LINGUISTICALLY MOTIVATED FEATURES FOR CCG REALIZATION RANKING

DISSERTATION

Presented in Partial Fulfillment of the Requirements for
the Degree Doctor of Philosophy in the
Graduate School of The Ohio State University

By

Rajakrishnan Rajkumar, M.A.
Graduate Program in Linguistics
The Ohio State University

2012

Dissertation Committee:
Professor Michael White, Advisor
Professor Peter Culicover
Professor William Schuler
© Copyright by

Rajakrishnan Rajkumar

2012
ABSTRACT

Natural Language Generation (NLG) is the process of generating natural language text from an input, which is a communicative goal and a database or knowledge base. Informally, the architecture of a standard NLG system consists of the following modules (Reiter and Dale, 2000): content determination, sentence planning (or microplanning) and surface realization. This thesis is about designing novel, linguistically motivated features for surface realization (the final NLG module mentioned above), the process by which text is created from an abstract representation of language according to the rules of syntax and morphology. It primarily involves three interrelated problems: constituent ordering, inflection and agreement and function word insertion. For addressing these problems, most state-of-the-art realization ranking models (Velldal and Oepen, 2005; White and Rajkumar, 2009) employ features which are based on very basic insights from linguistic theory (POS tags, rules derived from parse trees, for example). More sophisticated insights of linguistic theory have not been widely perceived as necessary for increased system performance, with very basic insights providing the most gains (similar to the situation Johnson (2009) describes in the context of natural language parsing).

In contrast, our goal is to design features motivated by insights from theoretical linguistics and also based on cognitively plausible accounts of language comprehension discussed in the linguistics literature, so that the realization ranking model can better approximate human judgements of fluency and acceptability. We show that the minimal dependency
length theory (Gibson, 1998; Temperley, 2007) helps with the constituent ordering problem in surface realization. For the problem of generating correct inflected word forms, we demonstrate that a machine learning-based approach is well-suited to encode insights from the theoretical linguistics literature on English agreement (Kathol, 1999; Pollard and Sag, 1994). This approach leads to improvements over a competitive baseline model containing \( n \)-gram and parsing features (of the kind described in Johnson, 2009). Finally, we demonstrate empirically that the uniform information density principle discussed in (Jaeger, 2010) contributes towards the that-complementizer choice in the context of surface realization.

Thus the primary contribution of this thesis is the design and evaluation of linguistically motivated features that model all the three classes of linguistic phenomena described above. We demonstrate that such features further enhance the output of a state-of-the art realization model and also lead to more fluent output. The secondary contribution of this thesis is a novel and particularly appropriate evaluation method that complements standard NLG evaluation techniques, viz. BLEU scores and human evaluations. We adopted distributional analyses to compare the output of ranking models with the gold standard corpus, as these analyses show how the output of these models match the dependency length and constituent order distributions in the reference corpus. We also evaluated how various types of complement clauses predicted by classification models consisting of various feature sets performed in comparison to the gold standard corpus. Thus these analyses also provided insights on how different feature sets fared in preferring constructions of interest.
For

My Linguistics 201 & 384 students
ACKNOWLEDGMENTS

This thesis evolved out of several research assistantships\(^1\) supervised by my academic advisor, Professor Michael White, who was closely involved in all aspects of this work. Though we had several disagreements over the years, I thank my advisor for introducing me to some challenging problems in computational and experimental linguistics. I would also like to express my profound debt of gratitude to my advisor for spending a considerable amount of his time reviewing and critiquing my work and especially for providing detailed feedback on my writing. The chapters of this thesis borrows from and further expands conference papers we jointly authored along the course of my studies here. My sincere thanks to my thesis committee members, Professors Peter Culicover and William Schuler, whose perceptive comments and pointers to existing work have contributed to the development of the ideas discussed in my thesis. The remaining problems of this work, however, are entirely my own contribution.

My time here was spent mostly in the company of fellow CL graduate students Dennis Mehay and Dominic Espinosa. I learned a lot about the practical side of computational linguistics from them. My free wheeling discussions with Omkar Lele, Andrew Plummer, DJ Hovermale and Salena Anderson (née Sampson) on the theory, practice and sociology of linguistics have also been an invaluable part of my education. From each person mentioned

\(^1\)This work was supported by Professor White’s grant NSF IIS-0812297, Linguistics 201 and 384 teaching support from the OSU department of linguistics and by an allocation of computing time from the Ohio Supercomputer Center. I am extremely grateful for this support.
so far, I have learned more about the English language than I did from all the English teachers who were in my life. At this point, I would also like to thank my mother for her consistent efforts to understand what I was actually doing for a living all these years.

This document will not be complete without a mention of the people I encountered during my years at Jawaharlal Nehru University (JNU), New Delhi, an institution which I feel exists within even greater contradictions than, perhaps, India itself. Nonetheless, a very unique place to study language for many reasons. I fondly remember the sustained support offered by Nabarun Roy, Kishore Jose, Suhail Ebrahim, Krishnakumar Trippunttura and Vinod Wayanadan all these years hence. Along with them, I would also like to acknowledge the role of my childhood friend from Trivandrum, Krishnachandran Balakrishnan, in lending credence to the view that laymen could often be hard to deal with in matters related to language.

Finally, I would like to thank my former teachers at JNU, in particular, Professors Ayesha Kidwai, Pramod Pandey, Anvita Abbi and Girish Nath Jha. Their encouragement and the wisdom imparted by their classes inspired me to pursue a doctoral degree in linguistics. A special note of thanks to JNU and OSU Linguistics alumnus, Professor Shravan Vashist, who reviewed and helped revise my OSU graduate school application back in November 2004. Since then, his blog posts and papers have provided perspective on many a weary day.

2“Laymen are generally lousy linguists: they do not know what questions to ask, they do not know how to look for answers to them and they are too ready to accept generalizations to which they could easily find counter examples.”—James D. McCawley
VITA

June, 2002 ................................. B.Tech., Mechanical Engineering
College of Engineering Trivandrum, India

May, 2004 ................................. M.A., Linguistics
Jawaharlal Nehru University, New Delhi, India

November, 2010 .......................... M.A., Linguistics
The Ohio State University, USA

June, 2012 ................................. Ph.D., Linguistics
The Ohio State University, USA

PUBLICATIONS

Eye Tracking and Speech Synthesis Evaluation


Machine Translation

**Natural Language Generation**


FIELDS OF STUDY

Major Field: Linguistics

Studies in Natural Language Generation: Professor Michael White
# 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 Tables</td>
<td>xiii</td>
</tr>
<tr>
<td>List of Figures</td>
<td>xvi</td>
</tr>
<tr>
<td><strong>Chapters:</strong></td>
<td></td>
</tr>
<tr>
<td>1. Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Contributions and Outlook</td>
<td>4</td>
</tr>
<tr>
<td>1.2 Organization</td>
<td>8</td>
</tr>
<tr>
<td>2. Background</td>
<td>10</td>
</tr>
<tr>
<td>2.1 Approaches to Surface Realization</td>
<td>10</td>
</tr>
<tr>
<td>2.1.1 Realization Ranking Approach</td>
<td>10</td>
</tr>
<tr>
<td>2.1.2 Feature Sets in Realization Ranking</td>
<td>11</td>
</tr>
<tr>
<td>2.1.3 Classification Based Surface Realization</td>
<td>15</td>
</tr>
<tr>
<td>2.1.4 Feature Sets in Classification Based Surface Realization</td>
<td>17</td>
</tr>
<tr>
<td>2.2 Evaluating Surface Realization Systems</td>
<td>20</td>
</tr>
<tr>
<td>2.2.1 Evaluating Realization Ranking Output</td>
<td>20</td>
</tr>
<tr>
<td>2.2.2 Evaluating the Output of Classifier Generation</td>
<td>23</td>
</tr>
<tr>
<td>2.3 Surface Realization with Combinatory Categorial Grammar (CCG)</td>
<td>23</td>
</tr>
<tr>
<td>2.3.1 Combinatory Categorial Grammar (CCG)</td>
<td>25</td>
</tr>
</tbody>
</table>
4. Designing Inflection and Agreement Features for Realization Ranking

4.1 Motivation for Incorporating Inflection and Agreement Features in the Statistical Ranking Model

4.2 The Heterogeneous Nature of Number Agreement
   4.2.1 Problems with a Purely Syntactic View
   4.2.2 Problems with a Purely Semantic View

4.3 The Nature of Animacy Agreement

4.4 CCG and Balanced Punctuation
   4.4.1 Absorption vs. Syntactic Features
   4.4.2 Features for Punctuation in CCG?

4.5 Feature Design
   4.5.1 Animacy and Number Agreement Features
   4.5.2 Balanced Punctuation Feature

4.6 Evaluation
   4.6.1 Experimental Conditions
   4.6.2 Results

4.7 Discussion
   4.7.1 Comparison with Hand-crafted Agreement Rules

4.8 Summary

5. Linguistically Motivated Complementizer Choice in Surface Realization

5.1 Complementizer Choice in Surface Realization

5.2 Uniform Information Density Principle and that-mentioning

5.3 Feature Design

5.4 Classification Experiment

5.5 Discussion

5.6 Summary

6. Conclusions and Overview

6.1 Conclusions and Future Work
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Features in LFG realization ranking (Cahill et al., 2007a)</td>
<td>11</td>
</tr>
<tr>
<td>2.2</td>
<td>Atomic features used in HPSG realization ranking (Nakanishi et al., 2005)</td>
<td>12</td>
</tr>
<tr>
<td>2.3</td>
<td>Information status features (Cahill and Riester, 2009)</td>
<td>14</td>
</tr>
<tr>
<td>2.4</td>
<td>Information status asymmetries (Cahill and Riester, 2009)</td>
<td>14</td>
</tr>
<tr>
<td>2.5</td>
<td>Adverb placement features designed by (Zhong and Stent, 2009)</td>
<td>17</td>
</tr>
<tr>
<td>2.6</td>
<td>Hypertagger testing on Section 00 of the uncollapsed corpus (1896 LFs &amp; 38104 predicates) &amp; partially collapsed corpus (1895 LFs &amp; 35370 predicates)</td>
<td>36</td>
</tr>
<tr>
<td>2.7</td>
<td>Basic and dependency features from Clark &amp; Curran’s (2007b) normal form model; distances are in intervening words, punctuation marks and verbs, and are capped at 3, 3 and 2, respectively</td>
<td>40</td>
</tr>
<tr>
<td>2.8</td>
<td>Perceptron Training Details—number of features in the alphabet, number of features in the model, training accuracy and training time (hours) for 10 iterations on a single commodity server</td>
<td>42</td>
</tr>
<tr>
<td>2.9</td>
<td>Legend for Experimental Conditions</td>
<td>43</td>
</tr>
<tr>
<td>2.10</td>
<td>Section 00 (98.9% coverage) and Section 23 (97.06% coverage) Results—percentage of exact match and grammatically complete realizations, BLEU scores and average times for Section 00 in seconds</td>
<td>44</td>
</tr>
<tr>
<td>2.11</td>
<td>Examples of realized output</td>
<td>45</td>
</tr>
</tbody>
</table>
2.12 Section 00 OpenCCG mismatches ................................. 46

2.13 PTB Section 23 BLEU scores and exact match percentages in the NLG literature (Nakanishi et al.’s results are for sentences of length 20 or less) . . 48

2.14 Section 23 OpenCCG realization performance over the years ............. 49

3.1 Basic head-dependent and sibling dependent ordering features .............. 73

3.2 Ranking accuracy for various dependency length definitions (tested on 716 Section 00 blind n-best lists) ........................................ 74

3.3 Legend for experimental conditions ........................................... 76

3.4 Model Sizes—number of features in alphabet for each model (satisfying count cutoff of 5) along with number active in model after 5 training epochs 77

3.5 Development (Section 00) & Test (Section 23) Set Results—exact match percentage and BLEU scores, along with statistical significance of BLEU compared to the unmarked model in each group (* = p < 0.1, ** = p < 0.05, *** = p < 0.01); significant within-group winners (at p < 0.05) are shown in bold .................................................. 78

3.6 Targeted Human Evaluation—percentage of realizations preferred by two human judges in a 2AFC test among the 25 development set sentences with the greatest differences in dependency length, with a binomial test for significance .................................................. 80

3.7 Dependency Length Compared to Corpus—percentage of realizations with dependency length less than and greater than gold standard, along with mean dependency length, whose significance is tested against gold; 1671 development set (Section 00) complete realizations analyzed .................. 81

3.8 Distribution of various kinds of post-verbal constituents in the development set (Section 00); 4692 gold cases considered ....................... 83

3.9 Distribution of heavy unequal constituents (length difference > 5) in Section 00; 4692 gold cases considered and significance tested against the gold standard using a χ-square test ........................................ 84
3.10 Examples of realized output for models with and without the dependency
Length feature (see Table 3.3 for model details) ......................... 84

3.11 Distribution of the second PTB function tag in 295 long-short sequences . 87

3.12 Results of a preliminary study analyzing argument-adjunct patterns in PTB
data using Propbank annotation ........................................... 90

4.1 Section 00 OpenCCG inflection and agreement mismatches (from 847 com-
plete, but non-exact derivations) in the output of White and Rajkumar (2009) 93

4.2 New features introduced .................................................. 110

4.3 Legend for experimental conditions .................................... 113

4.4 Results (98.9% coverage)—percentage of exact match and grammatically
complete realizations and BLEU scores .................................. 113

4.5 Section 00 METEOR and TERP scores ................................ 114

4.6 Error analysis of Section 00 complete realizations (total of 1554 agreement
cases; total of 207 relativizer cases) ...................................... 114

4.7 Examples of realized output .............................................. 116

4.8 Error rates of hand-crafted subject-verb agreement rules of Section 00 gold
standard derivations corresponding to 1554 complete realizations in Table 4.6 117

5.1 New features introduced (the prefix of each feature encodes the type of CC;
subsequent parts supply the feature name) ............................. 125

5.2 Classification accuracy results (Section 00 has 170/543 that-CCs; Section
23 has 192/579 that-CCs) .................................................... 127

5.3 Section 00 construction-wise that-CC proportions and model accuracies (to-
tal CC counts given in brackets alongside labels); gold standard obviously
has 100% accuracy; models are local that-classification and White and Ra-
jkumar’s (2009) global realization ranking model ....................... 127

5.4 Examples from model comparison ...................................... 128
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Reproduced from Filippova and Strube (2009): A tree of the noun phrase <em>all the brothers of my neighbor</em></td>
<td>15</td>
</tr>
<tr>
<td>2.2</td>
<td>Syntactic derivation from the CCGbank for <em>He has a point he wants to make [...]</em></td>
<td>24</td>
</tr>
<tr>
<td>2.3</td>
<td>Semantic dependency graph from the CCGbank for <em>He has a point he wants to make [...]</em>, along with gold-standard supertags (category labels)</td>
<td>51</td>
</tr>
<tr>
<td>2.4</td>
<td>BLEU scores plotted against human judgements of fluency and adequacy</td>
<td>52</td>
</tr>
<tr>
<td>2.5</td>
<td>Averaged perceptron training algorithm</td>
<td>52</td>
</tr>
<tr>
<td>3.1</td>
<td>Reproduced from Tily (2010): Mean per-sentence dependency lengths from Gildea and Temperley (2007). Numbers show length ratios relative to the theoretically possible minimum.</td>
<td>57</td>
</tr>
<tr>
<td>3.2</td>
<td>Reproduced from Tily (2010): Mean total dependency length for each manuscript</td>
<td>58</td>
</tr>
<tr>
<td>3.3</td>
<td>Reproduced from Wasow and Arnold (2003): Effect of complexity when length differences are controlled</td>
<td>64</td>
</tr>
<tr>
<td>3.4</td>
<td>Reproduced from Wasow and Arnold (2003): Interaction between complexity and length</td>
<td>64</td>
</tr>
<tr>
<td>3.5</td>
<td>Reproduced from Anttila et al. (2010): Correlation of different definitions of weight (log scale) with PP-realization in dative alternation</td>
<td>70</td>
</tr>
<tr>
<td>3.6</td>
<td>Reproduced from Temperley (2007): Illustrating pre-modifying adjunct sequences</td>
<td>88</td>
</tr>
</tbody>
</table>
4.1 Object extraction in CCG ........................................... 106

4.2 Crossing composition in CCG ................................. 107
CHAPTER 1: INTRODUCTION

Natural Language Generation (NLG) is the process of generating natural language text from an input, which is a communicative goal and a database or knowledge base. Informally, the architecture of a standard NLG system consists of the following modules (Reiter and Dale, 2000): content determination, sentence planning (or microplanning) and surface realization. This thesis is about designing novel, linguistically motivated features for surface realization (the final NLG module mentioned above), the process by which text is created from an abstract representation of language according to the rules of syntax and morphology. Many broad-coverage surface realization systems (Velldal and Oepen, 2005; White and Rajkumar, 2009) operate in the generate and select paradigm where for each input, the system generates multiple outputs and subsequently a stochastic ranking model picks the best output. Surface realization primarily involves three interrelated problems, viz. constituent ordering, inflection and agreement and function word insertion. We describe these problems in more detail below.

1. **Constituent ordering**: The ranking model has to choose between competing realizations exemplified below:

   (1) Czechoslovakia said [in May] [[it could seek] [$2 billion from Hungary]]
       [if the twindam contract were broken]. (WSJ0037.66)

   (2) *Czechoslovakia said [it could seek] [if the twindam contract was broken]
       [$2 billion from Hungary] [in May].

1
In the second sentence above (the output preferred by the baseline ranking model described in White and Rajkumar, 2009), the constituent *in May* (post-verbal adjunct of the verb *said*) is at the end of the sentence and there is an intervening constituent between the verb *seek* and its argument constituent *$2\ billion\ from\ Hungary*, thus leading to difficulties in interpretation.

2. **Inflection and agreement**: In surface realization, the input to the surface realizer could be underspecified to varying degrees. Generally the logical form from which text is generated encodes word lemmas and the realizer has to generate the correct word form using rules of English inflectional morphology. In the case of verbal word forms, this also interacts with the number information of the subject. As illustrated in (4) below, both the sentences are identical except for the subject-verb agreement error (emboldened) in (4) below.

(3) The **plant**, which is owned by Hollingsworth & Vose Co., **was** under contract with Lorillard to make the cigarette filters. (WSJ0003.18)

(4) *The plant*, which is owned by Hollingsworth & Vose Co., **were** under contract with Lorillard to make the cigarette filters.

Underspecification can also occur to the extent that information about a word is not at all specified in the input to the realizer. Function words often fall in this category. The following examples below illustrate the problems that can arise in surface realization when relativizers and punctuation marks are not encoded in the input and the task of generating the appropriate lexical items is left to the realizer.

(5) Neither Lorillard nor the **researchers who** studied the workers were aware of any research on smokers of the Kent cigarettes. (WSJ0003.8)
Neither Lorillard nor the researchers that studied the workers were aware of any research on smokers of the Kent cigarettes.

The panda walks into the bar. It eats shoots and leaves.

The panda walks into the bar. It eats, shoots and leaves.

Pierre Vinken, [61 years old], will join the board as a nonexecutive director Nov. 29. (WSJ0001.1)

Pierre Vinken, [61 years old] will join the board as a nonexecutive director Nov. 29.

The first two examples illustrate the importance of animacy agreement between the head noun and relativizer in relative clauses. The next two examples (adapted from Lynne Truss’s popular book, Eats Shoots and Leaves), illustrate the general importance of punctuation for parsing and generation systems dealing with written text. In (7) above, the lack of a comma in the second sentence (after eats) has the potential to induce ambiguity. One interpretation results from treating the emboldened words shoots and leaves as a conjunction of nouns, which in turn acts as the object of the verb eats. Another interpretation arises from considering all the three emboldened words as verbs being conjoined. However, in (8), the presence of the comma after eats helps communicate the second reading unambiguously. In this thesis we integrate linguistically motivated analyses of punctuation into our grammar for surface realization. In this context, the final two examples demonstrate the importance of balanced punctuation marks surrounding sentence-medial NP-appositions (e.g., 61 years old in examples (9) and (10) above).

The first i.e., noun conjunction reading is more obvious during a first pass or casual reading. In fact, the joke rests on the second example illuminating the alternate possibility.
3. **Function word insertion**: *That*-complementizers are optional function words that introduce sentential complements in English and many surface realizers do not encode complementizer choice in their input. In the Penn Treebank, they are left out roughly two-thirds of the time, thereby enhancing conciseness. While in many cases, adding or removing *that* results in an acceptable paraphrase, in the following example, the absence of *that* in (12) induces the interpretation where the adverb has to scope over the verb *say*, which the original Penn Treebank sentence avoids by including the complementizer.

(11) He said *that* [for the second month in a row], food processors reported a shortage of nonfat dry milk. (WSJ0036.61)

(12) ? He said [for the second month in a row], food processors reported a shortage of nonfat dry milk.

### 1.1 Contributions and Outlook

State-of-the-art realization ranking models (including the ones cited above) often employ features encoding very basic insights from linguistic theory (POS tags, rules derived from parse trees, for example). Examining the contribution of linguistic theory to natural language parsing, Johnson (2009) states that more sophisticated insights from linguistic theory have not been perceived as necessary for increased system performance, with very basic insights providing the most gains. Instead, parsing systems have benefited more by investing in statistical methods, specifically advanced machine learning methods. Johnson observes that in developing statistical parsing models, “shotgun” features — that is, myriad scattershot features that pay attention to superficial aspects of structure — tend to
be remarkably useful, while features based on linguistic theory seem to be of more questionable utility, with the most basic linguistic insights tending to have the greatest impact. Additionally, while arguing a case for the importance of linguistic theory, Moore (2009) highlights the importance of the knowledge of descriptive linguistics for engineering applications. In contrast, our goal is to design features motivated by insights from theoretical linguistics and also based on cognitively plausible accounts of language comprehension discussed in the linguistics literature, so that our stochastic realization ranking model can better approximate human judgements of fluency and acceptability. Thus the primary contribution of this thesis is the design and evaluation of linguistically motivated features that model all the three classes of linguistic phenomena described above. We demonstrate that such features further enhance the output of a state-of-the art realization model and also lead to more fluent output. Specifically, in the context of discriminative reranking for the OpenCCG surface realizer, this thesis provides answers to the following questions:

1. Does the dependency length theory of language comprehension and production (Gibson, 1998; Temperley, 2007) in conjunction with other factors influencing word order contribute towards better constituent ordering choices in surface realization?

Our rich feature set containing a novel global dependency length feature along with other features adapted from the parsing and realization literature, induced a modest increase in BLEU scores which was corroborated by a targeted human evaluation. In many instances, the model successfully performed in preferring realizations like (1) above (lower dependency length compared to 2) by balancing the drive to minimize

---

4The term “shotgun” feature appears in the slides for Johnson’s talk (http://www.cog.brown.edu/~mj/papers/johnson-eacl09-workshop.pdf), rather than in the paper itself.

5openccg.sf.net
dependency length with other factors like the tendency of verbal arguments to be adjacent to their heads (Hawkins, 2001).

The secondary contribution of this thesis is a novel and particularly appropriate evaluation method that complements standard NLG evaluation techniques, viz. BLEU scores and human evaluations. We adopted distributional analyses to compare the output of ranking models with the gold standard corpus as these analyses show how the models with a dependency length feature come to increasingly match the distributions in the corpus. Thus these analyses also provided insights on how different feature sets fared in preferring constructions of interest.

2. While generating text from logical forms underspecified for word forms, relativizers and punctuation marks, will the introduction of features to enforce inflected word generation and agreement choices into a ranking model for surface realization effectively model the complexity of English agreement discussed in the theoretical linguistics literature (Pollard and Sag, 1994; Kathol, 1999)?

We show that features specifically designed to better handle agreement phenomena (animacy, subject-verb and balanced punctuation agreement) when incorporated into a realization ranking model, makes fewer agreement errors compared to hand-crafted rules, while also yielding error reduction over a state-of-the-art baseline (White and Rajkumar, 2009), which contains features of the kind alluded to by Johnson, 2009, viz. $n$-gram and parsing features. Though traditionally such cases have been handled using hard-constraints in the grammar, we demonstrate empirically that our approach
is more suited to encode insights about English agreement discussed in the theore-
tical linguistics literature (illustrated by the preference of our models containing
agreement features towards the acceptable realizations in 5-10 above).

3. Do linguistically motivated features, in particular the uniform information density
principle and other features discussed in (Jaeger, 2010) contribute towards the that-
complementizer choice in the context of surface realization?

For that-complementizer prediction, our complementizer features do lead to im-
provements over a baseline ranking model (containing n-gram and parsing-based
features) and also successfully counteract the bias of n-gram models towards fewer
words (which results in the baseline model under-proposing optional complementiz-
ers and in some cases difficulties in interpretation as in 12 above). We also evaluated
how different types of complement clauses predicted by classification models con-
sisting of various feature sets performed in comparison to the gold standard corpus.

Thus this thesis demonstrates the efficacy of insights based on linguistic theory in im-
proving the results of a broad-coverage realization system, mitigating some of the concerns
expressed in Johnson (2009). Recent developments in the field of computational linguis-
tics suggest that the field is giving serious thought to designing NLP systems directly based
on explanatory accounts of human language processing. For example, Schuler et al. (2010)
used insights from theories of human memory to obtain competitive broad-coverage parsing
performance. More recently, this approach has been successfully integrated into a state-of-
the-art MT decoder, Moses (Koehn et al., 2003) by Schwartz et al. (2011) with encouraging
results. Further, engineering technologies have also been contributing to inquiries into lin-
guistic phenomena. There has been a considerable body of recent work on theme of using
statistical parsing models as psycholinguistic models of comprehension (Hale, 2001; Levy, 2008). These works follow pioneering proposals on the processing of linguistic structure, notably Fodor’s (1978) influential paper on parsing strategies to identify gap locations and Pritchett’s (1992) account of the processing of garden path structures and syntactic reanalysis. More recently, eye-tracking methodology has been used to study various aspects of the comprehension of many syntactic phenomena (Staub et al., 2006; von der Malsburg and Vasishth, 2011). This in turn has inspired studies where information theoretic metrics of language comprehension generated by statistical parsers have been validated by examining their correlation with response times obtained from eye-tracking experiments (Boston et al., 2008; Demberg and Keller, 2008; Wu et al., 2010). However, the use of NLG models to investigate aspects of language production is much less explored than recognition models for comprehension. Thus in future, the feature sets proposed as part of this work could be used in computational models of language production to study the effect of factors influencing choices in language production.

