNP</td>
</tr>
<tr>
<td>NPS</td>
<td>NNPS</td>
</tr>
</tbody>
</table>

Table C.1: Tagset Mapping
<table>
<thead>
<tr>
<th>Tag</th>
<th>Description</th>
<th>Tag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>Dollar</td>
<td>NNS</td>
<td>noun, common, plural</td>
</tr>
<tr>
<td>&quot;</td>
<td>opening quotation mark</td>
<td>PDT</td>
<td>pre-determiner</td>
</tr>
<tr>
<td>’</td>
<td>closing quotation mark</td>
<td>POS</td>
<td>genitive marker</td>
</tr>
<tr>
<td>(</td>
<td>opening parenthesis</td>
<td>PRP</td>
<td>pronoun, personal</td>
</tr>
<tr>
<td>)</td>
<td>closing parenthesis</td>
<td>PRPS</td>
<td>pronoun, possessive</td>
</tr>
<tr>
<td>,</td>
<td>comma</td>
<td>RB</td>
<td>Adverb</td>
</tr>
<tr>
<td>–</td>
<td>Dash</td>
<td>RBR</td>
<td>adverb, comparative</td>
</tr>
<tr>
<td>:</td>
<td>sentence terminator</td>
<td>RBS</td>
<td>adverb, superlative</td>
</tr>
<tr>
<td>:</td>
<td>colon or ellipsis</td>
<td>RP</td>
<td>Particle</td>
</tr>
<tr>
<td>CC</td>
<td>conjunction, coordinating</td>
<td>SYM</td>
<td>Symbol</td>
</tr>
<tr>
<td>CD</td>
<td>numeral, cardinal</td>
<td>TO</td>
<td>“to”</td>
</tr>
<tr>
<td>DT</td>
<td>Determiner</td>
<td>UH</td>
<td>Interjection</td>
</tr>
<tr>
<td>EX</td>
<td>existential there</td>
<td>VB</td>
<td>base form verb</td>
</tr>
<tr>
<td>FW</td>
<td>foreign word</td>
<td>VBD</td>
<td>past tense verb</td>
</tr>
<tr>
<td>IN</td>
<td>preposition/subordinating conjunction</td>
<td>VBG</td>
<td>present participle verb, gerund</td>
</tr>
<tr>
<td>JJ</td>
<td>adjective or numeral, ordinal</td>
<td>VBN</td>
<td>past participle verb</td>
</tr>
<tr>
<td>JJR</td>
<td>adjective, comparative</td>
<td>VBP</td>
<td>present tense verb</td>
</tr>
<tr>
<td>JJS</td>
<td>adjective, superlative</td>
<td>VBZ</td>
<td>present tense verb, 3rd person, sg</td>
</tr>
<tr>
<td>LS</td>
<td>list item marker</td>
<td>WDT</td>
<td>WH-determiner</td>
</tr>
<tr>
<td>MD</td>
<td>modal auxiliary</td>
<td>WP</td>
<td>WH-pronoun</td>
</tr>
<tr>
<td>NN</td>
<td>noun, common, singular or mass</td>
<td>WPS</td>
<td>WH-pronoun, possessive</td>
</tr>
<tr>
<td>NNP</td>
<td>noun, proper, singular</td>
<td>WRB</td>
<td>Wh-adverb</td>
</tr>
<tr>
<td>NNPS</td>
<td>noun, proper, plural</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table C.2: Penn Tagset
<table>
<thead>
<tr>
<th>ABILITY</th>
<th>GOAL</th>
<th>QUALITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACHIEVEMENT</td>
<td>GOVERNMENT</td>
<td>QUANTITY</td>
</tr>
<tr>
<td>ACT</td>
<td>GROUP</td>
<td>REASON</td>
</tr>
<tr>
<td>ACTIVITY</td>
<td>IDEA</td>
<td>RELATION</td>
</tr>
<tr>
<td>ANIMAL</td>
<td>INFORMATION</td>
<td>RELATIONSHIP</td>
</tr>
<tr>
<td>APPRAISAL</td>
<td>INSTRUMENTATION</td>
<td>RELIGION</td>
</tr>
<tr>
<td>ART</td>
<td>INVESTIGATION</td>
<td>REPRESENTATION</td>
</tr>
<tr>
<td>ARTIFACT</td>
<td>LANGUAGE</td>
<td>RESULT</td>
</tr>
<tr>
<td>ATTITUDE</td>
<td>LANGUAGE_UNIT</td>
<td>ROOM</td>
</tr>
<tr>
<td>ATTRIBUTE</td>
<td>LETTER</td>
<td>SHOW</td>
</tr>
<tr>
<td>CITY</td>
<td>LINEAR_UNIT</td>
<td>SPEECH_ACT</td>
</tr>
<tr>
<td>COLOR</td>
<td>LIQUID</td>
<td>SPORT</td>
</tr>
<tr>
<td>COMMUNICATION</td>
<td>LOCATION</td>
<td>STATE</td>
</tr>
<tr>
<td>COMPANY</td>
<td>MANNER</td>
<td>STRUCTURE</td>
</tr>
<tr>
<td>COMPONENT</td>
<td>MEDICINE</td>
<td>SUBSTANCE</td>
</tr>
<tr>
<td>CONDITION</td>
<td>MEDIUM</td>
<td>SYMBOL</td>
</tr>
<tr>
<td>COUNTRY</td>
<td>MESSAGE</td>
<td>SYSTEM</td>
</tr>
<tr>
<td>CREATION</td>
<td>MOUNTAIN</td>
<td>TEMP</td>
</tr>
<tr>
<td>CURRENCY</td>
<td>NONACHIEVEMENT</td>
<td>TIME</td>
</tr>
<tr>
<td>DATE</td>
<td>NUMBER</td>
<td>TRAIT</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>OBJECT</td>
<td>VEHICLE</td>
</tr>
<tr>
<td>DEVICE</td>
<td>ORGAN</td>
<td>VIRTUE</td>
</tr>
<tr>
<td>DISTANCE</td>
<td>ORGANIZATION</td>
<td>WEALTH</td>
</tr>
<tr>
<td>ENTITY</td>
<td>PERCEPTION</td>
<td>WEATHER</td>
</tr>
<tr>
<td>EVENT</td>
<td>PERSON</td>
<td>WEIGHT</td>
</tr>
<tr>
<td>FEELING</td>
<td>PHENOMENON</td>
<td>WORK</td>
</tr>
<tr>
<td>FIELD</td>
<td>PLANT</td>
<td>WRITING</td>
</tr>
<tr>
<td>FOOD</td>
<td>POSSESSION</td>
<td></td>
</tr>
</tbody>
</table>

