Deciding Polarity of Opinions over Multi-Aspect Customer Reviews

A thesis presented to
the faculty of
the Russ College of Engineering and Technology of Ohio University

In partial fulfillment
of the requirements for the degree
Master of Science

Naime F. Kayaalp
December 2014

© 2014 Naime F. Kayaalp. All Rights Reserved.
This thesis titled
Deciding Polarity of Opinions over Multi-Aspect Customer Reviews

by
NAIME F. KAYAALP

has been approved for
the Department of Industrial and Systems Engineering
and the Russ College of Engineering and Technology by

Gary R. Weckman
Associate Professor of Industrial and Systems Engineering

Dennis Irwin
Dean, Russ College of Engineering and Technology
ABSTRACT

KAYAALP, NAIME F., M.S., December 2014, Industrial and Systems Engineering

Deciding Polarity of Opinions over Multi-Aspect Customer Reviews

Director of Thesis: Gary R. Weckman

The problem studied in this research is related to text mining. This research provides a solution for the multi-aspect segmentation problem of textual reviews and aims to fill the gap of the existing segmentation models in literature. In this research, in order to identify aspect opinion pairs in reviews, a review segmentation approach is proposed based on the conjunction and punctuation. The approach aims to segment the reviews in such a way that each of the segments represents a different feature or aspect of the reviewed product or service.

An existing segmentation model, called a multi-aspect segmentation model, is adapted and implemented as well as the proposed segmentation model, named the improved multi-aspect segmentation model. Two systems achieved by two different segmentation approaches were evaluated in the task of aspect-opinion extraction from the English restaurant reviews. The proposed model achieved 8.2% improvement over the existing model when used for aspect-opinion identification.
ACKNOWLEDGMENTS

At the very first, I would like to start with my appreciation to my academic advisor, Dr. Weckman, for his continuous support and mentorship throughout my master’s level studies. Thanks to his supportive, organized and friendly personality I was able to complete my studies in a timely and successful manner. I also would like to thank to my thesis committee members for their support and guidance.

Moreover, special thanks to my family, especially my mom, dad and sister, who believe in me more than everybody else and supported me in any difficulty I faced during this period. Without their love and encouragements, it would be very difficult for me to complete my studies successfully. I also would like to thank to the most valuable family members, two of my grandmas and my grandfather. Their prayers were with me all the time. They believed in me more than I do and their support was priceless.