1.2 Organization

This thesis is organized in the form of following chapters. Chapter 2 provides necessary background on surface realization in general as well as details of the OpenCCG surface realizer, the system using which thesis experiments have been conducted. Chapter 3 describes how minimal dependency length considerations can help with the constituent ordering problem in surface realization. Chapter 4 describes and evaluates features designed to enforce inflected word-form and agreement choices in surface realization. Chapter 5 demonstrates the efficacy of complementizer prediction using features adapted from
Jaeger’s (2010) investigation of the uniform information density principle. Chapter 6 summarizes the main findings of the study and also presents ideas as to future work.
CHAPTER 2: BACKGROUND

This chapter provides background on surface realization in general and subsequently introduces surface realization using Combinatory Categorial Grammar (CCG). Section 2.1 introduces two influential approaches to surface realization. Section 2.2 describes evaluation metrics. Section 2.3 describes CCG surface realization and presents results of a CCG ranking model, which serves as the baseline for the rest of the experiments in this thesis. Section 2.4 critically examines the output of the baseline model and motivates the need for linguistic insight in realization ranking. Section 2.5 provides an overview of system performance as well as a comparison of the system with other surface realization systems.

2.1 Approaches to Surface Realization

A survey of the NLG literature revealed that two approaches have been influential in surface realization, viz. classification based surface realization and the realization ranking approach to surface realization. This section describes these approaches briefly with an emphasis on the feature sets deployed in previous works.

2.1.1 Realization Ranking Approach

The ranking approach to surface realization primarily involves generating a large set of candidate realizations given an abstract representation of language (commonly referred to as the input or logical form). In grammar based realization systems, grammar rules (either hand-crafted or extracted from corpora) are used to constrain the space of possible
re-orderings. For generating text, grammar-based surface realization systems generation generally use chart realization (Kay, 1996), which is analogous to chart parsing except that it is a transducer from structures to strings. Subsequently, a separate stochastic ranking model is used to choose the best possible output. The latter parts of this chapter (and the reminder of this thesis) will be primarily concerned with one such surface realization system based on CCG, viz. OpenCCG\(^6\) (described in detail later in the chapter).

2.1.2 Feature Sets in Realization Ranking

<table>
<thead>
<tr>
<th>Feature Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count of functional uncertainty paths</td>
</tr>
<tr>
<td>Count of a (&lt; GF1 &gt;) precedes a (&lt; GF2 &gt;) of the same (sub-)f-structure</td>
</tr>
<tr>
<td>Count of (&lt; GF1 &gt;) subcategorized for by a PRED (&lt; Lemma &gt;) precedes a (&lt; GF2 &gt;) subcategorized for by the same PRED</td>
</tr>
<tr>
<td>Distance between a relative clause and its antecedent</td>
</tr>
</tbody>
</table>

Table 2.1: Features in LFG realization ranking (Cahill et al., 2007a)

Earlier surface realization systems relied mostly on \(n\)-gram models to rank their outputs. For example, the Nitrogen system (Langkilde, 2000) used a bigram model and the Fergus system (Bangalore and Rambow, 2000a) used a trigram model for selecting the best output from the choices available. Though training data for \(n\)-gram models are abundant and hence the technique is easy to implement, they are not very effective in dealing with reorderings outside of their context windows and involving many constituents. A good

\(^6\)openccg.sf.net
Table 2.2: Atomic features used in HPSG realization ranking (Nakanishi et al., 2005)

<table>
<thead>
<tr>
<th>RULE</th>
<th>Name of applied schema</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIST</td>
<td>Distance between head words of daughters</td>
</tr>
<tr>
<td>COMMA</td>
<td>Whether a comma exists between daughters</td>
</tr>
<tr>
<td></td>
<td>and/or inside daughter phrases</td>
</tr>
<tr>
<td>SPAN</td>
<td>The number of words dominated by the phrase</td>
</tr>
<tr>
<td>SYM</td>
<td>The symbol of the phrasal category</td>
</tr>
<tr>
<td>WORD</td>
<td>The surface form of the head word</td>
</tr>
<tr>
<td>LE</td>
<td>The lexical entry assigned to the head word</td>
</tr>
</tbody>
</table>

illustration of this is provided by (Cahill et al., 2007a). They note that their surface realization system for German generated 144 strings for one particular input and the reference sentence (given below) was ranked seventh by their language model.

(13) Verheugen habe die Worte des Generalinspekteurs falsch interpretiert.
Verheugen had the words the-GEN inspector-general wrongly interpreted.
Verheugen had mis-interpreted the words of the inspector-general.

They point to the fact that features encoding linear ordering tendencies (e.g., subject precedes object), adjunct position and markedness of partial VP fronting would help boost more acceptable realizations. To some extent, this has been made possible due to the advent of treebanks based on influential syntactic formalisms, viz. Head Driven Phrase Structure Grammar (HPSG), Lexical Functional Grammar (LFG) and Combinatory Categorical Grammar (CCG). Grammar-based broad coverage realization systems using grammars extracted from these treebanks also started using rankers containing a combination of $n$-gram and structural features imported from prior work on statistical parsing. This is evident from an examination of the feature sets deployed in three different broad-coverage realization ranking models based on the aforementioned syntactic formalisms.
Table 2.1 illustrates syntactic features from an LFG-based ranking model (Cahill et al., 2007a). The model had three classes of features: Language model features, LFG $c$ & $f$-structure features and additional features like distance between the head noun and the relativizer, and order of grammatical functions (e.g., SUBJ precedes OBJ). Table 2.2 illustrates the atomic features used in the log-linear ranking model of the HPSG-based system described in Nakanishi et al. (2005). In addition to language model features, particular combinations of these atomic syntactic features were also used to rank realizations. These features operated on parts of a syntactic derivation and were imported from prior work on HPSG parsing (Miyao and Tsujii, 2005). Each syntactic feature consists of a combination of the properties of the rule and the left and right daughters (e.g., `<RULE, DIST, COMMA>`). Further on, Section 2.3.6 will discuss in detail the feature sets used in a CCG-based realization system (White and Rajkumar, 2009), where again the feature set consists of $n$-gram and CCG parsing features. Though these features were successful in contributing towards better realization performance (as measured by BLEU scores) of all these systems, many linguistic phenomena can be better modelled by designing more sophisticated features, as Cahill and Riester (2009) demonstrate in the context of German realization ranking, a work which we will now describe.

Cahill and Riester (2009) improve upon earlier work by incorporating features modelling information status (IS) considerations. Table 2.3 illustrates IS labels (along with the discourse context) discussed in the literature. Table 2.4 depicts certain asymmetries in the IS labels in the context of definite and indefinite words/constituents. Using a small corpus manually annotated for these IS labels, they also examined the distribution of these asymmetries in actual German text. In a broad-coverage realization setting, without extensive human annotation to assign appropriate labels to text, it is not possible to make use of these
generalizations about language as features in a ranking model. The main contribution of this work was to identify some syntactic properties corresponding to these IS labels and use those as heuristics to annotate training and test data for realization ranking. In some cases there is a direct link between the IS label and the syntactic type of the word/phrase in question. For example, the label D-GIVEN-PRONOUN always corresponds to a pronoun. However, there are many labels like NEW, ACCESSIBLE-GENERAL or ACCESSIBLE-DESCRIPTION which do not correspond to any one type of constituent. So they extracted an inventory of syntactic features corresponding to each IS label. Subsequently, these syntactic features were used to assign IS labels to the training corpus for realization ranking.

Table 2.3: Information status features (Cahill and Riester, 2009)

<table>
<thead>
<tr>
<th>Type</th>
<th>Context resource</th>
<th>IS label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definites</td>
<td>Discourse context</td>
<td>D-GIVEN</td>
</tr>
<tr>
<td></td>
<td>Knowledge context</td>
<td>ACCESSIBLE-GENERAL</td>
</tr>
<tr>
<td></td>
<td>Environment context</td>
<td>SITUATIVE</td>
</tr>
<tr>
<td></td>
<td>Bridging context</td>
<td>BRIDGING</td>
</tr>
<tr>
<td></td>
<td>Accommodation (no context)</td>
<td>ACCESSIBLE-DESCRIPTION</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type Indefinites</th>
<th>part-whole relation</th>
<th>IS label</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>unrelated to context</td>
<td>NEW</td>
</tr>
<tr>
<td></td>
<td>PARTITIVE</td>
<td>INDEF-REL</td>
</tr>
<tr>
<td></td>
<td>to previous entity</td>
<td></td>
</tr>
<tr>
<td></td>
<td>other (unspecified)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>relation to context</td>
<td></td>
</tr>
</tbody>
</table>

Table 2.4: Information status asymmetries (Cahill and Riester, 2009)

>>> D-GIV-PRO >> INDEF-REL
D-GIV-PRO >> D-GIV-CAT
D-GIV-REL >> NEW
D-GIV-PRO >> SIT
ACC-DESC >> INDEF-REL
ACC-DESC >> ACC-GEN-TY
As their results illustrate, the IS-model results in a significantly better BLEU score of 0.7534 (compared to 0.7366 for the baseline having no IS-features) and 2% more increased exact matches. Thus this work offers concrete evidence that features based on linguistically motivated considerations can help improve surface realization results.

2.1.3 Classification Based Surface Realization

![Diagram of noun phrase tree](image)

Figure 2.1: Reproduced from Filippova and Strube (2009): A tree of the noun phrase *all the brothers of my neighbor*

Classification based approaches to surface realization, use specialized probabilistic ordering models (Ringger et al., 2004) or maximum entropy ordering models (Filippova and Strube, 2007, 2009) to predict the relative positions of constituents for the given input. Typically, the input to the system is a dependency tree illustrated in Figure 2.1. For a node having $n$ dependents, $(n + 1)!$ head-dependent orders\(^7\) are possible. For example, in the figure, the head *brothers* and its three dependents can be ordered in twenty four different orders.

\(^7\)Some systems (Filippova and Strube, 2009) restrict all possible orders by means of hard grammatical constraints like determiners should precede heads.
ways (all of the brothers, all the of brothers and so on). At each node in the input, given a head and a list of unordered dependents, the task of the classification model is to predict the correct order of dependents with respect to the head. Model parameters are estimated from a training corpus of dependency trees with the help of features based on various relative ordering properties of heads and dependents. Most proposals in classification based surface realization are similar to the approach described in (Filippova and Strube, 2009). For each head, they determined the correct dependent order by performing a series of pairwise comparisons of dependents using a classification model containing weights learned from a training corpus of dependency trees. Word positions were assumed to be in random order at first and subsequently the model probabilities helped sort the list and returned the correct order. Given two adjacent constituents in the unsorted list, using model weights estimated from the training data, the probability of them being in the correct order was ascertained. If the probability was less than 0.5, the two dependents were swapped and the one on the right was compared with the next dependent and so on.

Another aspect related to the classification based approach is that multiple classifiers might have to be used in a certain sequence to generate text. Generally, classification based systems use a series of classifiers for various linguistic phenomena. These classifiers operate on different parts of the input structure and the output of one classifier acts as the input to the next one. For example Amanda Stent’s work reports results of predicting adverb placement choices (Zhong and Stent, 2009) as well as another classification model to predict dative alternation (Zhong et al., 2006) using features adapted from Bresnan’s work (Bresnan et al., 2007). One issue which has not been adequately discussed in the classification literature in this connection is an effective method of combining the predictions of the separate classifiers. If these distinct classifiers are deployed in a realization system, it prompts
the question as to how competing ordering predictions from individual classifiers should be handled. In contrast, in a ranking model approach, all constituent ordering and lexical choice decisions are integrated into a global ranking model which evaluates different choices as part of the ranking process.

### 2.1.4 Feature Sets in Classification Based Surface Realization

In spite of the inherent shortcomings in the classification based approaches discussed in the previous section, most of the noteworthy advances in terms of developing novel features for constituent ordering based on insights from the theoretical linguistics literature have originated from classification based approaches (Zhong and Stent, 2009; Filippova and Strube, 2009). The next two sections illustrate the features used in the two works cited above.

**Features for English Adverb Placement**

<table>
<thead>
<tr>
<th>Type</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>lexical</td>
<td>preposition in this adverbial and in adverbial siblings 0-4; stems of lexical heads of this adverbial, its parent, non-adverbial siblings 0-4, and adverbial siblings 0-4; number of phonemes in lexical head of this adverbial and in lexical heads of adverbial siblings 0-4; number of words in this adverbial and in adverbial siblings 0-4</td>
</tr>
<tr>
<td>syntactic</td>
<td>syntactic categories of this adverbial, its parent, non-adverbial siblings 0-4, and adverbial siblings 0-4; adverbial type of this adverbial and of adverbial siblings 0-4 (one of DIR, EXT, LOC, MNR, PRP, TMP, ADV); numbers of siblings, non-adverbial siblings, and adverbial siblings</td>
</tr>
<tr>
<td>semantic</td>
<td>hypernyms of heads of this adverbial, its parent, non-adverbial siblings 0-4, and adverbial siblings 0-4; number of meanings for heads of this adverbial and adverbial siblings 0-4 (using WordNet)</td>
</tr>
<tr>
<td>sentence</td>
<td>sequence of children of S node (e.g., NP VP VP); form of sentence (declarative, imperative, interrogative, clause-other); presence of the following in the sentence: coordinating conjunction(s), subordinating conjunction(s), correlative conjunction(s), discourse cue(s) (e.g., however, therefore ), pronoun(s), definite article(s)</td>
</tr>
</tbody>
</table>

Table 2.5: Adverb placement features designed by (Zhong and Stent, 2009)
For the task of predicting the positions of adverbial constituents of a verb phrase, Zhong and Stent (2009) designed a host of features based on previous work related to adverb positioning in European Portugese (Costa, 2004). Table 2.5 shows the features they designed. Costa adopted a multi-factorial approach to adverb placement where adverb positions were conceived to be influenced by lexical, syntactic and information structure considerations. Zhong and Stent (2009) considered a wide variety of Penn Treebank adverbial constituents in their work. They approximated discourse features by means of lexical cues like definite articles and pronouns as well as sentence types like declarative, interrogative and imperative. They operate in a classification based framework, where given a verb phrase with a set of adverbial constituents, the objective is to predict the position of each verbal sibling correctly. They found that for their best classification models, models containing only syntactic features perform only about 3% worse than models containing all the features. They attributed the lesser impact of lexical and discourse features to the errors that might have been introduced while approximating these features.

**Features for Clausal Ordering of English**

Filippova and Strube (2009) used dependency trees of English as the input to their system. To order the dependents of verbal heads, their classification model used features inspired from the following properties discussed in the literature:

- Constituents in the nominative case precede those in other cases, and dative constituents often precede those in the accusative case

- The order of verbal arguments depends on the verb’s subcategorization properties

- Constituents with a definite article precede those with an indefinite one
• Pronominalized constituents precede non-pronominalized ones

• Animate referents precede inanimate ones

• Short constituents precede longer ones (the predictions made by minimal dependency length are different for pre-verbal positions)

• The preferred topic position is right after the verb

• The initial position is usually occupied by scene-setting elements and topics (heuristically determined using tpc function tag produced by parsers)

• There is a default order based on semantic properties of constituents:
  Actor < Temporal < SpaceLocative < Means < Addressee < Patient < Source < Destination < Purpose

These features were directly imported from their earlier work on German constituent ordering (Filippova and Strube, 2007). All of the above features were motivated by insights about various ordering phenomena as discussed in the theoretical linguistics literature and also from the features explored in Gerard Kempen’s work (e.g., Kempen and Harbusch, 2004) in the context of tree linearization and incremental sentence generation. In their approach to linearization, sentences were divided into phrasal and clausal units (i.e., constituents having a finite verb as head). While simple word trigram models determined ordering choices in phrases, clause order was determined by a maximum entropy classifier which used the features described above to come up with appropriate orders. Some of the features were approximated using information from POS tags and dependency labels provided by the Stanford Parser. At the clausal level, baseline trigram models resulted
in 49% classification accuracy while a classifier model based on the features above had a classification accuracy of 67%.

### 2.2 Evaluating Surface Realization Systems

Surface realization systems have been evaluated by comparing the output produced by the system with a reference sentence. The exact property being compared differs between the realization ranking based approaches and the classification approaches as the following subsections here describe.

#### 2.2.1 Evaluating Realization Ranking Output

Realization ranking systems have generally been evaluated using automatic metrics developed for Machine Translation (MT), though recent efforts have focussed on MT-style targeted manual evaluations (Espinosa et al., 2010; Cahill and Forst, 2009). Evaluation is generally done on a specific test set of sentences from which the input to the generator is created. The basic idea behind evaluation is comparing the top-ranked output preferred by the ranking model with the original gold standard reference sentence (from which the generator inputs were created). So generally papers report exact match figures, i.e., the number of generated strings which are identical to the gold standard sentence along with standard MT metrics which measure the similarity between generated and reference sentences. One commonly used metric is the BLEU-metric used in MT (Papineni et al., 2002a). BLEU uses a modified precision measure to compare a given sentence with one or more reference sentences. Unigram, bigram, trigram and 4-gram counts computed for the generated sentences are compared with corresponding values of the reference sentences. Finally a score is allotted for the entire test corpus by multiplying the geometric means of these precision scores with a brevity penalty. BLEU has been shown to correlate moderately well with
human judgements of realization quality (Espinosa et al., 2010; Cahill, 2009), though prior work has argued that increased BLEU scores do not guarantee improved translation quality and the metric is not fair towards systems based on different working principles, for example human aided vs. fully automated machine translation (Callison-Burch et al., 2006).

One of the issues associated with the evaluation of surface realization systems is dealing with alternate, acceptable variants that are not string identical with the gold standard sentence used as the reference sentence for evaluation. Automatic metrics like BLEU are not effective in detecting acceptable variation as illustrated below using a pair of examples (reference sentence is followed by an acceptable variant generated by our system):

(14) Reference: The government’s borrowing authority dropped at midnight Tuesday to $ 2.80 trillion from $ 2.87 trillion. (WSJ0008.3)

(15) Variant: The government’s borrowing authority dropped to $ 2.80 trillion from $ 2.87 trillion at midnight Tuesday.

(16) Reference: Regarded as the father of the supercomputer, Mr. Cray was paid $ 600,000 at Cray Research last year. (WSJ0018.33)

(17) Variant: Regarded as the father of the supercomputer, Mr. Cray was paid $ 600,000 last year at Cray Research.

There has been recent work on human evaluation of surface realization outputs (Cahill and Forst, 2009; Espinosa et al., 2010), which manually evaluate the performance of surface realizers. These works evaluate $n$-best lists of realizers manually and investigate factors affecting human judgements of sentence quality and also seek to find out automatic metrics which best mirror the judgements of humans. For German realization evaluation, Cahill and Forst (2009) present results from experiments where native speakers were presented
with alternative realizations generated from LFG f-structures and ranked using a log-linear model (Cahill et al., 2007b). The gold sentence was also included as part of the set. Given such a set, raters were asked to choose one sentence which they thought was the best given some preceding context. They report that for 70% of the items, raters choose the gold standard sentence as the most natural choice. It is interesting to note that the remaining 30% of the time, other alternatives were chosen. Taking into account these recent findings, we explore alternate methods of evaluation in this work, as summarized below and expanded in later sections.

A question which has not gained much attention in broad coverage surface realization is whether reported increases in BLEU scores actually correspond to observable improvements in quality. We view this situation as problematic, not only because Callison-Burch et al. (2006) have shown that BLEU does not always rank competing systems in accord with human judgments, but also because surface realization scores are typically much higher than those in MT—where BLEU’s performance has been repeatedly assessed—even when using just one reference. Thus, in this thesis, we present targeted manual evaluation results to confirm whether BLEU score increases translate to actual gains in quality, a practice we encourage others to adopt. In this chapter, Section 2.3 shows that the BLEU score increase induced by named entity classes in language models corresponds to a significant rise in fluency. Similarly, Chapter 3 presents human judgements which corroborate the effect of BLEU score increases due to encoding the minimal dependency length preference in realization ranking. There as well as in Chapter 5, we also propose a novel evaluation technique involving distributional similarity of the realized output with the gold standard corpus of reference sentences.
2.2.2 Evaluating the Output of Classifier Generation

In contrast to realization ranking approaches, classification approaches evaluate performance on ordering decisions. The basic method of reporting results is by means of classification accuracy, i.e., the percentage of constituents whose positions have been correctly predicted by the classification model (in comparison to the gold standard order). Going beyond this, Filippova and Strube (2009) use the following additional metrics to compare the generated order of $N$ elements with the actual order:

1. Kendall’s $\tau$, $\tau = 1 - \frac{4t}{N(N-1)}$ where $t$ is the minimum number of interchanges of consecutive elements to achieve the right order

2. Edit distance related $d_i, d_i = 1 - \frac{m}{N}$ where $m$ is the minimum number of deletions combined with insertions to get to the right order (Ringger et al., 2004)

Though in principle, the metrics described above could be applied to evaluate the output of realization ranking models, as discussed in the previous section, realization ranking systems adopt exact matches and MT metrics for evaluation. Hence in this thesis also, following the dominant trend in the realization ranking literature, we confine our evaluation to BLEU scores, exact matches and targeted human evaluations wherever applicable.

2.3 Surface Realization with Combinatory Categorial Grammar (CCG)

The following sub-sections in this section describe the set-up for surface realization used to perform thesis experiments.

1. Combinatory Categorial Grammar (CCG): Description of the grammar formalism and grammar-based chart realization
Figure 2.2: Syntactic derivation from the CCGbank for *He has a point he wants to make* [...]

2. Realization from an Enhanced CCGbank: Steps to transform the CCGbank for broad-coverage grammar extraction for parsing and realization

3. Language Models: A description of three language models used for constituent ordering and lexical choice

4. Hypertagging: Lexical category assignment for chart realization

5. Perceptron Reranking: Stochastic realization ranking algorithm

6. Baseline Model Features: Features for stochastic realization ranking

7. Experimental Conditions: A description of the test conditions used in this thesis

8. Realization Results using Baseline Model Features: A presentation of results of the baseline model experiments of this thesis
2.3.1 Combinatory Categorial Grammar (CCG)

CCG (Steedman, 2000) is a unification-based categorial grammar formalism which is defined almost entirely in terms of lexical entries that encode sub-categorization information as well as syntactic feature information (e.g. number and agreement). Complementing function application as the standard means of combining a head with its argument, type-raising and composition support transparent analyses for a wide range of phenomena, including right-node raising and long distance dependencies. An example syntactic derivation appears in Figure 2.2, with a long-distance dependency between point and make. Semantic composition happens in parallel with syntactic composition, which makes it attractive for generation.

OpenCCG is a parsing/generation library which works by combining lexical categories for words using CCG rules and multi-modal extensions on rules (Baldridge, 2002) to produce derivations. Surface realization is the process by which logical forms are transduced to strings. OpenCCG uses a hybrid symbolic-statistical chart realizer (White, 2006) which takes logical forms as input and produces sentences by using CCG combinators to combine signs. Edges are grouped into equivalence classes when they have the same syntactic category and cover the same parts of the input logical form. Alternative realizations are ranked using integrated n-gram or perceptron scoring, and pruning takes place within equivalence classes of edges. To more robustly support broad coverage surface realization, OpenCCG greedily assembles fragments in the event that the realizer fails to find a complete realization.

To illustrate the input to OpenCCG, consider the semantic dependency graph in Figure 2.3. In the graph, each node has a lexical predication (e.g. make.03) and a set of semantic features (e.g. ⟨NUM⟩sg); nodes are connected via dependency relations (e.g. ⟨ARG0⟩).
Gold-standard supertags, or category labels, are also shown; see Section 2.3.4 for their role in hypertagging.) Internally, such graphs are represented using Hybrid Logic Dependency Semantics (HLDS), a dependency-based approach to representing linguistic meaning (Baldridge and Kruijff, 2002). In HLDS, each semantic head (corresponding to a node in the graph) is associated with a nominal that identifies its discourse referent, and relations between heads and their dependents are modeled as modal relations. The following section illustrates OpenCCG chart-realization using HLDS graphs, with the help of a linguistic example.

**Example of Chart Realization**

The sentence “Ted adores a musician that Bob saw” receives the representation (White, 2006) in (18) below:

\[(18) \quad \@_e (\text{adore} \land \langle \text{TENSE} \rangle \text{pres} \land \langle \text{EXP} \rangle (t \land \text{Ted}) \land \langle \text{CONT} \rangle (m \land \text{musician} \land \langle \text{DEF} \rangle \land \langle \text{GENREL} \rangle (e_2 \land \text{see} \land \langle \text{TENSE} \rangle \text{past} \land \langle \text{PERC} \rangle (b \land \text{Bob}) \land \langle \text{PHEN} \rangle m)))\]

The HLDS term may be flattened to an equivalent flat structure representation which is a conjunction of fixed size elementary predications (EPs) of 3 types, viz. lexical predications (e.g., @e adore), semantic features (e.g., @e <TENSE>pres) and dependency relations (e.g., @e <EXP>t). With respect to (18), a detailed description of the EPs is provided below.