Figure C.1: Semantic Types Adapted from Li and Roth (2002)
<table>
<thead>
<tr>
<th>Relation</th>
<th>Description</th>
<th>Relation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dep</td>
<td>dependent</td>
<td>mod</td>
<td>modifier</td>
</tr>
<tr>
<td>aux</td>
<td>auxiliary</td>
<td>advcl</td>
<td>adverbial clause modifier</td>
</tr>
<tr>
<td>auxpass</td>
<td>passive auxiliary</td>
<td>purpcl</td>
<td>purpose clause modifier</td>
</tr>
<tr>
<td>cop</td>
<td>copula</td>
<td>tmod</td>
<td>temporal modifier</td>
</tr>
<tr>
<td>conj</td>
<td>conjunct</td>
<td>rcmod</td>
<td>relative clause modifier</td>
</tr>
<tr>
<td>cc</td>
<td>coordination</td>
<td>amod</td>
<td>adjectival modifier</td>
</tr>
<tr>
<td>arg</td>
<td>argument</td>
<td>infmod</td>
<td>infinitival modifier</td>
</tr>
<tr>
<td>subj</td>
<td>subject</td>
<td>partmod</td>
<td>participial modifier</td>
</tr>
<tr>
<td>nsubj</td>
<td>nominal subject</td>
<td>num</td>
<td>numeric modifier</td>
</tr>
<tr>
<td>nsubjpass</td>
<td>passive nominal subject</td>
<td>number</td>
<td>compound number element</td>
</tr>
<tr>
<td>csubj</td>
<td>clausal subject</td>
<td>appos</td>
<td>appositional modifier</td>
</tr>
<tr>
<td>comp</td>
<td>complement</td>
<td>nn</td>
<td>noun compound modifier</td>
</tr>
<tr>
<td>obj</td>
<td>object</td>
<td>abbrev</td>
<td>abbreviation modifier</td>
</tr>
<tr>
<td>dobj</td>
<td>direct object</td>
<td>advmod</td>
<td>adverbial modifier</td>
</tr>
<tr>
<td>iobj</td>
<td>indirect object</td>
<td>neg</td>
<td>negation modifier</td>
</tr>
<tr>
<td>pobj</td>
<td>object of preposition</td>
<td>poss</td>
<td>possession modifier</td>
</tr>
<tr>
<td>attr</td>
<td>attributive</td>
<td>possessive</td>
<td>possessive modifier (’s)</td>
</tr>
<tr>
<td>ccomp</td>
<td>clausal complement, internal subject</td>
<td>prt</td>
<td>phrasal verb particle</td>
</tr>
<tr>
<td>xcomp</td>
<td>clausal complement, external subject</td>
<td>det</td>
<td>determiner</td>
</tr>
<tr>
<td>compl</td>
<td>complementizer</td>
<td>prep</td>
<td>prepositional modifier</td>
</tr>
<tr>
<td>mark</td>
<td>marker for an advcl</td>
<td>prep_on</td>
<td>on modifier</td>
</tr>
<tr>
<td>rel</td>
<td>relative for a rcmod</td>
<td>prep_by</td>
<td>by modifier</td>
</tr>
<tr>
<td>acomp</td>
<td>adjectival complement</td>
<td>prep_of</td>
<td>of modifier</td>
</tr>
<tr>
<td>agent</td>
<td>agent</td>
<td>sdep</td>
<td>semantic dependent</td>
</tr>
<tr>
<td>ref</td>
<td>referent</td>
<td>xsubj</td>
<td>controlling subject</td>
</tr>
<tr>
<td>expl</td>
<td>expletive (there)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table C.3: Dependency Relations from de Marneffe et al. (2006b)
To step through a concrete example of the traditional marriage algorithm described in Chapter 5 for alignment selection in the CAM design, consider the following scenario, which is a modified version of one given in Roth and Sotomayor (1990). Assume there are five men and four women, whose preferences are listed in Table D.1.