Finally, I should state that Can Celikbilek’s encouragements have been the source of motivation for me. His sensibility and encouragements helped me to overcome every difficulty I encountered. I am so happy to have such a supportive, kind and lovely person in my life.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>3</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>4</td>
</tr>
<tr>
<td>List of Tables</td>
<td>7</td>
</tr>
<tr>
<td>List of Figures</td>
<td>9</td>
</tr>
<tr>
<td>1. Introduction</td>
<td>10</td>
</tr>
<tr>
<td>2. Literature Review and Justification</td>
<td>14</td>
</tr>
<tr>
<td>2.1. Supervised Approaches for Opinion Classification</td>
<td>14</td>
</tr>
<tr>
<td>2.2. Unsupervised Approaches for Opinion Classification</td>
<td>20</td>
</tr>
<tr>
<td>3. Problem Statement and Thesis Objective</td>
<td>26</td>
</tr>
<tr>
<td>3.1. Problem Statement</td>
<td>26</td>
</tr>
<tr>
<td>3.2. Thesis Objective</td>
<td>27</td>
</tr>
<tr>
<td>4. The Proposed Methodology</td>
<td>29</td>
</tr>
<tr>
<td>4.1. System Overview</td>
<td>29</td>
</tr>
<tr>
<td>4.2. Data Collection</td>
<td>30</td>
</tr>
<tr>
<td>4.3. Database Normalization</td>
<td>32</td>
</tr>
<tr>
<td>4.4. Aspect Related Term Extraction in Restaurants Domain</td>
<td>35</td>
</tr>
<tr>
<td>4.4.1. Indexing</td>
<td>37</td>
</tr>
<tr>
<td>4.4.2. Categorization</td>
<td>39</td>
</tr>
<tr>
<td>4.5 Aspect Based Segmentation</td>
<td>60</td>
</tr>
<tr>
<td>4.5.1 Aspect Based Segmentation Model – MAS</td>
<td>61</td>
</tr>
</tbody>
</table>
4.5.2 Proposed Aspect Based Segmentation Model - IMAS .......................... 67
4.6 Opinion Identification of each Aspect-Based Segment .......................... 70
5. Experimentation and Results ...................................................................... 76
  5.1 Preparation of Test Data ................................................................. 76
  5.2 Results of Aspect Related Term Identification Step ............................ 79
  5.3 Results of Aspect Based Segmentation ................................................ 81
  5.4 Overall Results Achieved After the Identification of Opinions .............. 88
    5.4.1 Results – Old Model ............................................................... 89
    5.4.2 Results – New Model ............................................................... 89
6. Conclusions and Future Work ................................................................. 90
References .......................................................................................................... 93
Appendix A: Examples from Final Aspect Dictionaries ..................................... 97
Appendix B: Example Reviews From the Dataset and Their Corresponding Outputs by Application of IMAS ................................................................. 101
Appendix C: Python Code Used Only for MAS ................................................ 103
Appendix D: Python Code Used Only for IMAS ................................................. 105
Appendix E: Python Code Used to Assign Aspects to Segments ...................... 107
LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1</td>
<td>Transformation of JSON into csv</td>
<td>33</td>
</tr>
<tr>
<td>Table 2</td>
<td>Distribution and Type of Data Instances in the Selected Dataset</td>
<td>35</td>
</tr>
<tr>
<td>Table 3</td>
<td>Storage of Indexed Nouns in Nouns.txt File</td>
<td>38</td>
</tr>
<tr>
<td>Table 4</td>
<td>Initial Seed Sets for Food, Service, Ambiance and Price Aspects</td>
<td>41</td>
</tr>
<tr>
<td>Table 5</td>
<td>Similarity Thresholds and Similarity Scores for Each of the Seed Lists</td>
<td>43</td>
</tr>
<tr>
<td>Table 6</td>
<td>Categorization of the Term <em>pizza</em> to Its Related Aspect List</td>
<td>44</td>
</tr>
<tr>
<td>Table 7</td>
<td>Categorization of the Term <em>check</em> to Its Related Aspect List</td>
<td>45</td>
</tr>
<tr>
<td>Table 8</td>
<td>Similarity Scores Used in the Second Categorization Round</td>
<td>51</td>
</tr>
<tr>
<td>Table 9</td>
<td>Similarity Scores Used in the Second Categorization Round</td>
<td>51</td>
</tr>
<tr>
<td>Table 10</td>
<td>Categorization Round 3 - Similarity Thresholds and Similarity Scores for Each Seed List</td>
<td>54</td>
</tr>
<tr>
<td>Table 11</td>
<td>Similarity Thresholds Used in the Fourth Categorization Round</td>
<td>55</td>
</tr>
<tr>
<td>Table 12</td>
<td>Similarity Scores Used in the Fourth Categorization Round</td>
<td>56</td>
</tr>
<tr>
<td>Table 13</td>
<td>Categorization Round 5 - Similarity Thresholds and Similarity Scores for Each Seed List</td>
<td>59</td>
</tr>
<tr>
<td>Table 14</td>
<td>Initial Score Assignments of the Initial Segments</td>
<td>63</td>
</tr>
<tr>
<td>Table 15</td>
<td>Initial Segments and Their Expected and Assigned Aspects</td>
<td>67</td>
</tr>
<tr>
<td>Table 16</td>
<td>Final Segments According to Their Aspects</td>
<td>67</td>
</tr>
<tr>
<td>Table 17</td>
<td>Segments and Their Associated Aspects</td>
<td>69</td>
</tr>
<tr>
<td>Table 18</td>
<td>Final Segments According to Their Aspects</td>
<td>70</td>
</tr>
</tbody>
</table>
Table 19: Expected Segments, Aspects and Their Associated Opinions ...................... 75
Table 20: Tagged Nouns According to Their Aspects .............................................. 77
Table 21: IMAS - Segments and Their Expected Aspects ....................................... 78
Table 22: MAS - Segments and Their Expected Aspects ....................................... 78
Table 23: Precision & Recall Matrix ..................................................................... 80
Table 24: Manually Tagged Data Instance – Initial Segmentation ......................... 84
Table 25: Manually Tagged Data Instance – Final Segmentation ........................... 85
Table 26: Test Data Instance – Initial Segmentation ............................................. 86
Table 27: Test Data Instance – Final Segmentation ............................................... 87
Table 28: Overall Accuracy – Comparison of MAS and IMAS Models ................. 89
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Flow Chart - Methodology</td>
<td>30</td>
</tr>
<tr>
<td>2</td>
<td>Flow Chart - Aspect-Related Term Identification</td>
<td>37</td>
</tr>
<tr>
<td>3</td>
<td>Algorithm - Indexing</td>
<td>38</td>
</tr>
<tr>
<td>4</td>
<td>Flow Chart - Overview of the Categorization Process</td>
<td>40</td>
</tr>
<tr>
<td>5</td>
<td>Flow Chart - Categorization Round 1</td>
<td>42</td>
</tr>
<tr>
<td>6</td>
<td>Flow Chart - Categorization Round 1 Execution</td>
<td>46</td>
</tr>
<tr>
<td>7</td>
<td>Algorithm - Categorization Round 1</td>
<td>47</td>
</tr>
<tr>
<td>8</td>
<td>Flow Chart - Categorization Round 2</td>
<td>49</td>
</tr>
<tr>
<td>9</td>
<td>Flow Chart - Categorization Round 3</td>
<td>52</td>
</tr>
<tr>
<td>10</td>
<td>Algorithm - Categorization Round 3</td>
<td>53</td>
</tr>
<tr>
<td>11</td>
<td>Flow Chart - Categorization Round 4</td>
<td>55</td>
</tr>
<tr>
<td>12</td>
<td>Flow Chart - Categorization Round 5</td>
<td>56</td>
</tr>
<tr>
<td>13</td>
<td>Flow Chart - Similarity Assignment between the Food List and with the Next Noun in Uncategorized Nouns List</td>
<td>58</td>
</tr>
<tr>
<td>14</td>
<td>Algorithm - Categorization Round 5</td>
<td>60</td>
</tr>
<tr>
<td>15</td>
<td>Algorithm - Segmentation Process</td>
<td>66</td>
</tr>
<tr>
<td>16</td>
<td>Algorithm - Segmentation Process (Continues)</td>
<td>66</td>
</tr>
<tr>
<td>17</td>
<td>Distribution of Review Stars in the Review Corpus</td>
<td>71</td>
</tr>
<tr>
<td>18</td>
<td>Flow Chart - Opinion Assignment to a Segment</td>
<td>74</td>
</tr>
<tr>
<td>19</td>
<td>Overall System Evaluation</td>
<td>88</td>
</tr>
</tbody>
</table>
1. INTRODUCTION