- \(e\): Discourse referent for the event of adoring which occurs in the present
• \( t \): Discourse referent for Ted

• \( m \): Discourse referent for musician

• \( b \): Discourse referent for Bob

• \( e2 \): Discourse referent for the past event of seeing

• \( e \) & \( t \) are related by the EXP(ERIENCER) role

• \( m \) & \( e2 \) are related by the GEN(ERAL) REL(ATION)

• \( e2 \) is related to \( b \) and back to \( m \), by the dependency roles

\begin{align*}
\text{PER(CEIVER)} & \text{ and } \text{PHEN(OMENON)} \text{ respectively.}
\end{align*}

The first phase of the chart realization algorithm is lexical look-up where, \( @_e \) see triggers lexical entries for the edges corresponding to see and saw. The EPs \( @m \) musician and \( @b \) Bob trigger entries for musician and Bob respectively. \( @m \ < \text{GENREL}> \) e triggers the edge corresponding to the lexical entry for the relative pronoun that. The feature EP \( @m \ < \text{DEF}> \) - triggers the lexical entry a. Semantic features are used to constrain the lexical look-up to certain lexical items which are always used to represent particular semantic relations. For example, the feature DEF ensures that the lexical entry a which denotes an indefinite determiner is looked-up to realize the given LF. Then, during the instantiation phase, the current EP is first unified with one of the lexical entry’s EPs, and then the unification of the remaining EPs in the lexical entry is attempted against the remaining EPs in the input LF. If instantiation succeeds, an edge is created for the instantiated entry and added to the agenda. Then the edges are removed from the agenda to the chart and combined (using CCG rules) with the edges already in the chart, with any resulting new edges added to the
agenda, until no more combinations are possible and the agenda becomes empty. Concurrently, the space of generated output in the chart is constrained by n-gram and perceptron scoring over words of possible realizations, rather than generating all possible realizations and ranking them subsequently.

2.3.2 Realization from an Enhanced CCGbank

Our starting point is an enhanced version of the CCGbank (Hockenmaier and Steedman, 2007)—a corpus of CCG derivations derived from the Penn Treebank—with Propbank (Palmer et al., 2005) roles projected onto it (Boxwell and White, 2008). The following subsections describe the three major steps we effected in order to create this corpus for broad-coverage realization with OpenCCG. These steps are related to the introduction of linguistically motivated punctuation analyses, named entity annotation and finally grammar extraction and logical form creation.

Linguistically Motivated Analyses for Punctuation

To engineer a grammar from the CCGbank suitable for realization with OpenCCG, the derivations were first revised to reflect the lexicalized treatment of coordination and punctuation assumed by the multi-modal version of CCG that is implemented in OpenCCG (White and Rajkumar, 2008). Punctuation analyses were necessitated by the fact that the original CCGbank corpus does not have lexical categories for punctuation. Instead, punctuation marks carry categories derived from their part of speech tags and form part of a binary rule. It is assumed that there are no dependencies between words and punctuation marks and that the result of punctuation rules is the same as a non-punctuation category (Hockenmaier, 2005). OpenCCG does not support binary rules as multi-modal CCG does not need them and requires that every lexical item have a lexical category. Binary rules
of the form: \( s = > s \rightarrow \), \( s \) could be converted to categories of the form: \( s_1/s_1 \). In fact, this would work well for parsing, but is inadequate for generation as binary rules do not encode correctly the semantic dependencies between constituents. Consider a sentence like:

(19) Despite recent declines in yields, investors continue to pour cash into money funds. (WSJ0004.10)

The above comma category corresponding to the binary rule states that the comma applies to a sentence, while it is more natural to think of the comma as holding a sentential adjunct (like \textit{despite recent declines in yields} here) in place, because such adjuncts are invariably connected to the main sentence using commas. Moreover, a category like the one shown above would end-up over-generating as sentences and sentential complements would be generated with a comma preceding them. Binary rules also miss out many linguistic generalizations like the presence of mandatory balancing marks in sentence medial comma/dash adjuncts. Also, the result of the above function application rule could act as its own argument, producing a string of commas. The literature discusses three methods address the issue of over-generation, viz. absorption rules (Nunberg, 1990), syntactic features (Doran, 1998; Briscoe, 1994) and semantic features (White, 2006). In the context of punctuation balancing, Section 4.4 evaluates and rejects the suitability of these proposals for our system and demonstrates the efficacy of a novel solution involving punctuation balancing features in our ranking model.

In our enhanced corpus, 89 punctuation categories were created (66 commas, 14 dashes and 3 each for the rest) out of 54 classes of binary rules (37 comma, 8 dash, 3 apiece of colon, parenthesis and dots). Contexts and constructions in which punctuation marks occurred were isolated and the corpus was then restructured by inserting new CCG categories. In many cases this also involved modifying the gold standard derivations substantially. It
has been an extremely time consuming enterprise, mainly because of the way punctuation
has been analyzed at present. But in exchange, this could serve as raw material for broad
coverage grammars able to deal with a wide variety of syntactic structures and construction
types. Two very frequent comma analyses are described below:

1. **Sentential Adjuncts**: The comma in example (19) has been analysed as selecting a
sentential modifier to its left, *Despite recent declines in yields*, to result in a sentential
modifier which then selects the rest of the sentence. This results in the following
lexical category and semantics for the comma category:

\[
(20) \quad , \quad \vdash s_{1(1)}^{md=X1,mod=M/s(1)} \backslash (s(1)/s(1))
\]

\[
: \quad \mathbb{Q}_M\langle (EMPH-INTRO) + \rangle
\]

Syntactic categories and their semantics are linked by index variables in the feature
structures of categories. Index variables for semantic heads (e.g. \(X_1\)) are convention-
ally named \(X\) plus the number of the feature structure. To support modifier modifiers,
as in 20, semantic heads of modifiers are also made available through a modifier in-
dex feature, with a variable conventionally named \(M\).\(^8\) Here, the effect of combining
the comma with the phrase headed by *despite* is to add the \(\langle EMPH-INTRO \rangle +\) feature
to the *despite*-phrase’s semantics. Following Bayraktar et al. (1998), this feature in-
dicates that the comma has the discourse function of emphasizing an introductory
clause or phrase. During realization, the feature triggers the look-up of the category
in 20, and prevents the re-application of the category to its own output (as the feature
should only be realized once).

\(^8\)A limited form of default unification is used in the implementation to keep multiple modifiers from
conflicting. As the names of index variables are entirely predictable, they are suppressed in the remainder of
the paper.
2. **Verbs of reported speech**: In 21, the comma which follows *Nevertheless* and sets off the phrase headed by *said* has the category in 22:

(21) Nevertheless, said Brenda Malizia Negus, editor of Money Fund Report, yields may blip up again before they blip down because of recent rises in short-term interest rates. (wsj_0004.8)

(22) \[ \text{, } \vdash s_2/s_2/\text{punct[,]}/\ast(s_1dcl \setminus s_2dcl) \]

\[ : @_2X(\{\text{ELABREL}\} \land X1) \]

In the genre of newswire text, this construction occurs frequently with verbs of reported speech. The CCGbank derivation of 21 assigns the category \( s_1dcl \setminus s_2dcl \) to the phrase headed by *said*, the same category that is used when the phrase follows the missing sentential complement. The comma category in 22 selects for this category and a balancing comma and then converts it to a pre-sentential modifier, \( s_2/s_2 \).

Semantically, an elaboration relation is added between the main clause and the reported speech phrase. Category 22 overgenerates to some extent in that it will allow a comma at the beginning of the sentence. To prevent this, an alternative would be to make the comma explicitly select for lexical material to its left (in this case for the category of *Nevertheless*). Another possibility would be to follow Doran (1998) in analyzing the above construction by using the verb itself to select for the comma. However, since our method involves changing the gold standard derivations, and since making the verb select extra commas or having the comma select leftward material would entail substantial further changes to the derivations, we have opted to go with 22, balancing adequacy and convenience.
Named Entity Annotation

An error analysis of OpenCCG output by (Rajkumar et al., 2009) revealed that out of 2331 named entities (NEs) annotated by the BBN corpus (Weischedel and Brunstein, 2005), 238 were not realized correctly. For example, multi-word NPs like Texas Instruments Japan Ltd. were realized as Japan Texas Instruments Ltd. Accordingly, inspired by Hogan et al.’s (2007) Experiment 1, Rajkumar et al. used the BBN corpus NE annotation to collapse certain classes of NEs. But unlike Hogan et al.’s experiment where all the NEs annotated by the BBN corpus were collapsed, Rajkumar et al. chose to collapse into single tokens only NEs whose exact form can be reasonably expected to be specified in the input to the realizer. For example, while some quantificational or comparatives phrases like more than $10,000 are annotated as MONEY in the BBN corpus, Rajkumar et al. only collapse $.10,000 into an atomic unit, with more than handled compositionally according to the semantics assigned to it by the grammar. Thus, after transferring the BBN annotations to the CCGbank corpus, Rajkumar et al. (partially) collapsed NEs which are CCGbank constituents according to the following rules: (1) completely collapse the PERSON, ORGANIZATION, GPE, WORK OF ART major class type entities; (2) ignore phrases like three decades later, which are annotated as DATE entities; and (3) collapse all phrases with POS tags CD or NNP(S) or lexical items % or $, ensuring that all prototypical named entities are collapsed.

Grammar Extraction and Logical Form Creation

Further changes are necessary to support semantic dependencies rather than surface syntactic ones; in particular, the features and unification constraints in the categories related to semantically empty function words such complementizers, infinitival-to and expletive
subjects, are adjusted to reflect their purely syntactic status. In the second step, a grammar is extracted from the converted CCGbank and augmented with logical forms. Categories and unary type changing rules (corresponding to zero morphemes) are sorted by frequency and extracted if they meet the specified frequency thresholds. A separate transformation then uses a few dozen generalized templates to add logical forms to the categories, in a fashion reminiscent of (Bos, 2005). As shown in Figure 2.3, numbered semantic roles are taken from PropBank when available, and more specific relations are introduced in the categories for closed-class items such as determiners.

After logical form insertion, the extracted and augmented grammar is loaded and used to parse the sentences in the CCGbank according to the gold-standard derivation. If the derivation can be successfully followed, the parse yields a logical form which is saved along with the corpus sentence in order to later test the realizer. Currently, the algorithm succeeds in creating logical forms for 98.85% of the sentences in the development section (Sect. 00) of the converted CCGbank, and 97.06% of the sentences in the test section (Sect. 23). Of these, 95.99% of the development LFs are semantic dependency graphs with a single root, while 95.81% of the test LFs have a single root. The remaining cases, with multiple roots, are missing one or more dependencies required to form a fully connected graph. Such missing dependencies usually reflect remaining inadequacies in the logical form templates.

2.3.3 Language Models

To score partial and complete realizations OpenCCG uses three language models interpolated by rank order centroid weights. Specifically, the language models deployed are: a word trigram model, a word model with semantic classes replacing named entities and
a factored language model (Bilmes and Kirchhoff, 2003) over words, part-of-speech tags and supertags. We discuss these in more detail below. The trigram models were created using the SRILM toolkit (Stolcke, 2002) on the standard training sections (02–21) of the CCGbank, with sentence-initial words (other than proper names) uncapitalized. Training data for the semantic class–replaced model was created by replacing (collapsed) words with their NE classes, in order to address data sparsity issues caused by rare words in the same semantic class. During realization, word forms are generated, but are then replaced by their semantic classes for scoring using the semantic class–replaced model, similar to Oh and Rudnicky (2002). As the specific words may still matter, the class replaced model is interpolated at the word level with an ordinary, word-based language model, as well as with a factored language model over POS tags and supertags. While these models are considerably smaller than the ones used in (Langkilde-Geary, 2002; Velldal and Oepen, 2005), the training data does have the advantage of being in the same domain and genre. The models employ interpolated Kneser-Ney smoothing with the default frequency cutoffs.

In our realization experiments, NE collapsing in combination with language models employing NE classes induced a BLEU score increase on Section 23 of the CCGbank (0.7940 to 0.8173) compared to a baseline lacking either of these additions. A semi-automatic analysis of Section 00 revealed that most of the corrections involved proper names that are no longer mangled during realization. We examined the development section data for the same experiment and noticed that some classes of examples showed improvements. Correct adjective ordering is achieved in some cases; for example, Dutch publishing group is enforced by the class-replaced models, while all the other models realize this as publishing Dutch group. Additionally, the class-replaced model sometimes helped with animacy marking on

9With CCG, supertags (Bangalore and Joshi, 1999) are lexical categories considered as fine-grained syntactic labels.
relative pronouns, as in _Mr. Otero, who . . ._ instead of _Mr. Otero, which . . ._. However, other examples were made worse, and some were just changed to different, but equally acceptable paraphrases. For this reason, we carried out a targeted manual evaluation to confirm the BLEU results. Along the lines of (Callison-Burch et al., 2006), two native speakers\(^{10}\) provided ratings for a random sample of 49 realizations that differed between the baseline and best conditions on the collapsed corpus. Note that the selection procedure excluded exact matches and thus focused on sentences whose realization quality may be lower on average than in an arbitrary sample. Sentences were rated in the context of the preceding sentence (if any) for both fluency and adequacy in comparison to the original sentence. The judges were not aware of the condition (best/baseline) while doing the rating. Ratings of the two judges were averaged for each item. In the human evaluation, the best system’s mean scores were 4.4 for adequacy and 3.61 for fluency, compared with the baseline’s scores of 4.35 and 3.36 respectively. Figure 2.4 shows these results including the standard error for each measurement, with the BLEU scores for this specific test set. The sample size was sufficient to show that the increase in fluency from 3.36 to 3.61 represented a significant difference (paired t-test, 1-tailed, \(p = 0.015\)), while the adequacy scores did not differ significantly.

The trigram factored language model mentioned before chains a POS model with a supertag model, where the POS model \((P)\) conditions on the previous two POS tags, and the supertag model \((S)\) conditions on the previous two POS tags as well as the current one, as shown below:

\[
p^{PS}(\vec{F}_i \mid \vec{F}_{i-2}) = p(P_i \mid P_{i-2})p(S_i \mid P_{i-2}) \tag{2.1}
\]

\(^{10}\)I thank Michael White and Dominic Espinosa for providing the ratings.
It should be noted that the use of supertags in the factored language model to score possible realizations is distinct from the prediction of supertags for lexical category assignment: the former takes the words in the local context into account (as in supertagging for parsing), while the latter takes features of the logical form into account. This latter process we call hypertagging, to which we now turn.

### 2.3.4 Hypertagging

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Condition</th>
<th>Tags/pred</th>
<th>Pred</th>
<th>Graph</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uncollapsed</td>
<td>Baseline</td>
<td>1.0</td>
<td>93.56%</td>
<td>39.14%</td>
</tr>
<tr>
<td></td>
<td>Baseline</td>
<td>1.5</td>
<td>98.28%</td>
<td>78.06%</td>
</tr>
<tr>
<td>Partly Collapsed</td>
<td>Baseline</td>
<td>1.0</td>
<td>92.22%</td>
<td>35.04%</td>
</tr>
<tr>
<td></td>
<td>Baseline+NE</td>
<td>1.0</td>
<td>92.89%</td>
<td>38.31%</td>
</tr>
<tr>
<td></td>
<td>Baseline</td>
<td>1.5</td>
<td>97.87%</td>
<td>73.14%</td>
</tr>
<tr>
<td></td>
<td>Baseline+NE</td>
<td>1.5</td>
<td>98.02%</td>
<td>75.30%</td>
</tr>
</tbody>
</table>

Table 2.6: Hypertagger testing on Section 00 of the uncollapsed corpus (1896 LFs & 38104 predicates) & partially collapsed corpus (1895 LFs & 35370 predicates)

A crucial component of the OpenCCG realizer is the **hypertagger** (Espinosa et al., 2008), or supertagger for surface realization, which uses a maximum entropy model to assign the most likely lexical categories to the predicates in the input logical form, thereby greatly constraining the realizer’s search space.\(^\text{11}\) Figure 2.3 shows gold-standard supertags for the lexical predicates in the graph; such category labels are predicted by the hypertagger at run-time. As in recent work on using supertagging in parsing, the hypertagger operates in a multitagging paradigm (Curran et al., 2006), where a variable number of predictions

\(^{11}\)The approach has been dubbed hypertagging since it operates at a level “above” the syntax, moving from semantic representations to syntactic categories.
are made per input predicate. Instead of basing category assignment on linear word and POS context, however, the hypertagger predicts lexical categories based on contexts within a directed graph structure representing the logical form (LF) of the sentence to be realized. The hypertagger generalizes Bangalore and Rambow’s (2000b) method of using supertags in generation by using maximum entropy models with a larger local context. During realization, the hypertagger returns a $\beta$-best list of supertags in order of decreasing probability. Increasing the number of categories returned clearly increases the likelihood that the most-correct supertag is among them, but at a corresponding cost in chart size. Accordingly, the hypertagger begins with a highly restrictive value for $\beta$, and backs off to progressively less-restrictive values if no complete realization can be found using the set of supertags returned. Clark and Curran (2007b) have shown this iterative relaxation strategy to be highly effective in CCG parsing.

We also examined the impact of named entity information on hypertagging. We experimented with a hypertagging model trained over a corpus with partial NE collapsing. We also trained a model where the semantic classes of the elementary lexical predications, along with the class features of their adjacent nodes, were added as features. Table 2.6 indicates the results of our experiments. The hypertagging model does worse in terms of per-logical predication accuracy & per-whole-graph accuracy on the collapsed corpus. To some extent this is not surprising, as collapsing eliminates many easy tagging cases; however, a full explanation is still under investigation. Note that class information does improve performance somewhat on the collapsed corpus.
2.3.5 Perceptron Reranking

As Collins (2002) observes, perceptron training involves a simple, on-line algorithm, with few iterations typically required to achieve good performance. Moreover, *averaged* perceptrons—which approximate voted perceptrons, a maximum-margin method with attractive theoretical properties—seem to work remarkably well in practice, while adding little further complexity. Additionally, since features only take on non-zero values when they appear in training items requiring updates, perceptrons integrate feature selection with model training, and often produce quite small models, especially when starting with a good baseline.

The generic averaged perceptron training algorithm appears in Figure 2.5. In our case, the algorithm trains a model for reranking the $n$-best realizations generated using our existing factored language model for scoring, with the oracle-best realization considered the correct answer. Accordingly, the input to the algorithm is a list of pairs $(x_i, y_i)$, where $x_i$ is a logical form, $\text{GEN}(x_i)$ are the $n$-best realizations for $x_i$, and $y_i$ is the oracle-best member of $\text{GEN}(x_i)$. The oracle-best realization is determined using a 4-gram precision metric (approximating BLEU) against the reference sentence.

We have followed Huang (2008) in using oracle-best targets for training, rather than gold standard ones, in order to better approximate test conditions during training. However, following Clark and Curran (2007a), during training we seed the realizer with the gold-standard supertags, augmenting the hypertagger’s $\beta$-best list, in order to ensure that the $n$-best realizations are generally of high quality; consequently, the gold standard realization (i.e., the corpus sentence) usually appears in the $n$-best list.\footnote{As in Clark & Curran’s approach, we use a single $\beta$ value during training, rather than iteratively loosening the $\beta$ value; the chosen $\beta$ value determines the size of the discrimination space.} In addition, we use
a hypertagger trained on all the training data, to improve hypertagger performance, while excluding the current training section (in jack-knifed fashion) from the word-based parts of the language model, in order to make the language model scores more realistic. It remains for future work to determine whether using a different compromise between ensuring high-quality training data and remaining faithful to the test conditions would yield better results.

Since realization of the $n$-best lists for training is the most time-consuming part of the process, in our current implementation we perform this step once, generating event files along the way containing feature vectors for each candidate realization. The event files are used to calculate the frequency distribution for the features, and minimum cutoffs are chosen to trim the feature alphabet to a reasonable size. Training then takes place by iterating over the event files, ignoring features that do not appear in the alphabet. As Figure 2.5 indicates, training consists of calculating the top-ranked realization according to the current model $\alpha$, and performing an update when the top-ranked realization does not match the oracle-best realization. Updates to the model add the feature vector $\Phi(x_i, y_i)$ for the missed oracle-best realization, and subtract the feature vector $\Phi(x_i, z_i)$ for the mistakenly top-ranked realization. The final model averages the models across the $T$ iterations over the training data, and $N$ test cases within each iteration.

Note that while training the perceptron model involves $n$-best reranking, realization with the resulting model can be viewed as forest rescoring, since scoring of all partial realizations is integrated into the realizer’s beam search. In future work, we intend to investigate saving the realizer’s packed charts, rather than event files, and integrating the unpacking of the charts with the perceptron training algorithm.
Table 2.7: Basic and dependency features from Clark & Curran’s (2007b) normal form model; distances are in intervening words, punctuation marks and verbs, and are capped at 3, 3 and 2, respectively

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>LexCat + Word</td>
<td>s/s/np + before</td>
</tr>
<tr>
<td>LexCat + POS</td>
<td>s/s/np + IN</td>
</tr>
<tr>
<td>Rule</td>
<td>$dcl \rightarrow np\ s_{dcl}\ \setminus np$</td>
</tr>
<tr>
<td>Rule + Word</td>
<td>$dcl \rightarrow np\ s_{dcl}\ \setminus np + bought$</td>
</tr>
<tr>
<td>Rule + POS</td>
<td>$dcl \rightarrow np\ s_{dcl}\ \setminus np + VBD$</td>
</tr>
<tr>
<td>Word-Word</td>
<td>⟨company, $s_{dcl} \rightarrow np\ s_{dcl}\ \setminus np$, bought⟩</td>
</tr>
<tr>
<td>Word-POS</td>
<td>⟨company, $s_{dcl} \rightarrow np\ s_{dcl}\ \setminus np$, VBD⟩</td>
</tr>
<tr>
<td>POS-Word</td>
<td>⟨NN, $s_{dcl} \rightarrow np\ s_{dcl}\ \setminus np$, bought⟩</td>
</tr>
<tr>
<td>Word + Δw</td>
<td>⟨bought, $s_{dcl} \rightarrow np\ s_{dcl}\ \setminus np$ + $d_w$⟩</td>
</tr>
<tr>
<td>POS + Δw</td>
<td>⟨VBD, $s_{dcl} \rightarrow np\ s_{dcl}\ \setminus np$ + $d_w$⟩</td>
</tr>
<tr>
<td>Word + Δp</td>
<td>⟨bought, $s_{dcl} \rightarrow np\ s_{dcl}\ \setminus np$ + $d_p$⟩</td>
</tr>
<tr>
<td>POS + Δp</td>
<td>⟨VBD, $s_{dcl} \rightarrow np\ s_{dcl}\ \setminus np$ + $d_p$⟩</td>
</tr>
<tr>
<td>Word + Δv</td>
<td>⟨bought, $s_{dcl} \rightarrow np\ s_{dcl}\ \setminus np$ + $d_v$⟩</td>
</tr>
<tr>
<td>POS + Δv</td>
<td>⟨VBD, $s_{dcl} \rightarrow np\ s_{dcl}\ \setminus np$ + $d_v$⟩</td>
</tr>
</tbody>
</table>

2.3.6 Baseline Model Features

The features we employ in our perceptron models are of three kinds:

1. **Log Prob features**: As in the log-linear models of Velldal & Oepen and Nakanishi et al., we incorporate the log probability of the candidate realization’s word sequence according to our factored language model as a single feature in the perceptron model. Since our language model linearly interpolates three component models, we also include the log prob from each component language model as a feature, so that the combination of these components can be optimized.

2. **Discriminative n-gram features**: Second, we include discriminative n-gram features in our model, following Roark et al.’s (2004) approach to discriminative n-gram
modeling for speech recognition. By discriminative \( n \)-gram features, we mean features counting the occurrences of each \( n \)-gram that is scored by our factored language model, rather than a feature whose value is the log prob determined by the language model. As Roark et al. note, discriminative training with \( n \)-gram features has the potential to learn to negatively weight \( n \)-grams that appear in some of the \( \text{GEN}(x_i) \) candidates (sequences which are licensed by the grammar), but which never appear in the naturally occurring corpus used to train a standard, generative language model. Since our factored language model considers words, semantic classes, part-of-speech tags and supertags, our \( n \)-gram features represent a considerable generalization of the sequence-oriented features in Velldal & Oepen’s model, which never contain more than one word and do not include semantic classes.

3. **CCG Parsing features**: Third, we include syntactic features in our model by implementing Clark & Curran’s (2007b) normal form model in OpenCCG.\(^{13}\) The features of this model are listed in Table 2.7; they are integer-valued, representing counts of occurrences in a derivation. These syntactic features are quite comparable to the dominance-oriented features in the union of the Velldal & Oepen and Nakanishi et al. models, except that our feature set does not include grandparenting, which has been found to have limited utility in CCG parsing. Our syntactic features also include ones that measure the distance between headwords in terms of intervening words, punctuation marks or verbs; these features generalize the ones in Nakanishi et al.’s model. Note that in contrast to parsing, in realization distance features are non-local, since different partial realizations in the same equivalence class typically differ in

\(^{13}\)We have omitted Clark & Curran’s root features, since the category we use for the full stop ensures that it must appear at the root of any complete derivation.
word order; as we are working in a reranking paradigm though, the non-local nature of these features is unproblematic.

More generally, the feature sets described above are similar to the feature sets which had been proposed for HPSG and LFG surface realization (described earlier in Section 2.1.2). These features form the baseline for the rest of the experiments described in this work.

First, we provide experimental results using these feature sets.

### 2.3.7 Experimental Conditions

<table>
<thead>
<tr>
<th>Model</th>
<th>#Alph-feats</th>
<th>#Feats</th>
<th>Acc</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>full-model</td>
<td>2402173</td>
<td>576176</td>
<td>96.40%</td>
<td>08:53</td>
</tr>
<tr>
<td>lp-ngram</td>
<td>1127437</td>
<td>342025</td>
<td>94.52%</td>
<td>05:19</td>
</tr>
<tr>
<td>lp-syn</td>
<td>1274740</td>
<td>291728</td>
<td>85.03%</td>
<td>05:57</td>
</tr>
</tbody>
</table>

Table 2.8: Perceptron Training Details—number of features in the alphabet, number of features in the model, training accuracy and training time (hours) for 10 iterations on a single commodity server.