The steps of the algorithm are as follows:

1. Tim, Dan and Eli propose to Mary. Bob and Joe propose to Anne. Mary rejects Dan and Eli and says “maybe” to Tim. Anne rejects Joe and says “maybe” to Bob. Tim and Bob are now engaged to Mary and Anne, respectively.

<table>
<thead>
<tr>
<th>Tim</th>
<th>Bob</th>
<th>Joe</th>
<th>Dan</th>
<th>Eli</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary</td>
<td>Anne</td>
<td>Anne</td>
<td>Mary</td>
<td>Mary</td>
</tr>
<tr>
<td>Jane</td>
<td>Jane</td>
<td>Kate</td>
<td>Anne</td>
<td>Jane</td>
</tr>
<tr>
<td>Kate</td>
<td>Kate</td>
<td>Mary</td>
<td>Kate</td>
<td>Anne</td>
</tr>
<tr>
<td>Anne</td>
<td>Mary</td>
<td>Jane</td>
<td>Jane</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mary</th>
<th>Jane</th>
<th>Kate</th>
<th>Anne</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bob</td>
<td>Joe</td>
<td>Eli</td>
<td>Tim</td>
</tr>
<tr>
<td>Joe</td>
<td>Tim</td>
<td>Dan</td>
<td>Dan</td>
</tr>
<tr>
<td>Tim</td>
<td>Bob</td>
<td>Tim</td>
<td>Eli</td>
</tr>
<tr>
<td>Dan</td>
<td>Dan</td>
<td>Bob</td>
<td>Bob</td>
</tr>
<tr>
<td>Eli</td>
<td>Eli</td>
<td>Joe</td>
<td>Joe</td>
</tr>
</tbody>
</table>

Table D.1: Example preferences for Men and Women in TMA example
2. Joe, Dan and Eli propose to their second choices. Joe proposes to Kate, Dan proposes to Anne and Eli to Jane. Kate and Jane are unattached. But both have ranked their proposers, so Kate and Jane become engaged to Joe and Eli, respectively. Anne is already engaged to Bob, but Dan is ranked higher than Bob on her list. So, she rejects Bob and says “maybe” to Dan. Tim, Dan, Joe and Eli are now engaged. Bob is not.

3. Bob proposes to his second choice, Jane. Jane breaks her engagement to Eli and becomes engaged to Bob. Tim, Dan, Joe and Bob are now engaged. Eli is not.

4. Eli proposes to his third choice, Anne. Anne prefers her current fiancé, Dan, to Eli, so she rejects Eli and stays engaged to Dan.

5. Eli has no one left to propose to and all other men are paired with a woman from their list. Thus, the algorithm terminates with the following marriages:

   • Tim - Mary
   • Bob - Jane
   • Joe - Kate
   • Dan - Anne
   • Eli - (unmarried)
BIBLIOGRAPHY


DAVIES, NORMAN. 1976. Receptive versus Productive Skills in Foreign Language


Jurafsky, Dan and James Martin. 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice Hall.


318


Rudner, Lawrence, Veronica Garcia, and Catherine Welch. 2005. An


SHINYAMA, YUSUKE, SATOSHI SEKINE, KIYOSHI SUDO, AND RALPH GRISHMAN.