Rapid use of Internet increases the amount of information available online. Nowadays, people are not just searching or looking for information on the Internet but they are also contributing to that information pool. Facebook, which is the most well-known online social networking site, can be shown as an example of the contributions of people make to the increasing information on web. Each and every like, comment, post or message of a Facebook user contributes to online information. As people interact with the Internet more in that way, the amount of subjective resources grows rapidly. Subjective resources are the ones, which contains assessments and state people’s opinions. In other words, they are the opinion-rich resources. Reviews on products, businesses or movies; and comments on politics, news, and ideas are some of the examples for those opinion-rich resources.

Since the Internet allows everybody to have easy and fast access to the reviews and comments on the web, people’s opinions have become more important for businesses because of their effects on advertisement and marketing applications (Schindler et al., 2014). A review generally indicates the weaknesses or strengths of products or services. Internet users read those reviews and make their decisions on buying reviewed products or trying reviewed services (Tang et al., 2009). For example, Ye et al. (2009) found out that positively oriented online hotel reviews have a direct effect on the traveler’s decision to book a hotel. Additionally, Zhu & Zhang (2010) showed that the sales of products, especially the ones that are not popular, are affected by online consumer reviews easily. Additionally, the people who are affected form those reviews are the ones who use the
Internet most regularly. Other than these, there is an interesting study conducted by Ghose & Ipeirotis (2011) which aims to observe the changes in customer behavior according to the product reviews that are available on the Web. Their study mostly focused on the effects of the subjectivity, readability and spelling errors of the reviews on customer behavior. However, one of their findings suggested that the increased sales of the products are proportional with the helpfulness of the reviews (Ghose & Ipeirotis, 2011). That is, if a review clearly explains the pros and cons of the product, it is more likely to have an effect on the sales of that reviewed product accordingly. As the examples demonstrate, reviews on the Web might be shown as one of the key business drives since analysis of reviews can provide feedback to the businesses on their products and services (Gamon & Aue, 2005). It allows businesses to strengthen their products and services accordingly so as to increase their customer satisfaction. Thereby, they may achieve stronger marketing functionality.