For the realization ranking experiments reported in this thesis, we used a lexico-grammar extracted from Sections 02–21 of our enhanced CCGbank, a hypertagging model incorporating named entity class features, and a trigram factored language model over words, named entity classes, part-of-speech tags and supertags, as described in the preceding section. BLEU scores were calculated after removing the underscores between collapsed NEs.

Events were generated for each training section separately. As already noted, the hypertagger and POS/supertag language model was trained on all the training sections, while separate word-based models were trained excluding each of the training sections in turn. Event files for 26530 training sentences with complete realizations were generated in 7
<table>
<thead>
<tr>
<th>Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline-w3</td>
<td>No perceptron (3g wd only)</td>
</tr>
<tr>
<td>baseline</td>
<td>No perceptron</td>
</tr>
<tr>
<td>syn-only-nodist</td>
<td>All syntactic features except distance</td>
</tr>
<tr>
<td>ngram-only</td>
<td>Just ngram features</td>
</tr>
<tr>
<td>syn-only</td>
<td>Just syntactic features</td>
</tr>
<tr>
<td>lp-only</td>
<td>Just log prob features</td>
</tr>
<tr>
<td>lp-ngram</td>
<td>Log prob + Ngram features</td>
</tr>
<tr>
<td>lp-syn</td>
<td>Log prob + Syntactic features</td>
</tr>
<tr>
<td>full-model</td>
<td>Log prob + Ngram + Syntactic features</td>
</tr>
</tbody>
</table>

Table 2.9: Legend for Experimental Conditions

hours and 16 minutes on a cluster using one commodity server per section, with an average $n$-best list size of 18.2. Perceptron models were trained on single machines; details for three of the models appear in Table 2.8. The complete set of models is listed in Table 2.9.

### 2.3.8 Realization Results using Baseline Model Features

Realization results on the development section are given in Table 2.10. As the first block of rows after the baseline shows, of the models incorporating a single kind of feature, only the one with the $n$-gram log prob features beats the baseline BLEU score, with the other models falling well below the baseline (though faring better than the trigram-word LM baseline). This result confirms the importance of including $n$-gram log prob features in discriminative realization ranking models, in line with Velldal & Oepen’s findings, and contra those of Nakanishi et al., even though it was Nakanishi et al. who experimented with the Penn Treebank corpus, while Velldal & Oepen’s experiments were on a much smaller, limited domain corpus. The second block of rows shows that both the discriminative $n$-gram features and the syntactic features provide a substantial boost when used with the $n$-gram log prob features, with the syntactic features yielding a more than 3 BLEU point
Table 2.10: Section 00 (98.9% coverage) and Section 23 (97.06% coverage) Results—percentage of exact match and grammatically complete realizations, BLEU scores and average times for Section 00 in seconds

<table>
<thead>
<tr>
<th>Section</th>
<th>Model</th>
<th>%Exact</th>
<th>%Compl.</th>
<th>BLEU</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>baseline-w3</td>
<td>26.00</td>
<td>83.15</td>
<td>0.7646</td>
<td>1.8</td>
</tr>
<tr>
<td></td>
<td>baseline</td>
<td>29.00</td>
<td>83.28</td>
<td>0.7963</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>syn-only-nodist</td>
<td>26.02</td>
<td>82.69</td>
<td>0.7754</td>
<td>3.2</td>
</tr>
<tr>
<td></td>
<td>ngram-only</td>
<td>27.67</td>
<td>82.95</td>
<td>0.7777</td>
<td>3.0</td>
</tr>
<tr>
<td></td>
<td>syn-only</td>
<td>28.34</td>
<td>82.74</td>
<td>0.7838</td>
<td>3.4</td>
</tr>
<tr>
<td></td>
<td>lp-only</td>
<td>32.01</td>
<td>83.02</td>
<td>0.8009</td>
<td>2.1</td>
</tr>
<tr>
<td></td>
<td>lp-ngram</td>
<td>36.31</td>
<td>80.47</td>
<td>0.8183</td>
<td>3.1</td>
</tr>
<tr>
<td></td>
<td>lp-syn</td>
<td>39.47</td>
<td>79.74</td>
<td>0.8323</td>
<td>3.5</td>
</tr>
<tr>
<td></td>
<td>full-model</td>
<td>40.11</td>
<td>79.63</td>
<td><strong>0.8373</strong></td>
<td>3.6</td>
</tr>
</tbody>
</table>

| 23      | baseline            | 33.74  | 85.04   | 0.8173 |      |
|         | full-model          | **40.45** | 83.88 | **0.8506** |      |

Note that the baseline for Section 23 uses 4-grams and a filter for balanced punctuation (White and Rajkumar, 2008), unlike the other reported configurations, which would explain the somewhat smaller increase seen with this section.
We calculated statistical significance for the main results on the development section using bootstrap random sampling. After re-sampling 1000 times, significance was calculated using a paired t-test (999 d.f.). The results indicated that lp-only exceeded the baseline, lp-ngram and lp-syn exceeded lp-only, and the full model exceeded lp-syn, with \( p < 0.0001 \) in each case.

### 2.4 Motivation for Linguistically Motivated Features

<table>
<thead>
<tr>
<th>Ref-wsj,0020.10</th>
<th>that measure could compel Taipei’s growing number of small video-viewing parlors to pay ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline,syn-only,ngram-only</td>
<td>that measure could compel Taipei’s growing number of video-viewing small parlors to pay ...</td>
</tr>
<tr>
<td>lp-only, lp-ngram, full-model</td>
<td>that measure could compel Taipei’s growing number of <strong>small video-viewing</strong> parlors to pay ...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ref-wsj,0024.2</th>
<th>Esso Australia Ltd., a unit of new york-based Exxon Corp., and Broken Hill Pty. operate the fields ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>all except full-model</td>
<td>Esso Australia Ltd., a unit of new york-based Exxon Corp., and Broken Hill Pty. <em>operates</em> the fields ...</td>
</tr>
<tr>
<td>full-model</td>
<td>Esso Australia Ltd., a unit of new york-based Exxon Corp., and Broken Hill Pty. <em>operate</em> the fields ...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ref-wsj,0034.9</th>
<th>they fell into oblivion after the 1929 crash.</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline, lp-ngram</td>
<td>they fell after the 1929 crash into oblivion.</td>
</tr>
<tr>
<td>lp-only, ngram-only, syn-only, full-model</td>
<td>they fell into <strong>oblivion after the 1929 crash</strong>.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ref-wsj,0047.13</th>
<th>Antonio Novello , whom Mr. Bush nominated to serve as surgeon general, has reportedly assured ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline,baseline-w3, lp-syn, lp-only</td>
<td>Antonio Novello , <em>which</em> Mr. Bush nominated to serve as surgeon general, has <em>reportedly</em> assured ...</td>
</tr>
<tr>
<td>full-model, lp-ngram, syn-only, ngram-syn</td>
<td>Antonio Novello , <em>whom</em> Mr. Bush nominated to serve as surgeon general, has <em>reportedly</em> assured ...</td>
</tr>
</tbody>
</table>

Table 2.11: Examples of realized output

The motivation for designing further novel features in the ranking model was provided by a qualitative and quantitative analysis of the ranking model results described above. Table 2.11 presents four examples where the full model improves upon the baseline. Example sentence wsj,0020.10 in Table 2.11 is a case where the perceptron successfully weights the
component ngram models, as the lp-ngram model and those that build on it get it right. Note that here, the modifier ordering in small video-viewing is not specified in the logical form and either ordering is possible syntactically. In wsj_0024.2, number agreement between the conjoined subject noun phrase and verb is obtained only with the full model. This suggests that the full model is more robust in cases where the grammar is insufficiently precise (number agreement is enforced by the grammar in only the simplest cases). Example wsj_0034.9 corrects a VP ordering mismatch, where the corpus sentence is clearly preferred to the one where into oblivion is shifted to the end. Finally, wsj_0047.13 corrects an animacy mismatch on the wh-pronoun, in large part due to the high negative weight assigned to the discriminative n-gram feature PERSON, which. Note that the full model still differs from the original sentence in its placement of the adverb reportedly, choosing the arguably more natural position following the auxiliary.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Mismatch Type</th>
<th>#Mismatches</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consituent</td>
<td>Verbal constituent mismatches</td>
<td>294</td>
</tr>
<tr>
<td>Ordering</td>
<td>Acceptable variants</td>
<td>132</td>
</tr>
<tr>
<td></td>
<td>Unacceptable</td>
<td>98</td>
</tr>
<tr>
<td></td>
<td>Awkward</td>
<td>64</td>
</tr>
<tr>
<td></td>
<td>NP modifier errors</td>
<td>84</td>
</tr>
<tr>
<td>Inflection</td>
<td>Relativizer</td>
<td>67</td>
</tr>
<tr>
<td>&amp; Agreement</td>
<td>Agreement errors</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>Unbalanced punctuation</td>
<td>49</td>
</tr>
<tr>
<td>Function</td>
<td>that-complementizer</td>
<td>82</td>
</tr>
<tr>
<td>Word Insertion</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2.12: Section 00 OpenCCG mismatches
Although the current set of syntactic features substantially improved clausal constituent ordering, a variety of disfluent cases still remained as revealed by a subsequent semi-automatic error analysis. The analysis was performed by comparing the 847 complete (and also non-exact), realized derivations with their corresponding gold standard derivations. Table 2.12 depicts the types and frequency of the major types of mismatches with the gold standard which emerged. As illustrated in the error table, 294 examples involved differences in verbal constituent ordering choices. These were analyzed manually as many of these divergent cases were acceptable variants of the reference sentence. Unacceptable variants were again classified into awkward and totally unacceptable. The frequency of awkward and unacceptable examples point to the need for novel features in the ranking model which are targeted towards constituent ordering decisions that approximate human judgements of acceptability and grammaticality. We develop this theme in more detail in the next chapter. NP-modifier errors where NPs like intelligence data handling were realized as data intelligence handling arise because of the uniform right-branching analysis of NPs adopted in the CCGbank. In future we plan to tackle this problem by specifying the order of modifiers in the logical form based on the NP-structure annotation described in Vadas and Curran (2008), now available for use. Chapters 4 & 5 discuss and deal with inflected word-form generation and agreement choice issues (involving balanced punctuation, relativizers, agreement errors and complementizer mismatches) by proposing novel features.

2.5 Overview of System Performance

Before discussing other experiments we conducted, it is pertinent to place the performance of the OpenCCG realizer in the context of other broad-coverage surface realization
<table>
<thead>
<tr>
<th>System</th>
<th>Coverage</th>
<th>BLEU</th>
<th>%Exact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Callaway (05)</td>
<td>98.5%</td>
<td>0.9321</td>
<td>57.5</td>
</tr>
<tr>
<td>Bohnet et al.-S (11)</td>
<td>100%</td>
<td>0.8911</td>
<td>-</td>
</tr>
<tr>
<td><strong>OpenCCG (12)</strong></td>
<td>97.1%</td>
<td>0.8596</td>
<td>42.1</td>
</tr>
<tr>
<td>OpenCCG (09)</td>
<td>97.1%</td>
<td>0.8506</td>
<td>40.5</td>
</tr>
<tr>
<td>Ringger et al. (04)</td>
<td>100.0%</td>
<td>0.836</td>
<td>35.7</td>
</tr>
<tr>
<td>Bohnet et al.-D (11)</td>
<td>100%</td>
<td>0.7943</td>
<td>-</td>
</tr>
<tr>
<td>Langkilde-Geary (02)</td>
<td>83%</td>
<td>0.757</td>
<td>28.2</td>
</tr>
<tr>
<td>Guo et al. (08)</td>
<td>100.0%</td>
<td>0.7440</td>
<td>19.8</td>
</tr>
<tr>
<td>Hogan et al. (07)</td>
<td>≈100.0%</td>
<td>0.6882</td>
<td>-</td>
</tr>
<tr>
<td>OpenCCG (08)</td>
<td>96.0%</td>
<td>0.6701</td>
<td>16.0</td>
</tr>
<tr>
<td>Nakanishi et al. (05)</td>
<td>90.8%</td>
<td>0.7733</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 2.13: PTB Section 23 BLEU scores and exact match percentages in the NLG literature (Nakanishi et al.’s results are for sentences of length 20 or less)

systems as well as present various development of the system over the years that have led to competitive performance. While there have been quite a few papers to date reporting results on Penn Treebank data, since the various systems make different assumptions regarding the specificity of their inputs, all but the most broad-brushed comparisons remain impossible at present, and thus detailed studies such as the present one can only be made within the context of different models for the same system, a theme pursued in the next section. Some progress on this issue has been made in the context of the Generation Challenges Surface Realization Shared Task (Belz et al., 2011), but it remains to be seen to what extent fair cross-system comparisons using common inputs can be achieved. For (very) rough comparison purposes, Table 2.13 lists our results in the context of those reported for various other systems on PTB Section 23. As the table shows, the OpenCCG scores are quite competitive, exceeded only by Callaway’s (2005) extensively hand-crafted system as well as Bohnet et al.’s (2011) system on shared task shallow inputs (-S), which performs much better than their system on deep inputs (-D) that more closely resemble OpenCCG’s.
### 2.5.1 OpenCCG System Performance

<table>
<thead>
<tr>
<th>Thesis Chapter</th>
<th>Paper</th>
<th>Contribution</th>
<th>%Exact</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>White et al. (2007)</td>
<td>Language Modelling</td>
<td>9.8</td>
<td>0.5768</td>
</tr>
<tr>
<td></td>
<td>Espinosa et al. (2008)</td>
<td>Category prediction</td>
<td>16.0</td>
<td>0.6701</td>
</tr>
<tr>
<td></td>
<td>White and Rajkumar (2008)</td>
<td>Analysis of punctuation</td>
<td>21.6</td>
<td>0.7323</td>
</tr>
<tr>
<td></td>
<td>Rajkumar et al. (2009)</td>
<td>NE Collapsing, class based LM</td>
<td>33.74</td>
<td>0.8173</td>
</tr>
<tr>
<td></td>
<td>White and Rajkumar (2009)</td>
<td>Perceptron reranking</td>
<td>40.45</td>
<td>0.8506</td>
</tr>
<tr>
<td>4</td>
<td>Rajkumar and White (2010)</td>
<td>Agreement feats</td>
<td>40.09</td>
<td>0.8446</td>
</tr>
<tr>
<td>3</td>
<td>White and Rajkumar (2012)</td>
<td>Dependency length</td>
<td>42.05</td>
<td>0.8596</td>
</tr>
</tbody>
</table>

Table 2.14: Section 23 OpenCCG realization performance over the years

The OpenCCG surface realizer has evolved into a state-of-the-art broad coverage, surface realization system as a result of various enhancements in corpora and ranking models we effected over the past few years. Table 2.14 illustrates the major milestones in the development of OpenCCG. Initially White et al. (2007) used only language models for realization ranking (trigram word models interpolated with trigram pos-supertag factored language models). Subsequently, hypertagging (Espinosa et al., 2008) and an analysis of punctuation (White and Rajkumar, 2008) resulted in a dramatic BLEU score improvement. The hypertagging and ranking models when augmented with named entity information as well as NE collapsing resulted in further gains as documented in Rajkumar et al. (2009). The introduction of a perceptron ranking model with discriminative $n$-gram and syntactic features helped improve performance further (White and Rajkumar, 2009). It should also be noted that some of the performance gains are attributable to the increase in single-rooted logical forms from 69.7% to 95.8% due to our improved grammar engineering process described in Martin et al. (2009). More recently, using linguistically motivated features, we
investigated the extent to which machine learning could be used to model the acceptable variation in animacy agreement and the heterogeneous nature of number agreement (Rajkumar and White, 2010). The slight BLEU score decrease (compared to the previous row) is because of the fact that the addition of quotation marks to the corpus had an impact on realizer performance. The addition of the dependency length global feature along with other dependency ordering features also further enhanced the existing results, as shown in the last row in the table.

2.6 Summary

This chapter presented general background on two influential approaches to surface realization, viz. classification based realization as well as the realization ranking approach. The key differences between these approaches were highlighted followed by a discussion of the feature sets used. More specifically, the discussion moved on to the details of surface realization using OpenCCG, a CCG-based realization system. Details of the OpenCCG’s realization set-up, viz. corpus creation, grammar extraction, logical forms, language modelling, hypertagging and perceptron reranking were presented. Subsequently, the impact of a baseline model using features similar to what has been used in realization ranking was also described in detail. An error analysis of the output of this state-of-the-art model was conducted to motivate the need for linguistic insight in designing features for realization ranking. Finally, a comparison of the OpenCCG realization system with other realization systems was presented followed by details of system internal progress over the years.
Figure 2.3: Semantic dependency graph from the CCGbank for *He has a point he wants to make [...]*, along with gold-standard supertags (category labels)
Figure 2.4: BLEU scores plotted against human judgements of fluency and adequacy

**Input:** training examples \((x_i, y_i)\)

**Initialization:** set \(\bar{\alpha} = 0\), or use optional input model

**Algorithm:**

\[
\begin{align*}
\text{for } t = 1 \ldots T, i = 1 \ldots N \\
z_i &= \arg\max_{y \in \text{GEN}(x_i)} \Phi(x_i, y) \cdot \bar{\alpha} \\
\text{if } z_i \neq y_i \\
\bar{\alpha} &= \bar{\alpha} + \Phi(x_i, y_i) - \Phi(x_i, z_i)
\end{align*}
\]

**Output:** \(\bar{\alpha} = \frac{\sum_{t=1}^{T} \sum_{i=1}^{N} \alpha_{ti}}{TN}\)

Figure 2.5: Averaged perceptron training algorithm
CHAPTER 3: MINIMAL DEPENDENCY LENGTH IN REALIZATION RANKING

This chapter presents the results of encoding dependency length as a global feature in OpenCCG’s realization ranking model. Section 3.1 introduces the drive towards dependency length minimization in English and presents supporting evidence from the literature as well our own experiments towards this end. Section 3.2 examines the relationship between dependency length and other factors influencing constituent ordering discussed in the literature. Section 3.3 outlines the features we designed. Section 3.4 presents results from our realization experiment involving dependency length and Section 3.5 describes anti-dependency minimization tendencies in the light of our results as well as relevant discussions in the literature.

3.1 Minimal Dependency Length

Comprehension (Gibson, 1998, 2000) and corpus studies (Temperley, 2007) have found that the tendency to minimize dependency length has a strong influence on constituent ordering choices. The following examples illustrate this clearly (Tily, 2010):

(23) John took into account only his own personal acquaintances.
(24) ? John took only his own personal acquaintances into account.
(25) John took his friends into account.

The judgements here are valid only with neutral or default intonation, and out of context.
Choosing the noncanonical order (intervening constituent between the verb and the object instead of the verb followed immediately by the object) induces a much shorter dependency in (23) compared to (24). Here a long object contributes to the difference in dependency length. However, in (25) and (26), due to a short object, there is no difference in the total dependency length. Thus, there is a tendency to prefer a non-canonical order, when it substantially reduces dependency lengths, but not otherwise.

In this chapter, we show that for the constituent ordering problem in surface realization, incorporating insights from the minimal dependency length theory of language production (Temperley, 2007) into a discriminative realization ranking model yields significant improvements upon a state-of-the-art baseline. We also demonstrate the potential of dependency length to eliminate egregious ordering errors as well as better match the distributional characteristics of sentence orderings in news text. Our experiments deploy a global feature encoding the total dependency length of a given CCG derivation. Although other works in the realization literature have used head-dependent distances in their models (Filippova and Strube, 2009; Velldal and Oepen, 2005; White and Rajkumar, 2009), to the best of our knowledge, this chapter is the first to use insights from the minimal dependency theory and study its effects (both qualitatively and quantitatively). We also show that with simpler ranking models, dependency length minimization can go overboard, too often sacrificing canonical word order to shorten dependencies, while richer models manage to better counterbalance the dependency length minimization preference against competing canonical word order preferences. Before presenting our results, the chapter provides detailed background on the motivation for incorporating dependency length in our realization
models as well as the relationship between dependency length and other factors influencing constituent ordering choices.

The following sub-sections listed here present evidence from the literature which point towards dependency length minimization as a factor which has a strong influence on constituent ordering choices. Finally we also discuss examples from the output of our own surface realization experiment that point towards the need for incorporating dependency length along with other features to model constituent ordering choices better.

1. Comprehension and corpus studies (Gibson, 1998; Temperley, 2007)

2. Grammar optimization (Gildea and Temperley, 2007)

3. Evidence from language evolution and change (Tily, 2010)

4. Dependency length in statistical parsing (Eisner and Smith, 2009)

5. Results from surface realization (White and Rajkumar, 2009)

### 3.1.1 Evidence from Comprehension & Corpus Studies

A comprehensive corpus study conducted by David Temperley (Temperley, 2007) offers compelling evidence that dependency length minimization is a factor in language production. Temperley’s study is motivated by conclusions from Gibson’s Dependency Locality Theory (DLT) of language comprehension (Gibson, 1998, 2000). According to this theory, the syntactic complexity of a sentence is sum of two kinds of processing costs, viz. storage cost and integration cost. Storage cost refers to the cost of maintaining in memory the syntactic predictions or requirements of previous words. Integration cost is the cost of syntactically connecting a word to previous words with which it has dependent relations. The integration cost for a word increases with the distance to the previous words with which
it is connected, on the reasoning that the activation of words decays as they recede in time, making integration more difficult. Distance is measured in terms of the nature and the number of intervening discourse referents. Using self-paced reading experiments, Gibson demonstrates the greater processing complexity of object-extracted compared to subject-extracted relative clauses. Moreover, Tily (2010) suggests that the DLT metric is correlated with comprehension difficulty expressed in computational models of memory (Lewis et al., 2006; Lewis and Vasishth, 2005). DLT predictions have been further validated using a study involving eye-tracking data (Demberg and Keller, 2008).

Extending these ideas from comprehension, Temperley (2007) poses the question: Does language production reflect a preference for shorter dependencies in order to facilitate comprehension? By means of a study of Penn Treebank data, Temperley shows that English sentences do display a tendency to minimize the sum of all their head-dependent distances. However, in contrast to DLT which computes dependency distances in terms of the number of intervening discourse referents, Temperley computes the cost between heads and dependents in terms of words (excluding punctuation marks and adjacent words accorded a distance of 1). In phenomena involving syntactic choice, the tendency to minimize the overall dependency length is illustrated by facts like the greater length of subject noun phrases in inverted versus uninverted quotation constructions, greater length of post-modifying versus pre-modifying adverbial clauses, the tendency towards short-long ordering of post-modifying adjuncts and shorter length of the first adjunct compared to the second adjunct in clauses with three post-modifying adjuncts. Additionally, for head-final languages, dependency length minimization results in the “long-short” constituent ordering in language
production (Yamashita and Chang, 2001). The results of these corpus studies were instrumental in thinking about dependency length as a global feature in a model for ranking competing constituent ordering choices.

### 3.1.2 Evidence from Optimizing Grammars

![Figure 3.1: Reproduced from Tily (2010): Mean per-sentence dependency lengths from Gildea and Temperley (2007). Numbers show length ratios relative to the theoretically possible minimum.](image)

Gildea and Temperley (2007) report results from a tree-linearizing experiment, where given a dependency tree representation of an English sentence, the task is to order the children of each node so that the resulting sentence is a grammatical sentence of English or is very similar to English in terms of its word order properties. They investigate the problem of constructing a grammatical sentence using dynamic programming algorithms on projective tree structures to determine the word order of descendants of tree nodes. The algorithms order constituents based on the principle of minimizing dependency length and
compares the dependency length of the output with that of the baseline, actual English. Figure 3.1 shows the results of these tree linearizations. The results indicate that random linearizations have higher dependency lengths compared to actual English, while a dependency length based algorithm produces linearizations closer to actual English.

3.1.3 Evidence from Language Evolution and Change

Tily (2010) provides evidence that the pressure to minimize dependency length is significant in language change. By means of a corpus study involving the York-Toronto-Helsinki Parsed Corpus of Old English (YCOE) and PennParsed Corpus of Middle English 2 (PPCME2), Tily demonstrates this in the context of the evolution of the SVO word order in Present Day English (PDE). The diachronic trend towards dependency length minimization is depicted in 3.2. Black circles represent individual manuscripts and across time, a
curve of best fit connecting average dependency lengths was plotted as shown in the figure. As is well known, in Old English (OE) and Middle English (ME), both SVO and SOV orders were available and subjects as well as other pre-verbal dependents (including objects) were common. The study illustrates the tendency to avoid long dependencies between the verb and subject or other preverbal material by resorting to strategies like placing longer objects after the verb, thus ultimately leading to the frequent SVO order seen in Modern English.

In the context of a possible explanation for the decrease in average dependency length over time, Tily also emphasizes that Old English (or any language for that matter) should not be considered as a suboptimal system with respect to dependency length. In his discussion, Tily points out that Old English had a rich nominal case marking system and freer word order, a situation similar to that of Modern German. Adopting the view that dependency distance is a proxy for processing complexity (evidence from Demberg and Keller, 2008), difficulties associated with integrating more new words (i.e., longer dependencies) can be balanced by additional cues provided by case marking and word order. Both of these systems help integration and retrieval of new words in memory. However, the pressure to minimize dependency length was one of the factors which became dominant and this eventually lead to the evolution of Modern English, where relatively fixed word order and an impoverished case system are also observed. In contrast, Modern German still retains the rich case marking and word order freedom of Old English. Computational studies (Gildea and Temperley, 2010; Park and Levy, 2009) have also examined both Modern English and German in terms of average dependency length using tree linearization algorithms similar to the one described in the previous section. They found that though both languages are
closer to the optimal length compared to a random baseline, German is much less closer compared to English.

### 3.1.4 Dependency Length in Statistical Parsing

In their experiments connecting unlexicalized parsing and dependency length, Eisner and Smith (2009) incorporated dependency distance in their probabilistic model. The probability of each dependency was multiplied with a factor sensitive to the head-dependent distance. This factor $p(\Delta|\ldots)$ was estimated using three different variants of head-dependent distance: Directionality of attachment (signed distance between head and dependent), part-of-speech of the parent and parts-of-speech of both parent and child. They conducted parsing experiments in the case of English, German and Chinese to empirically ascertain the preference for shorter dependencies in these languages. In terms of parsing performance, for English, length-sensitive models improved dependency accuracy over their best-performing baseline (no dependency length probabilities) by 2.4%. A performance gain (though not so high) was observed in the case of Chinese, while for German their competitive baseline fared a bit better than the dependency length models (though dependency models beat some other baselines). Apart from the above cited work, the dependency parsing model described in McDonald et al. (2005) had features encoding direction of dependency attachment and parent-child distances in terms of each of the intervening POS tags. Collins (1997) also used features which measure between the distance between words in terms of number of intervening words, verbs and punctuation marks.

### 3.1.5 Results from Surface Realization

The syntactic features in White & Rajkumar’s (2009) realization ranking model are taken from Clark & Curran’s (2007b) normal form model (Table 2.7; see Section 2.3.6). In
this model, head-dependency distances are considered in conjunction with lexicalized and unlexicalized CCG derivation steps, thereby appearing in numerous features. A perceptron realization ranking model without these distance features resulted in 26.02% exact matches and a BLEU score of 0.7754 while a syntax model containing these features resulted in 28.34% exact matches and a BLEU score of 0.7838. Our analysis of the output of this model (introduced previously in Section 2.4) revealed examples like the following:

(27) Czechoslovakia said [in May] [[it could seek] [$2 billion from Hungary]] [if the twindam contract were broken]. (WSJ0037.66)

(28) * Czechoslovakia said [it could seek] [if the twindam contract was broken] [$2 billion from Hungary] [in May].

(29) People are queuing [at the door] [to take his product] but he does n’t have the working capital ...

(30) ? People are queuing [to take his product] [at the door] , but he does n’t have the working capital ...

As such, the model takes into account the interaction of dependency length with derivation steps, but in essence does not consider the main effect of dependency length itself. In this light, our investigation of dependency length minimization can be viewed as examining the question of whether realization ranking models can be made more accurate—and in particular, avoid egregious ordering errors—by incorporating a feature to account for the main effect of dependency length. (28) is a case of unacceptable ordering because the constituent in May (post-verbal adjunct of the verb said) is at the end of the sentence and there is an intervening constituent between the verb seek and its argument constituent.
$2 \text{ billion from Hungary,}$ thus leading to difficulties in interpretation. The reference sentence, (27), has lower total dependency length, providing motivation for thinking about dependency length considerations. Though (30) is an interpretable sentence, the ordering can potentially mislead the reader. The sentence can be remedied by resorting to shorter dependencies by placing the infinitival complement to take the product close to the main verb queuing, as is the case of the reference sentence, (29). Thus a perusal of the literature on language production and an error analysis convinced us of the need to include dependency length in our realization models. However, it would be reductive to consider dependency length as the only major factor influencing constituent order, a fact acknowledged in (Gildea and Temperley, 2010). A survey of the relevant linguistics and psycholinguistics literature revealed that many other interrelated factors also interact with dependency length to influence constituent ordering choices in language production. Before presenting our experimental results on dependency length in realization ranking, we explore this theme in more detail in the next section by examining the correlation between dependency length and other factors influencing ordering choices dealt with in prior work.

3.2 Dependency Length Minimization and Other Factors Influencing Constituent Ordering

Hawkins’ work (Hawkins, 1994) has been criticized as being reductive because of the tendency to explain the data in terms of a single factor, viz. length (Bresnan et al., 2007; Snider and Zaenen, 2006). A similar criticism also holds against the tendency to posit dependency length minimization as the only factor affecting constituent ordering choices. This section discusses other factors which have been described in the literature as influencing constituent ordering. We discuss the relationship of these factors with constituent
length (measure in number of words) and dependency length minimization. The results of the literature review are summarized below.

1. **Complexity**: Complexity and length are factors which independently influence constituent ordering in many constructions (Wasow, 2002; Wasow and Arnold, 2003).

2. **Animacy**: Independent of length in influencing constituent ordering choices (Snider and Zaenen, 2006)

3. **Information status considerations**: Independent of length (Wasow and Arnold, 2003; Bresnan et al., 2007)

4. **Semantic connectedness**: Length can override semantic connectedness of verb and post-verbal constituents (Hawkins, 2001).

5. **Argument-Adjunct distinction**: The tendency of verbal arguments to be closer to their heads in comparison to adjuncts (Hawkins, 2001) can override dependency length preferences.

6. **Lexical bias**: The verb influences the choice of realization in dative alternation (Wasow and Arnold, 2003; Bresnan et al., 2007) and can also influence heavy NP shift (Stallings et al., 1998; Staub et al., 2006).

7. **Prosodic factors**: Prosodic factors can be used to revise the principle of end weight by calculating weight in prosodic terms (Anttila et al., 2010). This has implications for dependency minimization calculated in terms of the number of words.

The discussion is then concluded by advocating a synthesis of competing explanations of constituent order.
3.2.1 Complexity

Following Chomsky and Miller’s (1963) original intuition that syntactic complexity could have an effect on the processing of syntactic structures independent of length, Wasow and Arnold (2003) examine effect of these factors in conjunction as well as in isolation.
Here it should be noted that their definition of complexity is the presence of a clause. To test the relationship between length and complexity they conducted a questionnaire study where subjects were asked to assign acceptability judgements to stimuli containing both complex and simple NPs (controlled for length) in both shifted as well as unshifted positions as shown below. They examined the following constructions: Heavy Noun Phrase Shift (HNPS), dative Alternation and the verb-particle construction. The following examples from the paper illustrate the types of stimuli used:

(31) John took only the people he knew into account. [Complex, Unshifted]
(32) John took into account only the people he knew. [Complex, Shifted]
(33) John took only his own personal acquaintances into account. [Simple, Unshifted]
(34) John took into account only his own personal acquaintances. [Simple, Shifted]

Figure 3.3 illustrates their results for HNPS and the verb-particle construction. Dative alternation also shows the same trend. The results show that when length is controlled, complexity independently contributes to ordering preferences. So complexity is a factor which might have a bearing on the choice between two constituent orders with equal dependency lengths. To test the effect of these factors when both of them vary, they conducted a corpus study based on the Aligned-Hansard corpus and examined the number of words and complexity in the constructions mentioned above. Figure 3.4 shows the results of this study. So when both length and complexity vary, both length and complexity are significant predictors of ordering independent of each other in the case of HNPS and dative alternation. The key finding is that it is the relative length of the constituents that determine order choices rather than the length of either one alone. But for the verb-particle construction,
length significantly contributes to ordering, while complexity does not seem to have much of an effect. This is because since the particle is a light constituent, sentences with noun phrases greater than three words always occur in the joined construction irrespective of the complexity. This work also confirms the tendency for short-long constituent orders that had been reported in the literature (in form of proposals like the principle of end weight) till then. So for HNPS and dative alternation, dependency length minimization is not the only driver in production. Structural complexity (defined as the presence of a clause) also independently influences production choices.

3.2.2 Animacy

Snider and Zaenen (2006) analyze the effect of animacy on NP fronting and the interaction between animacy and heaviness. Their study looks at these factors in the context of the general discussion of word order in terms of the Syntactic Accessibility Hierarchy which stipulates that ordering patterns adhere to the sequence SUBJ > OBJ > OBJ2 > OBL. They analyze spoken data from the Switchboard corpus using a logistic regression model to study the effect of animacy and length as predictors of topicalization and left dislocation. They conclude that inanimate entities are more likely to occupy the topic position while animate entities are more likely to be left-dislocated. Heavier constituents are likely to be topicalized or left-dislocated compared to light ones, going against purely linear order based accounts.\footnote{Peter Culicover (in p.c.) says “I suspect that it may have to do with the interaction between topicalization, intonation and focus. If something is not topicalized, and is heavy, it tends to draw the intonational accent, which marks it as focus. But if something else is focus in the neighborhood, it should receive the intonational accent. The conflict can be avoided by moving the heavy topic (which is not in focus) to clause-initial position.”} Overall, their study put forth the view that animacy and length independently influence ordering choices. Thus it is possible that considerations like the need to
foreground information may override the tendency to minimize dependency length as the next section discusses.

3.2.3 Information Status

Arnold et al. (2000) tested the effect of heaviness and newness of constituents in determining constituent order choices using a corpus study as well as a production experiment. Both length of NPs and discourse status (given vs. new element) contribute towards constituent ordering in the case of dative alternation and HNPS. Though both relative length of constituents and discourse status were significant predictors of order, heaviness accounted for more of the variation compared to discourse status. Discourse newness has an effect when heaviness does not make any predictions in either direction. In their study of dative alternation Bresnan et al. (2007) reported both these factors to be independent predictors of the choice of the dative realization. These studies point to the conclusion that discourse status is a factor which is independent of the drive to minimize dependency length and it needs to considered separately when deciding between competing ordering options.

3.2.4 Semantic Connectedness

Another factor which the literature discusses is the semantic connectedness between the verb and its dependent constituents. Wasow and Arnold (2003) discuss cases involving idioms (e.g., take our concerns into account) and collocations (e.g., bring that debate to an end). They report that 26% of non-idiom examples were in the non-canonical shifted order while around 60% of the idioms displayed shifting. Hawkins (2001) also studied the role of meaning in constituent ordering. He examined post-verbal prepositional phrases and reports that constituents with a greater semantic degree of semantic connectedness with the
verbal head (ascertained using entailment tests explained in the next section) occur more adjacent to the verb.

### 3.2.5 Complement-Adjunct distinction

Hawkins (2001) argues that complements lie closer to the verbal head because of the presence of more combinatory or dependency links between complements and heads. In this experiments, these links were detected using entailment tests of the form: “Does \( V\ PP_1\ PP_2\) entail \( V\) alone or does \( V\) have a meaning dependent on either \( PP_1\) or \( PP_2\)?” This is exemplified by the following sentences discussed in the above cited work:

(35) The man waited for his son in the early morning
(36) The man waited for his son
(37) The man counted on his son in his old age
(38) The man counted

(35) entails (36), but (37) does not entail (38) above. One other reason why complements tend to be adjacent to their verbal heads is that complements are specified in the lexical co-occurrence frame of the head as opposed to adjuncts which are not selected by the head (Pollard and Sag, 1994). Thus complements which are more central to the meaning of the sentence display a tendency to be be more adjacent to the verbal head as opposed to adjuncts. This preference often results in over-riding the preference for minimizing dependency length, a theme which we will elaborate subsequently in Section 3.5.1 in the context of analyzing patterns in the realized output.
3.2.6 Lexical Bias

Dative alternation is influenced by the verb involved as certain verbs have a bias towards the choice of the realized dative (Wasow and Arnold, 2003; Bresnan et al., 2007). Anttila et al. (2010) extend this proposal by examining the difference between one foot and two foot verbs in dative alternation. They show that the PP-choice in dative alternation and HNPS is more common with two-foot verbs. So if rhythmic feet in words are counted as part of dependency length calculations (in a revised definition), this factor is directly related to dependency length minimization. Further, in the case of heavy NP shift, both comprehension (Staub et al., 2006) and production (Stallings et al., 1998) studies have shown that the properties of individual verbs (e.g., transitivity) can influence the shifting of NPs. This has a direct effect on dependency length calculations. Thus this factor does interact with the minimal dependency length preference.

3.2.7 Prosodic Factors

In an OT-based constraint ranking framework, Anttila et al. (2010) demonstrate that the principle of end-weight (which stipulates that longer/heavier constituents come later) should be defined in terms of the number of prosodic phrases. In the literature, weight has been calculated in terms of words or syntactic nodes (Wasow, 2002). But Anttila et al. (2010) derive end-weight effects from stress and prosodic units. In an experiment which correlates eight different measure of weight with the response in dative alternation (i.e. NP vs. PP realization), they show that the log number of primary stresses in the theme shows the greatest correlation with the correct response (as show in Figure 3.5). This finding has the consequence that lexically unstressed words like function words (the, a, for example) do
not contribute towards weight. So this has implications for the calculation of dependency length in the framework discussed previously.

### 3.2.8 Dependency Length and Other Theories

Apart from specific alternate factors influencing constituent order, more generally, other theories proposed in the literature also provide explanations of constituent ordering phenomena. Minimal dependency length is only one among many plausible accounts. John Hawkins’ long line of work in defining efficiency principles for processing complexity (Hawkins, 1998, 1994, 2004) also isolates the length of constituents as a factor in syntactic processing in the face of competing constituent ordering choices. Hawkins (1994) puts forth the Early Immediate Constituents Principle (EIC), where it is hypothesized that
the human parser prefers linear orders that minimize constituent recognition domains or more specifically, maximize the ratio of number of immediate constituents (IC) to non-immediate constituents. The following examples from Hawkins (2004) help illustrate the predictions of the theory. In both the examples below, the listener encounters three immediate constituents (Verb, PP1, PP2). In the case of (39), in the course of reaching the first word of the final constituent, the listener has already encountered four words. In contrast, in the case of (41), the listener has encountered 6 words when the first word in the final constituent is being processed. So (39) is preferred by EIC because a larger IC-to-word ratio reflects less processing load for the listener compared to (41).

(39) John [ went [to London] [in the late afternoon] ]

(40) EIC: IC-to-word ratio for VP: 3/4=75%

(41) John [ went [in the late afternoon] [to London] ]

(42) EIC: IC-to-word ratio for VP: 3/6=50%

However, Temperley (2007) criticizes the above formulation on the grounds that in the case of verb phrases having three constituents, the metric does not express preferences for competing orders involving the first two constituents. As per domain minimization principles, both examples (43) and (46) below are equally preferred since the IC-to-word ratios are the same. To decide between these formulations, Temperley (2007) points to the need for more detailed investigations, preferably involving cross-lingual evidence. A revised formulation called Maximize On-line Processing (MaOP) introduced by Hawkins (2004) accounts for these cases as well. This is an online processing metric which computes processing ease and prefers orders with maximal processing ease. Thus the greater on-line ratio for (43) predicts that over (46).
Minimal dependency length also provides the same predictions as MaOP. Another important experimental study, viz. Arnold et al. (2000), puts forth the “short-first” principle reflecting production constraints. This study, along with Wasow and Arnold’s work on relative length (i.e., short-long vs. long-short orderings), makes several generalizations about post-verbal constituent ordering which are also captured by dependency length minimization. However, these two theories differ in their predictions in the case of the order of pre-modifying adjuncts, an issue discussed in more detail in Section 3.5.1.

3.3 Feature Design

Our feature design has been inspired by the conclusions of the above-cited works pertaining to the role of dependency length minimization in syntactic choice. However, going beyond Temperley’s corpus study, we confirm the utility of incorporating a feature for minimizing dependency length into machine-learned models with hundreds of thousands of features found to be useful in previous parsing and realization work, and investigate the extent to which these features can counterbalance a dependency length minimization preference in cases where canonical word order considerations should prevail.
Feature Type Example

| HeadBroadPos + Rel + Precedes + HeadWord + DepWord | ⟨VB, Arg0, dep, wants, he⟩ |
| . . . + HeadWord + DepPOS | ⟨VB, Arg0, dep, wants, PRP⟩ |
| . . . + HeadPOS + DepWord | ⟨VB, Arg0, dep, VBZ, he⟩ |
| . . . + HeadWord + DepPOS | ⟨VB, Arg0, dep, VBZ, PRP⟩ |

| HeadBroadPos + Side + DepWord1 + DepWord2 | ⟨NN, left, an, important⟩ |
| . . . + DepWord1 + DepPOS2 | ⟨NN, left, an, JJ⟩ |
| . . . + DepPOS1 + DepWord2 | ⟨NN, left, DT, important⟩ |
| . . . + DepPOS1 + DepPOS2 | ⟨NN, left, DT, JJ⟩ |
| ... + Rel1 + Rel2 | ⟨NN, left, Det, Mod⟩ |

Table 3.1: Basic head-dependent and sibling dependent ordering features

The feature sets explored in this chapter extend those in previous work on realization ranking with OpenCCG using averaged perceptron models (White and Rajkumar, 2009; Rajkumar et al., 2009; Rajkumar and White, 2010) to include more comprehensive ordering features. The feature classes are listed below, where DEPLEN, HOCKENMAIER and DEPORD are novel, and the rest are as in earlier OpenCCG models. The inclusion of the DEPORD features is intended to yield a model with a similarly rich set of ordering features as Cahill and Forster’s (2009) realization ranking model for German. Some of the non-dependency length feature sets also model constituent ordering factors discussed in Section 3.2 above, as evident from the feature description below.

**DEPLEN** The total of the length between all heads and dependents for a realization, where length is in intervening words excluding punctuation. For length purposes, collapsed named entities were counted as a single word in the experiments reported here and adjacent words were treated as having an intervening distance of 0. We also experimented with two other definitions of dependency length described in the literature,
Table 3.2: Ranking accuracy for various dependency length definitions (tested on 716 Section 00 blind \(n\)-best lists)

![Table 3.2](image)

namely (1) counting only nouns and verbs to approximate counting discourse referents\(^{17}\) (Gibson, 1998) and (2) omitting function words to approximate prosodic weight (Anttila et al., 2010). For each of the above, log length was also considered following previous corpus studies (Bresnan et al., 2007; Anttila et al., 2010). For these experiments, instead of surface realization results, we used perceptron ranking accuracy as the evaluation metric. Ranking accuracy was calculated using the number of times a perceptron model (trained using the relevant feature set) successfully preferred the top-ranked realization (computed using the modified BLEU metric). We adopted a baseline model which contained all the global features (except those based on dependency length). Subsequently different perceptron models containing all the baseline model features plus the relevant dependency length feature were trained. Table 3.2 depicts the accuracy of perceptron models trained on Sections 02-21. Testing was done on 716 blind \(n\)-best lists from Section 00. As is evident from the results, realization ranking accuracy was best for dependency length measured

\(^{17}\)As suggested by Shari Speer’s editorial comment in (Vasishth and Lewis, 2006), adverbs can also introduce discourse referents. So a revised definition of discourse referent may be appropriate in future inquiries.
by counting all non-punctuation words. Ranking accuracy was slightly worse for all the other definitions tested.

**NGRAMS** The log probabilities of the word sequence scored using three different \( n \)-gram models: a trigram word model, a trigram word model with named entity classes replacing words, and a trigram model over POS tags and supertags.

**HOCKENMAIER** As an extra component of the generative baseline, a re-implementation of Hockenmaier’s (2003) generative syntactic model.

**DISCRIMINATIVE NGRAMS** Sequences from each of the \( n \)-gram models as indicator features in the perceptron model.

**AGREEMENT** Indicator features for subject-verb and animacy agreement as well as balanced punctuation.

**C&C NF BASE** The features from Clark & Curran’s (2007b) normal form model, minus the distance features. The lexical category and word information help model lexical bias considerations discussed above.

**C&C NF DISTANCE** The distance features from the C&C normal form model. These approximate complexity considerations to some extent.

**DEPORD** Several classes of features for ordering heads and dependents as well as sibling dependents on the same side of the head. The basic features—using words, POS tags and dependency relations, grouped by the broad POS tag of the head—are shown in Table 3.1. There are also similar features using words and a word class (instead of words and POS tags), where the class is either the named entity class, COLOR for color words, PRO for pronouns, one of 60-odd suffixes culled from the web,
or HYPHEN or CAP for hyphenated or capitalized words. Additionally, there are features for detecting definiteness of an NP or PP (where the definiteness value is used in place of the POS tag). It should be noted that the dependency relation labels based on Propbank roles provide information as to the argument-adjunct status of constituents. POS and NE-class information provide cues as to animacy status.

### 3.4 Evaluation

#### 3.4.1 Experimental Conditions

<table>
<thead>
<tr>
<th>Model</th>
<th>Dep Len</th>
<th>Ngram Mods</th>
<th>Hockemaier</th>
<th>Discr Ngrams</th>
<th>Agreement</th>
<th>C&amp;C NF Base</th>
<th>C&amp;C NF Dist</th>
<th>Dep Ord</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLOBAL</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>DEPLEN-GLOBAL</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>DEPORD-NONF</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>DEPORD-NODIST</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>DEPLEN-NODIST</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>DEPORD-NF</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>DEPLEN</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

Table 3.3: Legend for experimental conditions

We followed the averaged perceptron training procedure of White and Rajkumar (2009) with a couple of updates. First, as noted earlier, we used a re-implementation of Hockemaier’s (2003) generative syntactic model as an extra component of our generative baseline; and second, only five epochs of training were used, which was found to work as well as using additional epochs on the development set. As in the earlier work, the models were trained on the standard training sections (02–21) of an enhanced version of the CCGbank, using a lexico-grammar extracted from these sections.
The models tested in the experiments reported below are summarized in Table 3.3. The three groups of models are designed to test the impact of the dependency length feature when added to feature sets of increasing complexity. In more detail, the GLOBAL and DEPLEN-GLOBAL models contain dense features on entire derivations; their values are the log probabilities of the three n-gram models used in the earlier work along with the Hockenmaier model (and the dependency length feature, in DEPLEN-GLOBAL). The second group is centered on DEPORD-NODIST, which contains all features except the dependency length feature and the distance features in Clark & Curran’s normal form model, which may indirectly capture some dependency length minimization preferences. In addition to DEPORD-NODIST—where the dependency length feature is added—this group also contains DEPORD-NONF, which is designed to test whether the Clark & Curran normal form base features are still useful even when used in conjunction with the new dependency ordering features. In the final group, DEPORD-NF contains all the features examined in this chapter except the dependency length feature, while DEPLEN contains all the features including the dependency length feature.

<table>
<thead>
<tr>
<th>Model</th>
<th># Alph Feats</th>
<th># Model Feats</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLOBAL</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>DEPLEN-GLOBAL</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>DEPORD-NONF</td>
<td>790,887</td>
<td>269,249</td>
</tr>
<tr>
<td>DEPORD-NODIST</td>
<td>1,035,915</td>
<td>365,287</td>
</tr>
<tr>
<td>DEPLEN-NODIST</td>
<td>1,035,916</td>
<td>366,094</td>
</tr>
<tr>
<td>DEPORD-NF</td>
<td>1,173,815</td>
<td>431,226</td>
</tr>
<tr>
<td>DEPLEN</td>
<td>1,173,816</td>
<td>428,775</td>
</tr>
</tbody>
</table>

Table 3.4: Model Sizes—number of features in alphabet for each model (satisfying count cutoff of 5) along with number active in model after 5 training epochs
Table 3.4 shows the sizes of the various models. For each model, the alphabet—whose size increases to over a million features—is the set of applicable features found to have discriminative value in at least 5 training examples; from these, a subset are made active (i.e., take on a non-zero weight) through perceptron updates when the feature value differs between the model-best and oracle-best realization. Note that the weight of the total dependency length feature was negative in each case, as expected.

### 3.4.2 Realization Results

<table>
<thead>
<tr>
<th>Section</th>
<th>Model</th>
<th>% Exact</th>
<th>BLEU</th>
<th>Signif</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>GLOBAL</td>
<td>33.03</td>
<td>0.8292</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>DEPLEN-GLOBAL</td>
<td>34.73</td>
<td>0.8345</td>
<td>***</td>
</tr>
<tr>
<td></td>
<td>DEPORD-NONF</td>
<td>42.33</td>
<td>0.8534</td>
<td>**</td>
</tr>
<tr>
<td></td>
<td>DEPORD-NODIST</td>
<td>43.12</td>
<td>0.8560</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>DEPLEN-NODIST</td>
<td>43.87</td>
<td>0.8587</td>
<td>***</td>
</tr>
<tr>
<td></td>
<td>DEPORD-NF</td>
<td>43.44</td>
<td>0.8590</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>DEPLEN</td>
<td>44.56</td>
<td>**.8610</td>
<td>**</td>
</tr>
<tr>
<td>23</td>
<td>GLOBAL</td>
<td>34.75</td>
<td>0.8302</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>DEPLEN-GLOBAL</td>
<td>34.70</td>
<td>0.8330</td>
<td>***</td>
</tr>
<tr>
<td></td>
<td>DEPORD-NODIST</td>
<td>41.42</td>
<td>0.8561</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>DEPLEN-NODIST</td>
<td>42.95</td>
<td>0.8603</td>
<td>***</td>
</tr>
<tr>
<td></td>
<td>DEPORD-NF</td>
<td>41.32</td>
<td>0.8577</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>DEPLEN</td>
<td>42.05</td>
<td>**.8596</td>
<td>**</td>
</tr>
</tbody>
</table>

Table 3.5: Development (Section 00) & Test (Section 23) Set Results—exact match percentage and BLEU scores, along with statistical significance of BLEU compared to the unmarked model in each group (* = \( p < 0.1 \), ** = \( p < 0.05 \), *** = \( p < 0.01 \)); significant within-group winners (at \( p < 0.05 \)) are shown in bold.
Following the usual practice in the realization ranking, we evaluate our results quantita-
tively using exact matches and BLEU (Papineni et al., 2002b), a corpus similarity metric
developed for MT evaluation described earlier in Section 2.2. Realization results for the
development and test sections appear in Table 3.5. For all three model groups, the depen-
dency length feature yields significant increases in BLEU scores, even in comparison to the
model (DEPORD-NF) containing Clark & Curran’s distance features in addition to the new
dependency ordering features (as well as all other features but total dependency length).
The second group additionally shows that the Clark & Curran normal form base features
do indeed have a significant impact on BLEU scores even when used with the new depen-
dency ordering model, as DEPORD-NONF is significantly worse than DEPORD-NODIST (the
impact of the distance features is evident in the increases from the second group to the third
group). As with the dev set, the dependency length feature yielded a significant increase in
BLEU scores for each comparison on the test set also.