However, it is not easy to obtain the desired information from customer reviews and so analyze them. There are thousands of reviews on the Web even for one specific business or product. Moreover, they are in free-text format. Reading thousands of reviews one by one takes time and so it is costly (Gamon & Aue, 2005). Some companies or businesses ask their customers to fill out surveys. Still, they usually do not get the desired amount of information from the surveys (Zhu et al., 2011). The situation creates the demand for an automated system that processes the reviews posted online and converts them into the desired form. This allows businesses to have faster and cost-efficient analysis.
Due to the significant effects of review analysis on marketing applications, deriving opinions from the publicly available reviews has become a popular research area. It is considered as a branch of “text mining” (Bagheri et al., 2013). Text mining is known as the way of extracting necessary information from the text that is either unstructured or semi-structured (Gaikwad et al., 2014). Text mining has a broad application area. It is mostly used in retail, electronics, hotel and restaurant industries to analyze the customer reviews but reviews can also reveal interesting facts in various other industries such as insurance and healthcare where it is possible to receive textual customer feedback according to their experiences (Sullivan & Ellingsworth, 2003; Greaves et al., 2013). For example, it is possible to retrieve information on the cleanliness of a hospital, the way of patient treatment and recommendation to other patients by using the online customer reviews (Greaves et al., 2013).

For simplicity, the application of text mining on reviews is called “review mining” in this thesis. In review mining, the goal is to extract the opinions with their associated features or aspects (Dave et al., 2003; Popescu & Etzioni, 2007). Since reviews are unstructured as stated above, it is difficult to extract meanings from the reviews even with an automated system. Previously, supervised approaches that requires reasonable amount of tagged data were used for classifying opinions (Pang et al., 2000; Riloff et al., 2006; Greaves et al., 2013). However, supervised approaches mostly deal with the opinion or feature classification on an individual basis. In recent years, unsupervised models that do not require any prior knowledge about the data have started to be used (Kim&Hovy, 2004; Gamon et al., 2005; Zhu et al. 2011; Zhu et al. 2012).
These models generally aim to extract the opinion and the topic that belongs to that opinion. Still, there are deficiencies associated each of these approaches and more research on algorithms of opinion extraction from free-text is needed for better analysis.

Considering the significance of the text mining applications, the objective of this thesis is to develop an unsupervised feature-opinion extraction model for the reviews in which more than one feature and opinion is mentioned at the same time. The proposed model is developed in the restaurants domain. The next section of this document is the literature review where the existing approaches for deriving opinions from customer reviews will be introduced with some background information. The drawbacks of the existing methodologies will be addressed and the motivation behind the proposed approach for opinion polling in multi-feature (aspect) reviews will be explained in the following section. Next, the methodology for the proposed approach will be provided and the results achieved at the end of each step that is used in the methodology will be discussed. In the conclusion section, the evaluation and limitations of the overall system and possible future work will be provided.
2. LITERATURE REVIEW AND JUSTIFICATION

Deriving opinions from textual reviews can be considered as a classification problem because reviews can be classified as positive, negative or neutral in terms of the opinion they state. Opinion extraction is also called sentiment analysis or sentiment classification. Sentiment has the same meaning as opinion. In the rest of this document sentiment and opinion are used interchangeably. Currently, there are two major types of approach for classifying opinions in textual reviews. These are supervised and unsupervised approaches that are discussed in the next two sections.

2.1. Supervised Approaches for Opinion Classification

The first approach used for opinion classification is supervised learning. Given a labeled review corpus, called as training data, supervised learning methods aim to classify unlabeled reviews according to the information learned from the training data. A descriptive model for supervised sentiment classification of textual reviews is shown below (Tsytsarau & Palpanas, 2011):

\[ \text{Training data: } \{(f(x) = R_i \in R, S_i \in S)\}, \text{ find } g: R \rightarrow S, g(R_i) = \arg \max f(R_i, S_i) \quad (2.1) \]

In the above model (2.1), R represents the review set while S represents the sentiment set. The aim is to find the best model, denoted with f, which maximizes the accuracy of the mapping g between a review and a sentiment.

Supervised learning methods have shown to be successful in the area of sentiment analysis. Pang et al. (2002) performed classification on movie reviews in terms of the opinion they state. They compared three popular supervised classification algorithms,
which are Naïve Bayes (NB), Maximum Entropy (ME) and Support Vector Machines (SVMs). Brief information on each of these algorithms is given below.