For each group, the statistical significance of the difference in BLEU scores between a
model and the unmarked model (-) is determined by bootstrap resampling (Koehn, 2004).\(^{18}\)
Note that although the differences in BLEU scores are small, they end up being statistically
significant because the models frequently yield the same top scoring realization, and reli-
ably deliver improvements in the cases where they differ. In particular, note that DEPLEN
and DEPORD-NF agree on the best realization 81% of the time, while DEPLEN-NODIST
and DEPORD-NODIST have 78.1% agreement, and DEPLEN-GLOBAL and GLOBAL show
77.4% agreement; by comparison, DEPORD-NODIST and GLOBAL only agree on the best
realization 51.1% of the time. With exact matches, the dependency length feature increases

cmu.edu/MT/paired_bootstrap_v13a.tar.gz.
Table 3.6: Targeted Human Evaluation—percentage of realizations preferred by two human judges in a 2AFC test among the 25 development set sentences with the greatest differences in dependency length, with a binomial test for significance

<table>
<thead>
<tr>
<th>Model</th>
<th>% Preferred</th>
<th>% Agr</th>
<th>Signif</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLOBAL</td>
<td>22</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DEPLEN-GLOBAL</td>
<td>78</td>
<td>84</td>
<td>***</td>
</tr>
<tr>
<td>DEPORD-NODIST</td>
<td>24</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DEPLEN-NODIST</td>
<td>76</td>
<td>92</td>
<td>***</td>
</tr>
<tr>
<td>DEPORD-NF</td>
<td>26</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DEPLEN</td>
<td>74</td>
<td>96</td>
<td>***</td>
</tr>
</tbody>
</table>

the exact match percentage in each comparison group, but the differences are not statistically significant according to a chi-squared test. To ascertain whether the modest BLEU score increases translated into better constituent ordering choices, we performed a human evaluation, the results of which are documented in the next section.

3.4.3 Targeted Human Evaluation

To determine whether heavy-light ordering differences often represent ordering errors, including egregious ones such as those in Table 3.10, we conducted a targeted human evaluation on examples of this kind. Specifically, for each of the DEPLEN* models and their corresponding models without the dependency length feature, we chose the 25 sentences from the development section whose realizations exhibited the greatest difference in dependency length between sibling constituents appearing in opposite orders, and asked two judges\(^\text{19}\) to choose which of the two realizations best expressed the meaning of the reference sentence in a grammatical and fluent way, with the choice forced (2AFC). Table 3.6 shows the results. Agreement between the judges was high, with only one disagreement on

\(^\text{19}\)Thanks to Scott Martin and Dennis Mehay.
the realizations from the DEPLEN and DEPORD-NF models (involving an acceptable paraphrase), and only four disagreements on the DEPLEN-GLOBAL and GLOBAL realizations. Pooling the judgments, the preference for the DEPLEN* models was well above the chance level of 50% according to a binomial test ($p < 0.001$ in each case). Inspecting the data ourselves, we found that many of the items did indeed involve egregious ordering errors that the DEPLEN* models managed to avoid.

### 3.4.4 Distributional Analyses

<table>
<thead>
<tr>
<th>Model</th>
<th>% DL Lower</th>
<th>% DL Greater</th>
<th>DL Mean</th>
<th>Signif</th>
</tr>
</thead>
<tbody>
<tr>
<td>GOLD</td>
<td>n.a.</td>
<td>n.a.</td>
<td>41.02</td>
<td>-</td>
</tr>
<tr>
<td>GLOBAL</td>
<td>17.23</td>
<td>21.59</td>
<td>42.40</td>
<td>***</td>
</tr>
<tr>
<td>DEPLEN-GLOBAL</td>
<td>24.37</td>
<td>12.81</td>
<td>40.29</td>
<td>***</td>
</tr>
<tr>
<td>DEPORD-NONF</td>
<td>15.76</td>
<td>19.34</td>
<td>42.34</td>
<td>***</td>
</tr>
<tr>
<td>DEPORD-NODIST</td>
<td>14.58</td>
<td>19.06</td>
<td>42.03</td>
<td>***</td>
</tr>
<tr>
<td>DEPLEN-NODIST</td>
<td>17.75</td>
<td>14.82</td>
<td>40.87</td>
<td>n.s.</td>
</tr>
<tr>
<td>DEPORD-NF</td>
<td>14.96</td>
<td>17.65</td>
<td>41.58</td>
<td>***</td>
</tr>
<tr>
<td>DEPLEN</td>
<td>16.28</td>
<td>14.78</td>
<td>40.97</td>
<td>n.s.</td>
</tr>
</tbody>
</table>

Table 3.7: Dependency Length Compared to Corpus—percentage of realizations with dependency length less than and greater than gold standard, along with mean dependency length, whose significance is tested against gold; 1671 development set (Section 00) complete realizations analyzed

Going beyond BLEU scores and human evaluations, we also analyzed the distribution of dependency lengths and constituent orders in the realized models in comparison to the gold standard corpus. Adopting distributional analyses is one of the contributions of this thesis. As explained in Section 2.2, traditionally, surface realization evaluation has been done by MT evaluation metrics, with more recent works (Cahill, 2009; Espinosa et al.,
2010) paying attention to human evaluations. Complementing the above, these analyses help ascertain the extent to which specific properties in the realized output are similar to the reference corpus. For the properties of dependency length and the relative length of post-verbal constituents, we apply this technique below using the output of our realization models.

The effect of the dependency length feature on the distribution of dependency lengths is illustrated in Table 3.7. The table shows the mean of the total dependency length of each realized derivation compared to the corresponding gold standard derivation, as well as the number of derivations with greater and lower dependency length. According to paired t-tests, the mean dependency lengths for the DEPLEN-NODIST and DEPLEN models do not differ significantly from the gold standard. In contrast, the mean dependency length of all the models that do not include the dependency length feature does differ significantly ($p < 0.001$) from the gold standard. Additionally, all these models have more realizations with dependency length greater than the gold standard, in comparison to the dependency length minimizing models; this shows the efficacy of the dependency length feature in approximating the gold standard. Interestingly, the DEPLEN-GLOBAL model significantly undershoots the gold standard on mean dependency length, and has the most skewed distribution of sentences with greater vs. lesser dependency length than the gold standard.

Apart from studying dependency length directly, we also looked at one of the attested effects of dependency length minimization, viz. the tendency to prefer short-long post-verbal constituents in production (Temperley, 2007). The relative lengths of adjacent post-verbal constituents were computed and their distribution is shown in Table 3.8. Four kinds of constituents were found in the post-verbal domain. For every verb, in addition to single constituents and equal length constituents, short-long and long-short sequences were also
<table>
<thead>
<tr>
<th>Model</th>
<th>%Short Long</th>
<th>%Long Short</th>
<th>%Eq</th>
<th>%Single Const</th>
</tr>
</thead>
<tbody>
<tr>
<td>GOLD</td>
<td>25.25</td>
<td>4.87</td>
<td>4.08</td>
<td>65.79</td>
</tr>
<tr>
<td>GLOBAL</td>
<td>23.15</td>
<td>7.86</td>
<td>3.94</td>
<td>65.04</td>
</tr>
<tr>
<td>DEPLEN-GLOBAL</td>
<td>24.58</td>
<td>5.57</td>
<td>4.09</td>
<td>65.76</td>
</tr>
<tr>
<td>DEPORD-NONF</td>
<td>23.13</td>
<td>6.61</td>
<td>4.03</td>
<td>66.23</td>
</tr>
<tr>
<td>DEPORD-NODIST</td>
<td>23.38</td>
<td>6.52</td>
<td>3.94</td>
<td>66.15</td>
</tr>
<tr>
<td>DEPLEN-NODIST</td>
<td>24.03</td>
<td>5.38</td>
<td>4.01</td>
<td>66.58</td>
</tr>
<tr>
<td>DEPORD-NF</td>
<td>23.74</td>
<td>5.92</td>
<td>3.96</td>
<td>66.40</td>
</tr>
<tr>
<td>DEPLEN</td>
<td>24.36</td>
<td>5.36</td>
<td>4.07</td>
<td>66.21</td>
</tr>
</tbody>
</table>

Table 3.8: Distribution of various kinds of post-verbal constituents in the development set (Section 00); 4692 gold cases considered.

observed. Table 3.8 demonstrates that for both the gold standard corpus as well as the re-alizer models, short-long constituents were more frequent than long-short or equal length constituents. This follows the trend reported by previous corpus studies of English (Tem-perley, 2007; Wasow and Arnold, 2003). The figures reported here show the tendency of the DEPLEN* models to be closer to the gold standard than the other models, especially in the case of short-long constituents.

We also performed an analysis of relative constituent lengths focusing on light-heavy and heavy-light cases; specifically, we examined unequal length constituent sequences where the length difference of the constituents was greater than 5, and the shorter constituent was under 5 words. Table 3.9 shows the results. Using a χ-square test, the distribution of heavy unequal length constituent counts in the DEPLEN-NODIST and DEPLEN models does not significantly differ from that of the gold standard. In contrast, for all the other models, the counts do differ significantly from the gold standard.
<table>
<thead>
<tr>
<th>Model</th>
<th>% Heavy</th>
<th>% Light</th>
<th>Signif</th>
</tr>
</thead>
<tbody>
<tr>
<td>GOLD</td>
<td>8.60</td>
<td>0.36</td>
<td>-</td>
</tr>
<tr>
<td>GLOBAL</td>
<td>7.73</td>
<td>2.02</td>
<td>***</td>
</tr>
<tr>
<td>DEPLEN-GLOBAL</td>
<td>8.35</td>
<td>0.75</td>
<td>**</td>
</tr>
<tr>
<td>DEPORD-NONF</td>
<td>7.98</td>
<td>1.15</td>
<td>***</td>
</tr>
<tr>
<td>DEPORD-NODIST</td>
<td>8.04</td>
<td>1.12</td>
<td>***</td>
</tr>
<tr>
<td>DEPLEN-NODIST</td>
<td>8.23</td>
<td>0.45</td>
<td>n.s.</td>
</tr>
<tr>
<td>DEPORD-NF</td>
<td>8.26</td>
<td>0.71</td>
<td>**</td>
</tr>
<tr>
<td>DEPLEN</td>
<td>8.36</td>
<td>0.51</td>
<td>n.s.</td>
</tr>
</tbody>
</table>

Table 3.9: Distribution of heavy unequal constituents (length difference > 5) in Section 00; 4692 gold cases considered and significance tested against the gold standard using a \( \chi^2 \)-square test

Table 3.10: Examples of realized output for models with and without the dependency Length feature (see Table 3.3 for model details)
3.5 Discussion

Our experiments show a consistent positive effect of the dependency length feature in improving BLEU scores and achieving a better match with the corpus distributions of dependency length and short/long constituent orders. The results in Table 3.9 are particularly encouraging, as they show that minimizing dependency length reduces the number of realizations in which a heavy constituent precedes a light one down to essentially the level of the corpus, thereby eliminating many realizations that can be expected to have egregious errors.

Table 3.10 shows examples of how the dependency length feature affects the output in comparison to a model with a rich set of discriminative syntactic and dependency ordering features, but no features directly targeting relative weight (see Table 3.3 for model details). In wsj_0015.7, the dependency length models produce an exact match, while the DEPORD model fails to shift the short temporal adverbial \textit{next year} next to the verb, leaving a confusingly repetitive \textit{this year next year} at the end of the sentence. In wsj_0020.1, the dependency length models produce a nearly exact match with just an equally acceptable inversion of \textit{closely watching}. By contrast, the DEPORD model mistakenly shifts the direct object \textit{South Korea, Taiwan and Saudia Arabia} to the end of the sentence where it is difficult to understand following two very long intervening phrases. In wsj_0021.8, all the models mysteriously put \textit{not} in front of the auxiliary and leave out the complementizer, but DEPORD also mistakenly leaves \textit{before} at the end of the verb phrase where it is again apt to be interpreted as modifying the preceding verb.

There is some evidence that a negatively weighted total dependency length feature can go too far in minimizing dependency length, in the absence of other informative features to counterbalance it. In particular, the DEPLEN-GLOBAL model in Table 3.7 has significantly
lower dependency length than the corpus, but in the richer models with discriminative syntactic and dependency ordering features, there are no significant differences. In our setup, the preference to minimize dependency length can be balanced by features capturing preferences for alternate choices (e.g., the argument-adjunct distinction in the dependency ordering model). Via distributional analyses, we show that while simpler realization ranking models can go overboard in minimizing dependency length, richer models largely succeed in overcoming this issue, while still taking advantage of dependency length minimization to avoid egregious ordering errors. As shown in wsj_0075.13 in Table 3.10, the final two adjuncts are in the long-short order (resulting in higher dependency length), but the models still prefer that ordering, going counter to the minimal dependency length choice.

However, it may still be though that additional features are necessary to counteract the tendency towards dependency length minimization, for example to ensure that initial constituents play their intended role in establishing and continuing topics in discourse. A case in point is wsj_0014.2 in the table where DEPORD results in a near exact match (except for a missing comma) but the dependency length models front the PP on the 12-member board, where it is grammatical but rather marked (and not motivated in the discourse context). We now elaborate on this issue in more detail by calling attention to attested “anti-dependency length” tendencies discussed in the literature. In the next section, we examine four specific instances discussed in the literature, where the drive to minimize dependency length does not result in acceptable constituent orders in language production.

### 3.5.1 Divergences from Canonical Word Order

Cases like the final example above point to the fact that dependency length is more of a preference than an optimization objective, which must be balanced against other order
preferences at times. Moreover, anti-locality effects have been discussed in the psycholinguistics literature on German (Konieczny, 2000) and Hindi (Vasishth and Lewis, 2006). Based on counter-evidence presented in the literature as well as examining examples from the output of surface realization, the following constructions involving “anti-dependency length” tendencies are discussed here:

1. Facts from adverb Placement (Gildea and Temperley, 2007)

2. Data from sentence initial pre-modifying adjuncts (Temperley, 2007)

3. Distribution of argument and adjuncts with respect to their verbal head (Hawkins, 2001)

4. Experimental evidence from non-projective dependencies (Park and Levy, 2009)

**Adverb Placement**

<table>
<thead>
<tr>
<th>Function Tag</th>
<th>%Short 2nd Constituents</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMP</td>
<td>42.4</td>
</tr>
<tr>
<td>CLR</td>
<td>12.2</td>
</tr>
<tr>
<td>LOC</td>
<td>11.2</td>
</tr>
<tr>
<td>PRP</td>
<td>4.4</td>
</tr>
<tr>
<td>ADV</td>
<td>4.4</td>
</tr>
<tr>
<td>DIR</td>
<td>4.4</td>
</tr>
<tr>
<td>MNR</td>
<td>3.05</td>
</tr>
</tbody>
</table>

Table 3.11: Distribution of the second PTB function tag in 295 long-short sequences

Gildea and Temperley (2007) suggest that adverb placement might involve cases which go against dependency length minimization. Pursuing this suggestion, 295 legitimate long-short post-verbal constituent orders (counter to dependency length) from Section 00 of the
Penn Treebank were examined. Table 3.11 shows the distribution of second constituent function tags in these sequences which counter dependency length minimization. The figures indicate that there is a predominant tendency for the shorter constituent to express temporal information. The following examples illustrate this pattern (temporal information constituents are underlined):

(49) The Treasury also said it plans to sell [$10 billion] [in 36-day cash management bills] [on Thursday]. (WSJ0075.13)

(50) Enterprise Rent-A-Car Inc. breaks [its first national ad campaign] [this week]. (WSJ0062.45)

Pre-modifying Adjuncts

![Pre-modifying Adjuncts](image)

Figure 3.6: Reproduced from Temperley (2007): Illustrating pre-modifying adjunct sequences

A closer look at Temperley’s original corpus study (Temperley, 2007), revealed counterexamples to dependency minimization. A case in point is the class of examples involving
pre-modifying adjunct sequences that precede both the subject and the verb (illustrated in Figure 3.6). Assuming that their parent head is the main verb of the sentence, a long-short sequence would minimize overall dependency length. However, in 613 examples found in the PTB, average length of the first adjunct was 3.15 words while the second adjunct was 3.48 words long, thus reflecting a short-long pattern. In the Brown Corpus, the length difference was more pronounced (first adjuncts being 2.44 words and the second one being 4.22 on an average). The following examples illustrate this (David Temperly p.c.):

(51)  [In 1976], [as a film student at the Purchase campus of the State University of New York], Mr. Lane, shot ...(WSJ0039.4)

(52)  As a film student at the Purchase campus of the State University of New York [in 1976], Mr. Lane, shot ...

(53)  [In Los Angeles], [in our lean years], we gave parties. (WSJ1367.31)

(54)  In our lean years , [in Los Angeles], we gave parties.

Informal native judgements revealed that examples (52) and (54) above which minimize dependency length, are less preferred compared to the original corpus sentences in (51) and (53). The “short-long” tendency (Arnold et al., 2000; Wasow, 2002) provides a better generalization to account for the above examples. In the sentence initial position, speakers might be overriding the tendency to minimize dependency length as a consequence of other considerations like foregrounding certain ideas.

Argument Placement

As discussed earlier, Hawkins (2001) demonstrates the tendency of complements to be located closer to verbal heads compared to adjuncts. Following that work, we did a preliminary investigation of the relationship between arguments, adjuncts and subjects and their
Table 3.12: Results of a preliminary study analyzing argument-adjunct patterns in PTB data using Propbank annotation

respective heads in Penn Treebank data. The complement-adjunct distinction was obtained from Propbank roles (Palmer et al., 2005), a set of manually annotated verbal semantic roles. Pre-verbal distances were calculated by counting the number of words separating the right edge of constituents to their verbal head. Post-verbal distances were calculated by counting the number words separating the head and the left edge of constituents. Table 3.12 illustrates some preliminary results. It can be seen that post-verbal arguments are closer to verbal heads compared to post-verbal adjuncts, confirming the patterns observed in Hawkins’ study. There were many cases where the drive to minimize dependency length was over-ridden by the tendency to place arguments closer to their verbal head than adjuncts (example (27) discussed earlier in Section 3.1.5 is one such instance).

**Non-Projective Dependencies**

An instance where dependency might not be the only factor in language production is the class of examples involving extraposed relative clauses (Park and Levy, 2009):

(55) Yesterday a woman who was wearing a hat arrived.

(56) Yesterday a woman arrived who was wearing a hat.

In the above examples (both of which are equally acceptable), (55) has higher dependency length compared to the extraposed relative clauses in (56), where the dependency

<table>
<thead>
<tr>
<th>Type</th>
<th>Pre-verbal Constituents</th>
<th>Post-verbal Constituents</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Freq</td>
<td>Mean Length</td>
</tr>
<tr>
<td>Adjunct</td>
<td>1613</td>
<td>2.37</td>
</tr>
<tr>
<td>Argument</td>
<td>1847</td>
<td>4.71</td>
</tr>
<tr>
<td>Subject</td>
<td>2507</td>
<td>3.30</td>
</tr>
</tbody>
</table>
between *Yesterday* and *arrived* crosses the dependency between *woman* and *who*. In recently submitted work (Levy et al., 2009), Roger Levy provides results from self-paced reading experiments that indicate that extraposed relative clauses are harder to process. Investigating the conditions affecting the processing of extraposed relative clauses, their experimental results show that when extraposition occurs from simple [determiner+noun] NPs across a verb, the resulting sentence is harder to process than the unextraposed variant. When extraposition from a direct object NP across a PP results in higher processing difficulty compared to the unextraposed relative clauses modifying either the direct object following the PP or the PP-internal NP. So the tendency to avoid non-projective (crossing) dependencies rather than minimize dependency length should also be entertained seriously as a factor influencing competing choices in language production.

### 3.6 Summary

This chapter introduced the theory of minimal dependency length by presenting evidence from the psycholinguistics and theoretical linguistics literature. The relationship between dependency length and other factors influencing language production was also examined. Further on, the efficacy of dependency length as a global feature in conjunction with other features in realization ranking was demonstrated by results from surface realization experiments. Subsequently, details of evaluation procedures involving BLEU scores, human judgements and a novel method using distributional analyses were presented. Finally constructions exhibiting “anti-dependency length” tendencies were discussed to provide a balanced view of dependency length minimization in language production.
CHAPTER 4: DESIGNING INFLECTION AND AGREEMENT FEATURES FOR REALIZATION RANKING

This chapter shows that incorporating linguistically motivated features to ensure correct inflection and agreement choices in an averaged perceptron ranking model for CCG realization helps improve a state-of-the-art baseline even further. The chapter is structured as follows. Section 4.1 motivates the need for a principled treatment of inflection and agreement choices in surface realization. Sections 4.3 and 4.2 describe the nature and complexity of agreement phenomena. Section 4.4 explains why balanced punctuation is a problem for CCG grammars. Section 4.5 describes the features we have designed for animacy and number agreement as well as for balanced punctuation. Section 4.6 presents our evaluation of the impact of these features in averaged perceptron realization ranking models, tabulating specific kinds of errors in the CCGbank development section. Finally Section 4.7 discusses examples from surface realization and compares the performance of the statistical agreement model with hand-crafted rules.

4.1 Motivation for Incorporating Inflection and Agreement Features in the Statistical Ranking Model

In recent years a variety of statistical models for realization ranking that take syntax into account have been proposed, including generative models (Bangalore and Rambow, 2000b; Cahill and van Genabith, 2006; Hogan et al., 2007; Guo et al., 2008), maximum entropy models (Nakanishi et al., 2005) and averaged perceptron models (White and Rajkumar,
Table 4.1: Section 00 OpenCCG inflection and agreement mismatches (from 847 complete, but non-exact derivations) in the output of White and Rajkumar (2009)

<table>
<thead>
<tr>
<th>Mismatch Type</th>
<th>#Mismatches</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relativizer</td>
<td>67</td>
</tr>
<tr>
<td>Agreement Errors</td>
<td>40</td>
</tr>
<tr>
<td>Unbalanced Punctuation</td>
<td>49</td>
</tr>
</tbody>
</table>

2009). To our knowledge, however, none of these models have included features specifically designed to handle grammatical agreement, an important task in surface realization. As discussed in the introduction, the input to the surface realizer could be underspecified to varying degrees. The logical forms used by the OpenCCG realizer contains lemmas as predicates in the HLDS graph and during generation, the realizer has to generate the correct word using rules of English inflectional morphology. In the case of verbal word forms, this also interacts with the number information of the subject, leading to subject-verb agreement considerations. Relativizers and punctuation marks are also not encoded in the OpenCCG input and the task of choosing the appropriate lexical item is again performed during surface realization. Thus the need to ensure animacy agreement and punctuation balancing is evident.

The starting point for our work is a CCG realization ranking model that incorporates Clark & Curran’s (2007b) normal-form syntactic model, developed for parsing, along with a variety of n-gram models. Although this syntactic model plays an important role in achieving top BLEU scores for a reversible, corpus-engineered grammar, an error analysis nevertheless revealed that many errors in relativizer animacy agreement and subject-verb number agreement remain with this model. Table 4.1 illustrates the frequency of inflection and agreement errors in the White and Rajkumar (2009) realization ranking model.
discussed in detail previously in Section 2.4. This along with the fact that human judges of surface realization output tended to harshly penalize sentences with number-agreement (noted in our evaluation efforts described in Espinosa et al., 2010) prompted us to investigate principled means of enforcing correct inflection and agreement choices.

Traditionally, grammatical agreement phenomena have been modelled using hard constraints in the grammar. Taking into consideration the range of acceptable variation in the case of animacy agreement and facts about the variety of factors contributing to number agreement, the question arises: tackle agreement through grammar engineering, or via a ranking model? In our experience, trying to add number and animacy agreement constraints to a grammar induced from the CCGbank (Hockenmaier and Steedman, 2007) turned out to be surprisingly difficult, as hard constraints often ended up breaking examples that were working without such constraints, due to exceptions, sub-regularities and acceptable variation in the data. With sufficient effort, it is conceivable that an approach incorporating hard agreement constraints could be refined to underspecify cases where variation is acceptable, but even so, one would want a ranking model to capture preferences in these cases, which might vary depending on genre, dialect or domain. Given that a ranking model is desirable in any event, we investigate here the extent to which agreement phenomena can be more robustly and simply handled using a ranking model alone, with no hard constraints in the grammar. More importantly, the heterogeneous and complex nature of agreement phenomena is well studied in the theoretical linguistics literature (Kathol, 1999; Pollard and Sag, 1994). Claims about the purely syntactic status of English agreement have been refuted by compelling evidence resulting from a long body of work on this theme. The next two sections will summarize the discussion about the nature of agreement in the literature. The decision to adopt a statistical approach is based on the insights derived from our perusal of
the literature as a result of difficulties encountered while writing grammar rules for number and animacy agreement. Subsequently, we show that incorporating linguistically motivated features to ensure correct animacy and verbal agreement in an averaged perceptron ranking model for CCG realization helps improve a state-of-the-art baseline even further. Compared to writing grammar rules, our method is more robust and allows incorporating information from diverse sources in realization.

We also demonstrate the utility of such an approach in ensuring the correct presentation of balanced punctuation marks. The function of punctuation marks in written language is to help people process text more easily. In computational systems, punctuation provides disambiguation cues which can help parsers arrive at the correct parse. Doran (1998) uses the following example to illustrate this:

\[(57)\]

a. Let’s eat Grandfather before we go.

b. Let’s eat, Grandfather, before we go.

In mainstream systems, punctuation is sometimes stripped off and grammar rules are designed keeping in mind only the bare text. But, in the examples given above, when the punctuation is stripped off, most parsers would get only the transitive case as the vocative use of Grandfather would go unrecognized, unless the grammar has been specially designed for that. The lack of punctuation rules in the grammar is also problematic from the standpoint of natural language generation as text without punctuation can be difficult to comprehend. Hence, integrating an analysis of punctuation into parsing/generation grammars is necessary. Working with such a bi-directional grammar, we further show here that the perceptron model can reduce balanced punctuation errors that would otherwise require a post-hoc filter over entire derivations in order to rule out infelicitous sequences like balanced commas at the end of sentences. As White and Rajkumar (2008) discuss, in CCG
it is not feasible to use features in the grammar to ensure that balanced punctuation (e.g., paired commas for NP appositives) is used in all and only the appropriate places, given the word-order flexibility that crossing composition allows. While a post-filter is a reasonably effective solution, it can be prone to search errors and does not allow balanced punctuation choices to interact with other choices made by the ranking model. Section 4.4 provides a detailed discussion of this theme using CCG syntactic analyses, before moving on to describe our feature design and finally the impact of these features on realization performance.

4.2 The Heterogeneous Nature of Number Agreement

Subject-verb agreement can be described as a constraint where the verb agrees with the subject in terms of agreement features (number and person). Agreement has often been considered to be a syntactic phenomenon and grammar implementations generally use syntactic features to enforce agreement constraints (e.g., Velldal and Oepen, 2005). However a closer look at our data and a survey of the theoretical linguistics literature points toward a more heterogeneous conception of English agreement. To elaborate further, the nature of agreement is influenced by syntactic, semantic and pragmatic factors and a unitary conception of agreement as merely one of these cannot explain the various patterns in the data. The following sections discuss examples from the relevant literature to illustrate the complex nature of English agreement and also present critiques of existing proposals. The discussion is based on examples from a variety of sources compiled as part of a survey of agreement proposals in the literature discussed in (Kim, 2004).
4.2.1 Problems with a Purely Syntactic View

The paradigm below involving numeral determiners illustrates the issues involved in casting English agreement as a purely syntactic phenomenon (data from Reid, 1991 and Hudson, 1999):

(58) Five miles is/*are a long distance to walk.
(59) Five pounds is/*are a lot of money.
(60) Two drops deodorizes/*deodorize anything in your house.
(61) Fifteen dollars in a week is/*are not much.
(62) Fifteen years represents/*represent a long period of his life.
(63) Two miles is/*are as far as they can walk.

In the preceding examples, the grammatical sentences involve numeral determiners and subjects which are plural, while the verb is singular. The following classes of examples also illustrate why a purely syntactic account of agreement is problematic.

1. Conjoined NP has a single referent

(64) This bomber and its cargo probably weighs over a hundred tons. (Corbett 1994)
(65) John and only John is allowed in here. (Biber et al., 1999)

In both the examples mentioned above, the conjoined NP has a single referent in terms of semantics, so the verb is in the singular form. In order to distinguish such cases from regular conjunct subjects, a constraint-based grammar implementation would need special features based on semantic information.

2. Reference transfer
(66) The hash browns at table nine are/is getting cold.

(67) The hash browns at table nine is/are getting angry. (Nunberg, 1995)

(68) King prawns cooked in chili salt and pepper was very much better, a simple dish succulently executed. (Biber et al., 1999)

In (66), the verb agrees with the plural subject *hash browns*. However (67) is appropriate in a conversation involving two waiters where one of them is referring to a customer who has ordered hash browns at the restaurant. Thus the reference of another discourse entity has changed the plural identity of the subject and the verb also encodes that change. For an example which does not require positing context information, (68) is a case where the singular verb agrees with the dish, rather than with individual prawns.

3. **Measure nouns**

   (69) “I think it will shake confidence one more time, and a lot of this business is based on client confidence.” (WSJ1866.10)

   (70) It’s interesting to find that a lot of the expensive wines aren’t always walking out the door. (WSJ0071.53)

Measure nouns such as *lot* and *ton* exhibit singular agreement with the determiner *a*, but varying agreement with the verb depending on the head noun of the measure noun’s *of*-complement.

4. **Morphosyntactic agreement features of the subject NP**

   (71) Two million dollars comes from corporations and foundations, but almost $400,000 from private gifts (from New York Times quoted by Reid, 1991)
The professional ethics arises from the requirement that analysis be unbiased. (Biber et al., 1999)

In the above examples, the subjects, viz. dollars and ethics exhibit plural morphology, but the verb is singular.

5. Collective nouns

The government are planning new tax increases.

The faculty are all agreed on this point.

The collective noun examples (from Kim, 2004) illustrate plural verbs in cases where it is possible to conceive the collective nouns to denote individual members of the group. This pattern is not predictable if only the morpho-syntactic properties of nouns are considered. As is also well known, British and American English differ in subject-verb agreement with collective nouns.

4.2.2 Problems with a Purely Semantic View

In the light of the evidence against a purely syntactic account of agreement, Kathol (1999) proposes an explanation where agreement is determined by the semantic properties of the noun rather than by its morphological properties. This accounts for most of the cases above. In the light of this explanation, specifying agreement features in the logical form for realization could perhaps solve the problem. However, the semantic view of agreement is not completely convincing due to the following classes of counter-examples discussed in the literature.

1. Intended referent of the subject
(75) I am/*is parked on the hill.

(76) You need/*needs a help from the one that can do this job. (Nunberg 1995)

In the first example, the subject I refers to a car, but the verb am displays syntactic agreement with the features of the subject. The second example is merely illustrating the fact that though the pronoun you can refer to a single individual or a group of individuals, the verb exhibits plural morphology (i.e., agreement with the plural feature in the subject).

2. Collective nouns

The semantic view of collective nouns allows them to assume both singular and plural identities and appear with both singular and plural verbs. The following example from (Pollard and Sag, 1994) involving change in the individuation mode of the collective noun illustrates a situation where the collective noun senate acquires both singular and plural features.

(77) The Senate just voted itself another raise. Most of them were already overpaid to begin with.

However they also point out the existence of cases where collective nouns do not always have the freedom to assume their own mode of individuation. In the example below, the agreement facts related to family illustrate this.

(78) His family are/*is all overweight.

(79) His family is/*are moving to Seoul.

3. Pronoun-Antecedent agreement

*I thank Bob Levine for pointing me to the remaining examples in this section*
4. Suppose you meet someone and they are totally full of themselves (Bender and Flickinger, 1999)

In (4), the pronoun they used in a generic sense is linked to the singular antecedent someone, but its plural feature triggers plural agreement with the verb.

5. Nonsemantic plural nominals

(80) Those scissors/glasses/trousers are missing.

(80) illustrates a situation where the subject scissors is arguably semantically singular, but exhibits plural morphology and plural syntactic agreement with both the determiner as well as the verb.

6. Royal We

(81) We are not amused. (attributed to Queen Victoria)

(82) We have become a grandmother. (Margaret Thatcher)

As is obvious, even when the semantics of the plural subject we denotes a single individual, the verb exhibits plural morphology.

Thus the preceding discussion suggests that English has a set of heterogeneous agreement patterns rather than purely syntactic or semantic ones. Our survey of the literature indicates that agreement phenomena are modelled best in a framework where the morphology tightly interacts with the system of syntax, semantics, or even pragmatics. To this end, our machine learning-based approach approximates the insights discussed in the theoretical linguistics literature. Writing grammar rules to get these facts right proved to be surprisingly difficult. One of the major problems we encountered was discerning the actual
nominal head contributing agreement feature in cases like areas of the factory were/was vs. a lot of wines are/is. This required a list of measure nouns and partitive quantifiers. Part of the difficulty in writing rules was because of the fact that the Penn Treebank does not annotate mass/count distinctions explicitly precipitating the need for heuristics to infer that. These factors led us to investigate the extent to which a machine learning–based approach is a simpler, practical alternative for acquiring the relevant generalizations from the data by combining information from various information sources.

4.3 The Nature of Animacy Agreement

To illustrate the variation that can be found with animacy agreement phenomena, consider first animacy agreement connected to relativizers. In English, an inanimate noun can be modified by a relative clause introduced by that or which, while an animate noun combines with who(m). With some nouns though — such as team, group and squad — animacy status is uncertain, and these can be found with all the three relativizers (who, which and that). Google counts suggest that all three choices are almost equally acceptable, as the examples below illustrate:

(83) The groups who protested against plans to remove asbestos from the nuclear submarine base at Faslane claimed victory when it was announced the government intends to dispose of the waste on site. (The Glasgow Herald; Jun 25, 2010)

(84) Mr. Dorsch says the HIAA is working on a proposal to establish a privately funded re-insurance mechanism to help cover small groups that can’t get insurance without excluding certain employees. (WSJ0518.35)

(85) Michael R. Dabney, 44, a managing director who directs the principal activities group which provides funding for leveraged acquisitions (WSJ0241.1)
4.4 CCG and Balanced Punctuation

A complex issue that arises in the design of bi-directional grammars is ensuring the proper presentation of punctuation. Among other things, this involves constraining over-generation of punctuation marks. One such instance where this is crucial is the correct realization of commas introducing noun phrase appositives. Appositives can occur sentence medially or finally. The conventions of writing mandate that sentence medial appositives should be balanced—i.e., the appositive NP should be surrounded by commas or dashes on both sides—while sentence final appositives should be unbalanced—i.e., they should only have one preceding comma or dash. The categories and semantics for unbalanced and balanced appositive commas in our CCG grammar are, respectively:

\[
\begin{align*}
\text{(86) a. } & , \vdash \text{np}_{(1)} / \text{np}_{(1)} / \text{np}_{(3)} : @X_1((\text{APPOSREL} \land X3)) \\
\text{b. } & , \vdash \text{np}_{(1)} / \text{np}_{(1)} / \text{punct}[,] / \text{np}_{(3)} : @X_1((\text{APPOSREL} \land X3))
\end{align*}
\]

Here, the unbalanced appositive has a category where the comma selects as argument the appositive NP and converts it to a nominal modifier. For balanced appositives, the comma selects the appositive NP and the balancing comma to form a nominal modifier. So choosing when to use category (86a) vs. (86b) is an important lexical choice decision in surface realization. The paradigm below helps illustrate the issues:

\[
\begin{align*}
\text{(87) } & \text{John, CEO of ABC, loves Mary.} \\
\text{(88) } & \text{* John, CEO of ABC loves Mary.} \\
\text{(89) } & \text{Mary loves John, CEO of ABC.}
\end{align*}
\]

\[21\]I thank Michael White for alerting me to this class of examples while introducing punctuation analyses into the CCGbank.
Now in the next two sections, we present two standard approaches which have been proposed in the literature to constrain the over-generation of balanced punctuation marks and evaluate their suitability for bi-directional grammars. We show that both these approaches are not well-suited for CCG grammar implementations and hence advocate an approach involving features in the ranking model.

4.4.1 Absorption vs. Syntactic Features

Nunberg (1990) argues that text adjuncts introduced by punctuation marks have an underlying representation where these adjuncts have marks on either side. They attain their surface form when a set of presentation rules are applied. This approach ensures that all sentence medial cases like (87) and (91) above are generated correctly, while unacceptable examples (88) and (92) would not be generated at all. (89) would at first be generated as (90): to deal with such sentences, where two points happen to coincide, Nunberg posits an implicit point which is absorbed by the adjacent point. Absorption occurs according to the “strength” of the two points. Strength is determined according to the Point Absorption Hierarchy, which ranks commas lower than dashes, semi-colons, colons and periods. As White (1995) observes, from a generation-only perspective, it makes sense to generate text adjuncts which are always balanced and post-process the output to delete lower ranked points, as absorption uses relatively simple rules that operate independently of the hierarchy of the constituents. However, using this approach for parsing would involve a pre-processing step which inserts commas into possible edges of possible constituents, as
described in (Forst and Kaplan, 2006). To avoid this considerable complication, Briscoe (1994) has argued for developing declarative approaches involving syntactic features, with no deletions or insertions of punctuation marks.

4.4.2 Features for Punctuation in CCG?

Unfortunately, the feature-based approach appears to be inadequate for dealing with the class of examples presented above in CCG. This approach involves the incorporation of syntactic features for punctuation into atomic categories so that certain combinations are blocked. To ensure proper appositive balancing sentence finally, the rightmost element in the sentence should transmit a relevant feature to the clause level, which the sentence-final period can then check for the presence of right-edge punctuation. Possible categories for a transitive verb and the full stop appear below:

\[(93) \text{loves} \vdash s(1)_{bal=\text{BAL}, end=\text{PE}} \setminus \text{np}(2)_{bal=+, end=\text{PE}} \setminus \text{np}(3)_{bal=\text{BAL}, end=\text{PE}}\]

\[(94) . \vdash \text{sent} \setminus s_{end=\text{nil}}\]

Here the feature variables \text{BAL} and \text{PE} of the rightmost argument of the verb would unify with the corresponding result category feature values to realize the main clauses of (89) and (90) with the following feature values:

\[(95) \text{Mary loves John, CEO of ABC} \vdash s(1)_{bal=\text{-, end=\text{nil}}}\]

\[(96) \text{Mary loves John, CEO of ABC,} \vdash s(1)_{bal=+, end=\text{comma}}\]

Thus, in (96), the sentence-final period would not combine with \(s(1)_{bal=+, end=\text{comma}}\) and the derivation would be blocked.\(^{22}\)

\(^{22}\)It is worth noting than an n-gram scorer would highly disprefer example (90), as a comma period sequence would not be attested in the training data. However, an n-gram model cannot be relied upon to eliminate examples like (92), which would likely be favored as they are shorter than their balanced counterparts.
**Issue 1: Extraction cases**

The solution sketched above is not adequate to deal with extraction involving di-transitive verbs in cases like (97) and (98):

(97) Mary loves a book that John gave Bill, his brother.

(98) * Mary loves a book that John gave Bill, his brother.

As Figure 4.1 shows, an unacceptable case like (98) is not blocked. Even when the sentence final NP is balanced, the end=comma value is not propagated to the root level. This is because the end feature for the relative clause should depend on the first (indirect) object of gave, rather than the second (direct) object as in a full di-transitive clause. A possible solution would be to introduce more features which record the presence of punctuation in the leftward and rightward arguments of complex categories and subsequently percolated to the root to be checked off against the full stop. However, this solution will still not solve the problem discussed in the next section.

**Issue 2: Crossing composition**

Another issue is how crossing composition, used with adverbs in heavy NP shift constructions, interacts with appositives, as in the following examples:
(99) Mary loves madly John, CEO of ABC.

(100) * Mary loves madly John, CEO of ABC.,

For examples (91) and (92), which do not involve crossing composition, the category for the adverb should be the one in 101:

(101) $\text{madly} \vdash s_{(1)} \text{end}=\text{nil} \setminus \text{np}_{(2)} \setminus (s_{(1)} \text{bal}=\text{+} \text{np}_{(2)})$

Here the $\text{bal}=\text{+}$ feature on the argument of the adverb $\text{madly}$ ensures that the direct object of the verb is balanced, as in (91); otherwise, the derivation fails, as in (92). Irrespective of the value of the $\text{end}$ feature of the argument, the result of the adverb has the feature $\text{end}=\text{nil}$ as the post-modifier is lexical material which occurs after the VP. With crossing composition, however, category (101) would licence an erroneous derivation for example (100), as the $\text{end}=\text{nil}$ feature on the result of the adverb category would prevent the percolation of the $\text{end}$ feature at the edge of the phrase to the clausal root, as Figure 4.2 shows.

\[
\begin{array}{cccccc}
\text{Mary} & \text{loves} & \text{madly} & \text{John, CEO,} & . \\
\text{np} & \text{s}_{\text{end}=\text{PE}} \setminus \text{np/np}_{\text{end}=\text{PE}} & \text{s}_{(1)} \text{end}=\text{nil} \setminus \text{np}_{(1)} \setminus (s_{(1)} \text{bal}=\text{+} \text{np}_{(2)}) & \text{np}_{\text{bal}=\text{+}, \text{end}=\text{comma}} & \text{sent} \\
\hline
\text{s}_{\text{end}=\text{nil}} \setminus \text{np/np}_{\text{end}=\text{PE}} & \text{s}_{(1)} \text{end}=\text{nil} \setminus \text{np}_{(1)} \setminus (s_{(1)} \text{bal}=\text{+} \text{np}_{(2)}) & \text{sent} \\
\end{array}
\]

Figure 4.2: Crossing composition in CCG

To block such derivations, one might consider giving the adverb another category for use with crossing composition:

(102) $\text{madly} \vdash s_{(1)} \setminus \text{np}_{(2)} \setminus (s_{(1)} \setminus \text{np}_{(2)})$
The use of the non-associative, permutative modality \( \times \) on the main slash allows the crossing composition rule to be applied, and feature inheritance ensures that the end feature from the verb *loves* is also copied over. Thus, in example (100), the punctuation at the edge of the phrase would be percolated to the clausal root, where the sentence-final period would block the derivation. However, in the slash modality inheritance hierarchy proposed by Baldridge (2002), the non-associative, permutative modality inherits the properties of function application. Consequently, this category could also lead to the erroneous derivation of example (92). In such a derivation, category 102 will not require the direct object to have a balanced appositive; meanwhile, the \textit{end=nil} feature on the direct object will propagate to the clausal root, where it will combine with the category for the full stop. Finally, having two distinct categories for the adverb would offset the advantage of multi-modal categorial grammar in dealing with word order variation, where it is possible to use one category in situations where otherwise several categories would be required.

Thus the preceding discussion demonstrates that the seemingly simple task of correct punctuation balancing cannot be performed by features in CCG grammars. Punctuation balancing participates in a range of other word order choices and ultimately in interactions involving the entire grammar. This merits a more satisfactory grammatical treatment involving constraints in independent orthographic derivations, perhaps along the lines of the autonomous prosodic derivations which Steedman and Prevost (1994) discuss. However, more in line with our current realization ranking set-up, we investigate a statistical approach to model punctuation balancing. Apart from its simplicity, this approach has the advantage that the feature weights learned by the ranking model would weigh punctuation balancing with other choices involving word order and lexical category selection. So
we now discuss the stochastic ranking features designed to model agreement and balanced punctuation choices.

4.5 Feature Design

White and Rajkumar’s (2009) realization ranking model discussed earlier serves as the baseline for this chapter. It is a global, averaged perceptron ranking model using three kinds of features: (1) the log probability of the candidate realization’s word sequence according to three linearly interpolated language models (as well as a feature for each component model), much as in the log-linear models of Velldal and Oepen (2005) and Nakanishi et al. (2005); (2) integer-valued syntactic features, representing counts of occurrences in a derivation, from Clark & Curran’s (2007b) normal form model; and (3) discriminative $n$-gram features (Roark et al., 2004), which count the occurrences of each $n$-gram in the word sequence. Table 4.2 depicts the new animacy, agreement and punctuation features being introduced as part of this chapter. These features make use of existing corpus annotations — specifically, PTB function tags and BBN named entity classes (Weischedel and Brunstein, 2005) — and thus they are relatively easy to implement. The next two sections describe these features in more detail.

4.5.1 Animacy and Number Agreement Features

Underspecification as to the choice of relativizer in the input leads to competing realizations involving the relativizers who, that, and which. The existing ranking models ($n$-gram models as well as perceptron) often allow the top-ranked output to have the relativizer that associated with animate nouns. The existing normal form model uses the word forms as well as part-of-speech tag based features. Though this is useful for associating proper nouns (tagged NNP or NNPS) with who, for other nouns (as in consumers who vs.
### Table 4.2: New features introduced

<table>
<thead>
<tr>
<th>Feature</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Animacy features</strong></td>
<td></td>
</tr>
<tr>
<td>Noun Stem + Wh-pronoun</td>
<td>researcher + who</td>
</tr>
<tr>
<td>Noun Class + Wh-pronoun</td>
<td>PER_DESC + who</td>
</tr>
<tr>
<td><strong>Number features</strong></td>
<td></td>
</tr>
<tr>
<td>Noun + Verb</td>
<td>people + are</td>
</tr>
<tr>
<td>NounPOS + Verb</td>
<td>NNS + are</td>
</tr>
<tr>
<td>Noun + VerbPOS</td>
<td>people + VBP</td>
</tr>
<tr>
<td>NounPOS + VerbPOS</td>
<td>NNS + VBP</td>
</tr>
<tr>
<td>Noun_of + Verb</td>
<td>lot_of + are</td>
</tr>
<tr>
<td>Noun_of + VerbPOS</td>
<td>lot_of + VBP</td>
</tr>
<tr>
<td>NounPOS_of + Verb</td>
<td>NN_of + are</td>
</tr>
<tr>
<td>NounPOS_of + VerbPOS</td>
<td>NN_of + VBP</td>
</tr>
<tr>
<td>Noun_of + of-complementPOS + VerbPOS</td>
<td>lot_of + NN + VBZ</td>
</tr>
<tr>
<td>NounPOS_of + of-complementPOS + VerbPOS</td>
<td>NN_of + NN + VBZ</td>
</tr>
<tr>
<td>Noun_of + of-complementPOS + Verb</td>
<td>lot_of + NN + is</td>
</tr>
<tr>
<td>NounPOS_of + of-complementPOS + Verb</td>
<td>NN_of + NN + is</td>
</tr>
<tr>
<td><strong>Punctuation feature</strong></td>
<td></td>
</tr>
<tr>
<td>Balanced Punctuation Indicator</td>
<td>$unbalPunct$</td>
</tr>
</tbody>
</table>

consumers that/which), the model often prefers the infelicitous relativizer. So here we designed features which also took into account the named entity class of the head noun as well as the stem of the head noun. These features aid the discriminative n-gram features (PERSON, which has high negative weight). As the results section discusses, NE classes like PER_DESC contribute substantially towards animacy preferences.

For number agreement, we designed three classes of features (c.f. Number Agr row in Table 4.2). Each of these classes results in 4 features. During feature extraction, subjects of the verbs tagged VBZ and VBP and verbs was, were were identified using the PTB NP-SBJ function tag annotation projected on to the appropriate arguments of lexical categories of verbs. The first class of features encoded all possible combinations of subject-verb
word forms and parts of speech tags. In the case of NPs involving of-complements like *a lot of ...* (Examples 69 and 70), feature classes 2 and 3 were extracted (class 1 was excluded). Class 2 features encode the fact that the syntactic head has an associated of-complement, while class 3 features also include the part of speech tag of the complement. In the case of conjunct and disjunct VPs, each verb was matched with the subject and word and POS features were extracted. In the case of conjunct and disjunct subjects, features were extracted from individual subjects. The part-of-speech tag as well as word form of the conjunction and disjunction were also included. The motivation behind such a design was to glean syntactic and semantic generalizations from the data. During feature extraction, from each derivation, counts of animacy and agreement features were obtained.

4.5.2 Balanced Punctuation Feature

Section 4.4 presented a complex issue that arises in the design of bi-directional grammars in the context of ensuring the proper presentation of punctuation. Detailed arguments as to why syntactic features are not well-suited for CCG grammars are also presented there. White and Rajkumar (2008) solves this problem by means of a post-hoc filter over derivations in order to rule infelicitous balanced commas at the end of sentences. This approach involves the incorporation of syntactic features for punctuation into atomic categories so that certain combinations are blocked. To ensure proper appositive balancing sentence finally, the rightmost element in the sentence should transmit a relevant feature to the clause level, which the sentence-final period can then check for the presence of right-edge punctuation. However, the feature schema does not constrain cases of balanced punctuation in cases involving crossing composition and extraction. So in this chapter we explore a statistical approach to ensure proper balancing of NP apposition commas. The first step in
this solution is the introduction of a feature in the grammar which indicates balanced vs. unbalanced marks. We modified the result categories of unbalanced appositive commas and dashes to include a feature marking unbalanced punctuation, as follows:

\[(103) \quad , \vdash np_{\text{unbal}=\text{comma}} \backslash *, np_{1} / *, np_{2}\]

Then, during feature extraction, derivations were examined to detect categories such as \(np_{\text{unbal}=\text{comma}}\), and checked to make sure this NP is followed by another punctuation mark in the string such as a full stop. The feature counts the number of sentence-medial unbalanced punctuation marks in the derivation.

4.6 Evaluation

4.6.1 Experimental Conditions

For the experiments reported below, we used a lexico-grammar extracted from Sections 02–21 of our enhanced CCGbank with collapsed NEs, a hypertagging model incorporating named entity class features, and a trigram factored language model over words, named entity classes, part-of-speech tags and supertags. Perceptron training events were generated for each training section separately. The hypertagger and POS/supertag language model were trained on all the training sections, while separate word-based models were trained excluding each of the training sections in turn. Event files for 26530 training sentences with complete realizations were generated, with an average n-best list size of 18.2. The complete set of models is listed in Table 4.3.

4.6.2 Results

Realization results on the development and test sections are given in Table 4.4. For the development section, in terms of both exact matches and BLEU scores, the model with all
Table 4.3: Legend for experimental conditions

<table>
<thead>
<tr>
<th>Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>full-model</td>
<td>All the feats from models below</td>
</tr>
<tr>
<td>agr-punct</td>
<td>Baseline Feats + Punct + Num-Agr</td>
</tr>
<tr>
<td>wh-punct</td>
<td>Baseline Feats + Punct + Animacy-Agr</td>
</tr>
<tr>
<td>baseline-punct</td>
<td>Baseline Feats + Punct</td>
</tr>
<tr>
<td>baseline</td>
<td>Log prob + n-gram + Syntactic features</td>
</tr>
</tbody>
</table>

Table 4.4: Results (98.9% coverage)—percentage of exact match and grammatically complete realizations and BLEU scores

<table>
<thead>
<tr>
<th>Section</th>
<th>Model</th>
<th>%Exact</th>
<th>%Compl.</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>baseline</td>
<td>38.18</td>
<td>82.47</td>
<td>0.8341</td>
</tr>
<tr>
<td></td>
<td>baseline-punct</td>
<td>37.97</td>
<td>82.47</td>
<td>0.8340</td>
</tr>
<tr>
<td></td>
<td>wh-punct</td>
<td>38.93</td>
<td>82.53</td>
<td>0.8360</td>
</tr>
<tr>
<td></td>
<td>full-model</td>
<td>40.47</td>
<td>82.53</td>
<td>0.8403</td>
</tr>
<tr>
<td></td>
<td>agr-punct</td>
<td>40.84</td>
<td>82.53</td>
<td>0.8414</td>
</tr>
<tr>
<td>23</td>
<td>baseline</td>
<td>38.98</td>
<td>83.39</td>
<td>0.8442</td>
</tr>
<tr>
<td></td>
<td>full-model</td>
<td>40.09</td>
<td>83.35</td>
<td>0.8446</td>
</tr>
</tbody>
</table>

the three features discussed above (agreement, animacy and punctuation) performs better than the baseline which does not have any of these features.\textsuperscript{23} Also note that our baseline results differ slightly from the corresponding results reported in White and Rajkumar (2009) in spite of using the same feature set because quotes were introduced into the corpus on which these experiments were conducted. Previous results were based on the original CCGbank text where quotation marks are absent. However, using these criteria, the best performing model is actually the model which has agreement and punctuation features. The

\textsuperscript{23}In Rajkumar and White (2010), we reported statistical significance using bootstrap random sampling by running scripts available at \texttt{http://projectile.sv.cmu.edu/research/public/tools/bootStrap/tutorial.htm}. However, we would like to retract these results since these scripts were found to be faulty recently.
Table 4.5: Section 00 METEOR and TERP scores

<table>
<thead>
<tr>
<th>Model</th>
<th>METEOR</th>
<th>TERP</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>0.9819</td>
<td>0.0939</td>
</tr>
<tr>
<td>baseline-punct</td>
<td>0.9819</td>
<td>0.0939</td>
</tr>
<tr>
<td>wh-punct</td>
<td>0.9827</td>
<td>0.0923</td>
</tr>
<tr>
<td>agr-punct</td>
<td>0.9821</td>
<td>0.0902</td>
</tr>
<tr>
<td>full-model</td>
<td>0.9826</td>
<td>0.0909</td>
</tr>
</tbody>
</table>

model containing all the features does better than the punctuation-feature only model, but performs slightly worse than the agreement-punctuation model. Section 23, the test section, confirms that the model with all the features performs better than the baseline model. However, exact matches and BLEU scores do not necessarily reflect the extent to which important grammatical flaws have been reduced. So to judge the effectiveness of the new features, we computed the percentage of errors of each type that were present in the best Section 00 realization selected by each of these models.

Table 4.6: Error analysis of Section 00 complete realizations (total of 1554 agreement cases; total of 207 relativizer cases)

<table>
<thead>
<tr>
<th>Model</th>
<th>#Punct-Errs</th>
<th>%Agr-Errs</th>
<th>%WH-Errs</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>39</td>
<td>11.05</td>
<td>22.44</td>
</tr>
<tr>
<td>baseline-punct</td>
<td>0</td>
<td>10.79</td>
<td>20.77</td>
</tr>
<tr>
<td>wh-punct</td>
<td>11</td>
<td>10.87</td>
<td>13.53</td>
</tr>
<tr>
<td>agr-punct</td>
<td>8</td>
<td>4.0</td>
<td>21.84</td>
</tr>
<tr>
<td>full-model</td>
<td>10</td>
<td>4.31</td>
<td>15.53</td>
</tr>
</tbody>
</table>

Table 4.6 reports results of the error analysis. It can be seen that the punctuation-feature is effective in reducing the number of sentences with unbalanced punctuation marks.
Similarly, the full model has fewer animacy mismatches and just about the same number of errors of the other two types, though it performs slightly worse than the agreement-only model in terms of BLEU scores and exact matches. We also manually examined the remaining cases of animacy agreement errors in the output of the full model here. Of the remaining 18 errors, 14 were acceptable paraphrases involving object relative clauses (e.g. WSJ0083.40 ... the business that\(\) a company can generate). We also provide METEOR and TERP scores for these models (Table 4.5). In recently completed work on the creation of a human-rated paraphrase corpus to evaluate NLG systems (Espinosa et al., 2010), our analyses showed that BLEU, METEOR and TERP scores correlate moderately with human judgments of adequacy and fluency, and that the most reliable system-level comparisons can be made only by looking at all three metrics.

4.7 Discussion

We now turn to a qualitative examination of the output of the surface realization models described above. Table 4.7 presents four examples where the full model differs from the baseline. Example WSJ0003.8 illustrates an example where the NE tag PER_DESC for researchers helps the perceptron model enforce the correct animacy agreement, while the two baseline models prefer the that realization. Example WSJ0003.18 illustrates an instance of simple subject-verb agreement being enforced by the models containing the agreement features. Example WSJ0070.4 presents a more complex situation where a single subject has to agree with both verbs in a conjoined verb phrase. The last example in Table 4.7 shows the case of a NP subject which is a disjunction of two individual NPs. In both these cases, while the baseline models do not enforce the correct choice, the models with the agreement features do get this right. This is because our agreement features are sensitive to the
neither Lorillard nor the researchers who studied the workers were aware of any research on smokers of the Kent cigarettes.

neither Lorillard nor the researchers that studied the workers were aware of any research on smokers of the Kent cigarettes.

either Lorillard nor the researchers that studied the workers were aware of any research on smokers of the Kent cigarettes.

the plant, which is owned by Hollingsworth & Vose Co., was under contract with lorillard to make the cigarette filters.

the plant, which is owned by Hollingsworth & Vose Co., were under contract with lorillard to make the cigarette filters.

while many of the risks were anticipated when minneapolis-based Cray Research first announced the spinoff ...

while many of the risks was anticipated when minneapolis-based Cray Research first announced the spinoff ...
Table 4.8: Error rates of hand-crafted subject-verb agreement rules of Section 00 gold standard derivations corresponding to 1554 complete realizations in Table 4.6

<table>
<thead>
<tr>
<th>Rule set</th>
<th>%Agr-Errs</th>
</tr>
</thead>
<tbody>
<tr>
<td>POS tags</td>
<td>15.56</td>
</tr>
<tr>
<td>POS+Words</td>
<td>5.03</td>
</tr>
<tr>
<td>agr-punct model</td>
<td>4.0</td>
</tr>
</tbody>
</table>

subject-verb agreement. Both subject and verb are both assumed to be singular by default and subsequently the number information of the subject and the verb is determined individually. Finally, if they match and that matches the actual data, it is counted as a success. Simple rules were based on POS tags and more complex rules involving wordform heuristics were used. The rules are described below.

**POS rules**

1. If subject has part-of-speech JJ, DT, NNS, NNPS or CC, or subject has wordform comma or semi-colon then agreement feature is plural

2. If the verb has POS tag VBP, verb has plural value for the agreement feature

**Word + numeral rules**

1. If subject has wordform comma or semi-colon then agreement feature is plural

2. If word form of the subject is I, You, We or They, then agreement feature is plural

3. If subject has part-of-speech tag CD (and word form is not one or I) or the word contains % sign (to deal with the plural 1%), then agreement feature is plural

4. If word form of the verb equals do, have, were, then the agreement feature for the verb is plural
The results of rule application are shown in Table 4.8. Rules to infer the number of the head of the subject noun based only on POS tags were not very effective resulting in 15.56% errors. This was significantly lower than the performance of the agreement model according to McNemar’s $\chi^2$ test ($p = 0.005$, two-tailed). Augmenting POS rules with wordform based heuristics to discern the number information of numerals, pronouns and percentage entities boosted the performance of the rules. Though in this case the performance difference was not significantly different (according to McNemar’s $\chi^2$ test), the accuracy of the augmented rules was lower than the performance of the agreement model. It should also be noted that the wordform heuristics were finalized with considerable effort and after refining them with multiple rounds of error analyses after applying them to the development section data.

### 4.8 Summary

This chapter looked at the problem of inflection and agreement choices in surface realization. Traditionally, these choices have been modelled using hard constraints in the grammar. However, given the acceptable variation found in the case of animacy agreement we argue a case for the use of a statistical model to rank competing preferences. Though subject-verb agreement is generally viewed to be syntactic in nature, a perusal of relevant examples discussed in the theoretical linguistics literature points toward the heterogeneous nature of English agreement. The complexity of ensuring punctuation balancing in bi-directional CCG grammars necessitate a statistical treatment of constraining punctuation over-generation. Compared to writing grammar rules, our method is more robust and allows incorporating information from diverse sources in realization. We also show that the
perceptron model can reduce balanced punctuation errors that would otherwise require a post-filter. The full model makes many fewer inflection and agreement errors.
CHAPTER 5: LINGUISTICALLY MOTIVATED
COMPLEMENTIZER CHOICE IN SURFACE REALIZATION

This chapter deals with the task of optional function word insertion in surface realization. We show that using linguistically motivated features for English that-complementizer choice in an averaged perceptron model for classification can improve upon the prediction accuracy of a state-of-the-art realization ranking model. Section 5.1 introduces the importance of complementizer choice in written text. Section 5.2 summarizes Florian Jaeger’s approach towards modelling complementizer selection. Section 5.3 describes the features we designed based on the literature. Section 5.4 evaluates system performance and Section 5.5 discusses the results of the study.

5.1 Complementizer Choice in Surface Realization

That-complementizers are optional words that introduce sentential complements in English. In the Penn Treebank, they are left out roughly two-thirds of the time, thereby enhancing conciseness. This follows the low complementizer rates reported in previous work (Tagliamonte and Smith, 2005; Cacoullos and Walker, 2009). While some surface realizers, such as FUF/SURGE (Elhadad, 1991), have made use of input features to control the choice of whether to include a that-complementizer, for many applications the decision seems best left to the realizer, since multiple surface syntactic factors appear to govern the choice, rather than semantic ones. In our experiments, we use OpenCCG logical form inputs underspecified for the presence of that in complement clauses (CCs). Though other
function words are also not specified in the logical form, some function words like infinitival to are looked up during realization (on the basis of the to feature in lexical categories of main verbs) in order to meet grammatical constraints. However, that-complementizers are left completely optional i.e., the grammar licences both overt that complement clauses as well as complement clauses with no complementizer. While in many cases, adding or removing that results in an acceptable paraphrase, the absence of that in (105) prompts the interpretation where the adverb has scope over the verb say, which the original Penn Treebank sentence avoids by including the complementizer.

(104) He [said that [for the second month in a row, food processors reported a shortage of nonfat dry milk]]. (WSJ0036.61)

(105) ? He [said for the second month in a row], [food processors reported a shortage of nonfat dry milk].

The starting point for this chapter is White and Rajkumar’s (2009) realization ranking model, a state-of-the-art model employing features designed using basic insights from linguistic theory and of the kind discussed in (Johnson, 2009). An error analysis of this model, performed by comparing CCGbank Section 00 realized derivations with their corresponding gold standard derivations, revealed that out of a total of 543 that-complementizer cases, the realized output did not match the gold standard choice 82 times. Most of these mismatches involved cases where a clause originally containing a that-complementizer was realized in reduced form, with no that. This under-prediction of that-inclusion is not surprising, since the realization ranking model makes use of baseline n-gram model features, and n-gram models are known to have a built-in bias for strings with fewer words.
We report here on experiments comparing this global model to ones that employ local features specifically designed for *that*-choice in complement clauses. As a prelude to incorporating these features into a model for realization ranking, we study the efficacy of these features in isolation by means of a binary classification task to predict the presence or absence of *that* in complement clauses. In a global realization ranking setting, the impact of these phenomenon-specific features might be less evident, as they would interact with other features for ordering and lexical choices that the ranker makes. Note that a comprehensive ranking model is desirable, since linear ordering and *that*-complementizer choices may interact. For example, Hawkins (2003) reports examples where explicitly marked phrases can occur either close to or far from their heads as in (106) and (107), whereas zero-marked phrases are only rarely attested at some distance from their heads and prefer adjacency, as (108) and (109) show.

(106) I realized [that he had done it] with sadness in my heart.
(107) I realized with sadness in my heart [that he had done it].
(108) I realized [he had done it] with sadness in my heart.
(109) ? I realized with sadness in my heart [he had done it].

5.2 Uniform Information Density Principle and *that*-mentioning

In the context of *that*-mentioning, the uniform information density principle discussed by Jaeger (2010) predicts that language production is affected by a preference to distribute information uniformly across the linguistic signal. This hypothesis is tested against evidence from syntactic reduction in spontaneous speech. In Jaeger’s investigation, uniform information density emerged as an important predictor of speakers’ syntactic reduction
preferences even when taking a sizeable variety of controls based on competing hypotheses into account. Jaeger’s experiments were based on the Switchboard corpus of spoken dialogues and used a multi-level logit model analysis with the following controls factors:

1. **Dependency processing accounts**: Since previous studies showed that dependency length processing accounts were a factor in *that*-choice, Jaeger’s study included the starting position of the complement clause in the sentence, distance between the complement clause and the verb and the length of the clause itself as control terms in his study.

2. **Overt production difficulty at complement clause onset**: This was modelled by means of log and squared speech rate, the presence of a pause immediately preceding the clause and the normalized disfluency rate at the complement clause.

3. **Lexical retrieval at and before the complement clause onset**: This factor was incorporated on the grounds that the nature of the matrix and complement clause subjects contributed to *that*-mentioning. These were calculated using types and frequencies of subjects and matrix verbs. Hawkins (2004) also comments on how the complement clause subject can influence the choice of *that*-mentioning:

   (110) I realize the boy knows the answer

   (111) I realize that the boy knows the answer

   (112) ? I realize the small young boy knows the answer

   (113) I realize that the small young boy knows the answer
Hawkins’ processing metric predicts an equal preference for the first two examples while (113) is judged better compared to (112) because of the presence longer (structurally more complex) CC subject the small young boy.

4. **Ambiguity avoidance at complement clause onset**: This was calculated using the ambiguity of a given matrix verb in terms of the number of subcategorization frames it is compatible with.

5. **Additional controls**: Syntactic persistence (i.e., presence or absence of that in the most recent complement clause) and the gender of the speakers were also considered.

Information density at the CC onset was estimated by using matrix verb subcategorization frequency using the formula $-\log P(CC \mid matrix\ verb\ lemma)$. Even in the presence of the above controls, information density emerged as a significant predictor of speakers’ production of the optional that-complementizer. Our experiments reported in the remainder of the chapter confirm the efficacy of the features based on Jaeger’s work (including information density–based features) for generating written text.

### 5.3 Feature Design

White and Rajkumar’s (2009) global realization ranking model discussed earlier serves as the baseline for this chapter. Table 5.1 shows the new complementizer-choice features investigated in this chapter. The example features mentioned in the table are taken from the two complement clause (CC) forms (with-that CC vs. that-less CC) of the sentence below:

(114) The finding probably will support those who **argue** [that/∅] the U.S. should regulate the class of asbestos including crocidolite more stringently than the common kind
of asbestos, chrysotile, found in most schools and other buildings], Dr. Talcott said.

(WSJ0003.19)

We introduced three new classes of features as part of this work:

1. **Dependency Processing Features**: The first class of features, dependency length and position of CC, have been adapted from the related control features in Jaeger’s (2010) study. For the above example, the position of the matrix verb with respect to the start of the sentence (feature name mvInd and having the value 7.0), the distance between the matrix verb and the onset of the CC (feature name mvCCDist with the value 1.0) and finally the length of the CC (feature ccLen with value of 29.0 for the that-CC and 28.0 for the that-less CC) are encoded as features.

2. **Matrix Verb Features**: The second class of features includes various properties of the matrix verb, viz. POS tag, form, stem and supertag (feature names mvPos, mvStem, mvForm, mvSt, respectively). These features were motivated by the fact that
Jaeger controls for the per-verb bias of this construction, as attested in the earlier literature.

3. **Uniform Density Features**: The third class of features are related to information density. Jaeger (2010) estimates information density at the CC onset by using matrix verb subcategorization frequency. In our case, more like the $n$-gram features employed by Levy and Jaeger (2007), we used log probabilities from two existing $n$-gram models, viz. a trigram word model and trigram word model with semantic class replacement. For each CC, two features (one per language model) were extracted by calculating the average of the log probs of individual words from the beginning of the complement clause. In the *that*-CC version of the example above, local complement clause features having the prefix $uidCCMean$ were calculated by averaging the individual log probs of the 3 words *that the U.S.* to get feature values of -0.8353556 and -2.0460036 per language model (see last part of Table 5.1). In the *that*-less CC version, $uidCCMean$ features were calculated by averaging the log probs of the first two words in the complement clause, i.e., *the U.S.*

### 5.4 Classification Experiment

To train a local classification model to predict the presence of *that* in complement clauses, we used an averaged perceptron ranking model with the complementizer-specific features listed in Table 5.1 to rank alternate with-*that* vs. *that*-less CC choices. For each CC classification instance in CCGbank Sections 02–21, the derivation of the competing alternate choice was created; i.e., in the case of a *that*-CC, the corresponding *that*-less CC was created and vice versa. Table 5.2 illustrates classification results on Sections 00 (development) using models containing different feature sets & Section 23 (final test) for the
Table 5.2: Classification accuracy results (Section 00 has 170/543 that-CCs; Section 23 has 192/579 that-CCs)

<table>
<thead>
<tr>
<th>Model Features</th>
<th>% 00</th>
<th>% 23</th>
</tr>
</thead>
<tbody>
<tr>
<td>Most Frequent Baseline</td>
<td>68.7</td>
<td>66.8</td>
</tr>
<tr>
<td>Global Realization Ranking</td>
<td>78.45</td>
<td>77.0</td>
</tr>
<tr>
<td>Local That-Classification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Only UID feats</td>
<td>74.77</td>
<td></td>
</tr>
<tr>
<td>Table 5.1 features except UID ones</td>
<td>81.4</td>
<td></td>
</tr>
<tr>
<td>Both feature sets above</td>
<td><strong>83.24</strong></td>
<td><strong>83.02</strong></td>
</tr>
</tbody>
</table>

best-performing classification and ranking models. For both the development as well as test sections, the local classification model performed significantly better than the global realization ranking model according to McNemar’s $\chi^2$ test ($p = 0.005$, two-tailed). Feature ablation tests on the development data (Section 00) revealed that removing the information density features resulted in a loss of accuracy of around 1.8%.

5.5 Discussion

Table 5.3: Section 00 construction-wise that-CC proportions and model accuracies (total CC counts given in brackets alongside labels); gold standard obviously has 100% accuracy; models are local that-classification and White and Rajkumar’s (2009) global realization ranking model
As noted in the introduction, in many cases, adding or removing *that* to/from the corpus sentence results in an acceptable paraphrase, while in other cases the presence of *that* appears to make a substantial difference to intelligibility or fluency. In order to better understand the effect of the complementizer-specific features, we examined three construction types in the development data, viz. non-adjacent complement clauses, gerundive matrix verbs and a host of sub-cases involving a matrix *be*-verb (notably *wh*-clefts and *be*+adjective constructions), where the presence of *that* seemed to make the most difference. The results are provided in Table 5.3. As is evident, the global realization ranking model under-proposes the *that*-choice, most likely due to the preference of n-gram models towards fewer words, while the local classification model is closer to the gold standard in terms of *that*-choice proportions. For all the three construction types as well as overall, classifier performance was better than global ranking model performance. The difference in performance between the local classification and global ranking models in the case of gerundive matrix verbs is statistically significant according to the McNemar’s $\chi^2$ test (Bonferroni corrected, two tailed $p = 0.001$). The performance difference was not significant with the other two constructions, however, using only the cases in Section 00.
Table 5.4 lists relevant examples where the classification model’s *that*-choice prediction matched the gold standard while a competing model’s prediction did not. Example WSJ0049.64 is one such instance of classifier success involving a gerundive matrix verb (in contrast to the realization ranking model), Example WSJ0020.16 exemplifies success with a *wh*-cleft construction and Example WSJ0010.5 contains a non-adjacent CC. Apart from these construction-based analyses, examples like WSJ0044.118 indicate that the classification model prefers the *that*-CC choice in cases that substantially improve intelligibility, as here the overt complementizer helps to avoid a local syntactic ambiguity where the NP in allowed NP is unlikely to be interpreted as the start of an S.

Finally, we also studied the effect of the uniform information density features by comparing the full classification model to a model without the UID features. The full classification model exhibited a trend towards significantly outperforming the ablated model (McNemar’s $p = 0.10$, 2-tailed); more test data would be needed to establish significance conclusively. Examples are shown at the bottom of Table 5.4. In WSJ0060.7, the full classification model predicted a *that*-less clause (matching the gold standard), while the ablated classification model predicted a clause with *that*. In all such examples except one, the information density features helped the classification model avoid predicting *that*-inclusion when not necessary. Example WSJ0018.4 is the only instance where the best classification model differed in predicting the *that*-choice.

Since *that*-complementizer choice interacts with other realization decisions, in future work we plan to investigate incorporating these features into the global realization ranking model. This move will require binning the real-valued features, as multiple complement clauses can appear in a single sentence. Should feature-level integration prove ineffective,
we also plan to investigate alternative architectures, such as using the local classifier outputs as features in the global model.

5.6 Summary

This chapter reports results on a binary classification task for predicting the presence or absence of a that-complementizer using features adapted from Jaeger’s (2010) investigation of the uniform information density principle in the context of that-mentioning. Our experiments confirm the efficacy of the features based on Jaeger’s work, including information density–based features. The experiments also show that the improvements in prediction accuracy apply to cases in which the presence of a that-complementizer arguably makes a substantial difference to fluency or intelligibility. Our ultimate goal is to improve the performance of a ranking model for surface realization, and to this end we conclude with a discussion of how we plan to combine the local complementizer-choice features with those in the global ranking model.
CHAPTER 6: CONCLUSIONS AND OVERVIEW

6.1 Conclusions and Future Work

The main contribution of this thesis is the design and evaluation of linguistically motivated features for surface realization ranking that contributes to the generation of fluent and readable text. Specifically, these features have been successful in effectively modelling the three inter-related tasks of surface realization described in the introduction, viz. constituent ordering, inflection and agreement and function word insertion. A summary of the findings of this study in the context of each of these, along with ideas for future work are presented below:

1. **Constituent ordering**: We investigated dependency length minimization in the context of realization ranking, focusing on its potential to eliminate egregious ordering errors as well as better match the distributional characteristics of sentence orderings in news text. When added to a state-of-the-art, comprehensive realization ranking model, we showed that including a dense, global feature for minimizing total dependency length yields statistically significant improvements in BLEU scores and significantly reduces the number of egregious heavy-light ordering errors. Going beyond the BLEU metric, we also conducted a targeted human evaluation to confirm the utility of the dependency length feature in models of varying richness. Interestingly, even with the richest model, in some cases we found that the dependency
length feature still appears to go too far in minimizing dependency length, suggesting that further counter-balancing features—especially ones for the sentence-initial position (Filippova and Strube, 2009)—warrant investigation in future work.

2. **Inflection and agreement**: We have shown for the first time that incorporating linguistically motivated features to ensure correct animacy and number agreement in a statistical realization ranking model yields improvements over a state-of-the-art baseline. While agreement has traditionally been modelled using hard constraints in the grammar, we have argued that using a statistical ranking model is a simpler and more robust approach that is capable of learning competing preferences and cases of acceptable variation. Our approach also approximates insights about agreement which have been discussed in the theoretical linguistics literature. We have also shown how a targeted error analysis can reveal substantial reductions in agreement errors. As future work, we also plan to learn such patterns from large amounts of unlabelled data and use models learned thus to rank paraphrases.

3. **Function word insertion**: We have shown that using linguistically motivated features for English *that*-complementizer choice in a local classifier can improve upon the prediction accuracy of a state-of-the-art global realization ranking model employing myriad shotgun features, confirming the efficacy of features based on Jaeger’s (2010) investigation of the uniform information density principle in the context of *that*-mentioning. Since *that*-complementizer choice interacts with other realization decisions, in future work we plan to investigate incorporating these features into the global realization ranking model. This move will require binning the real-valued features, as multiple complement clauses can appear in a single sentence. Should
feature-level integration prove ineffective, we also plan to investigate alternative architectures, such as using the local classifier outputs as features in the global model.

As future work, the following themes could also be pursued. The impact of the feature sets described in this thesis could also be tested using other machine learning algorithms like MIRA and SVM ranking. Finally a more comprehensive human evaluation of the output of the realization system using crowd sourcing technologies would also be informative.
Anttila, Arto, Matthew Adams and Mike Speriosu (2010). The role of prosody in the English dative alternation. Language and Cognitive Processes. xvi, 63, 69, 70, 74


Cacoullos, Rena Torres and James A. Walker (2009). On the persistence of grammar in discourse formulas: A variationist study of “that”. Linguistics 47(1), 1–43. 120


Cahill, Aoife, Martin Forst and Christian Rohrer (2007a). Designing Features for Parse Disambiguation and Realisation Ranking. In Miriam Butt and Tracy Holloway King


138


140
Hawkins, John A. (2003). Why are zero-marked phrases close to their heads? In Günter Rohdenburg and Britta Mondorf (eds.), *Determinants of Grammatical Variation in English*, Berlin: De Gruyter Mouton, Topics in English Linguistics 43. 122


Pollard, Carl and Ivan Sag (1994). Head-Driven Phrase Structure Grammar. University Of Chicago Press. iii, 6, 68, 94, 100


147

von der Malsburg, Titus and Shravan Vasishth (2011). What is the scanpath signature of syntactic reanalysis? *Journal of Memory and Language* 65(2), 109 – 127. 8

Wasow, Thomas and Jennifer Arnold (2003). *Post-verbal Constituent Ordering in English*. Mouton. xvi, 63, 64, 67, 69, 83


White, Michael and Rajakrishnan Rajkumar (2009). Perceptron Reranking for CCG Realization. In *Proc. of EMNLP-09*. ii, xv, 1, 2, 6, 13, 49, 54, 55, 60, 73, 76, 92, 93, 109, 121, 124, 127