NB: Naïve Bayes classifier is the implementation of probability theory, which is known as Bayes Theorem, through a classifier (Lewis, 1998). Bayes is one of the widely used theories, which is still being used in the field of sentiment classification (Melville et al., 2009; Tan et al., 2009). Below is the general formulation for Bayes Theorem in which variables are adapted for review classification problems:

\[
P(c | r) = \frac{P(c)P(r | c)}{P(r)}
\]  

(2.2)

In the above model (2.2), \( c \) denotes a class (positive, negative or neutral), whereas \( r \) denotes a review. \( P(c | r) \) is the probability of being in class \( c \) given review \( r \). In Bayes classifier, features are independent. That means, the probability of one event does not affect another event’s probability. Therefore, probability of \( c \) is not dependent on the probability of \( r \). From the review classification perspective, the aim of Bayes classifier is placing the review (\( r \)) into the class (\( c \)) that maximizes the probability of \( r \) being in \( c \), which is shown by \( c = argmax_c P(c | r) \) (Pang, et al., 2002).

ME: In 1996, Berga et al. (1996) presented the use of maximum entropy concept with the text mining. The maximum entropy model corresponds to a log-linear probabilistic classification model. Unlike Bayes Classifier, it does not have the independence assumption so that one can embed the known facts into the model to make it more powerful. Probability calculated by maximum entropy for a given review and class is shown below (2.3) (Osborne, 2002):
In the equation, $Z(r)$ denotes a normalization function. Maximum Entropy assigns weights to the features. It decides on whether to use feature $i$ or not by the following equation (2.4) (Osborne, 2002):

$$F_{i,c}(r, c') = \begin{cases} 1, & f_i \text{ occurs in the review and } c' = c \\ 0, & \text{otherwise} \end{cases}$$ (2.4)

SVM: Support Vector Machines, which are not probabilistic classifiers, were first introduced by Cortes and Vapnik in 1995. In the SVM approach, each review is represented as a vector. The aim is to find the optimal hyper plane ($w$) that separates vectors of reviews, such that one side of the hyper plane consists of positive review vectors while the other side consists of negative review vectors. At the same time, this hyper plane should maximize the angle between two vector groups (Pang et al., 2002). The formula of SVM is given below (2.5) (Pang et al., 2002):

$$\overline{w} = \sum_j a_j c_j r_j \text{ where } a_j \geq 0$$ (2.5)

In the review mining area, the common point of supervised learning methods is that they all require a feature set to represent each review. The reason is that supervised models are only able to learn from training data if some features are assigned to each data entry. Pang et al. (2002) used “bag of words” approach for representing features of reviews. With the use of surveys, they identified unigrams and bigrams that might indicate an opinion. Then, they created feature sets of each review according to both the presence and frequency of those predefined feature terms. At the end of their comparison among SVM, NB and ME, they demonstrated that when a feature vector consists of only
unigrams according to their presence in a review, SVM method could reach up to 82.9% accuracy in sentiment classification.

In 2006, Riloff et al. pointed to the importance of feature vector selection as an input to supervised learning methods. They proposed a subsumption hierarchy model for selecting proper features so that they can increase the overall sentiment classification accuracy. Subsumption hierarchy decides whether to use unigrams, bigrams or extraction pattern as features so that each feature become more likely to state an opinion (Riloff et al., 2006). For example: if “benefits” is used as a feature, “tax benefits” may be retrieved as a feature, which does not indicate an opinion. On the other hand, if “benefits” + preposition is used as a feature, “benefits to” will be retrieved which indicates a positive opinion (Riloff et al., 2006). They classified documents and sentences in terms of the opinion they state. They used their model in feature vector selection, which is then supplied as an input to SVM classifier. Although their methodology worked well with document level classification, it was able to achieve a 1% improvement on sentence level classification. Overall, they achieved up to 98.7% accuracy in document level opinion classification and 74.9% accuracy in sentence level opinion classification with their model.

Supervised classification approaches also used on restaurant reviews (Govindarajan, 2014; Fan et al., 2014). Govindarajan (2014) used the Yelp Academic Dataset to classify the reviews according to the opinions they state. He tried to predict the number of stars (1-5) that a review is assigned by a customer. His approach was different than the other supervised opinion classification methods since he used an algorithm that
combines three popular supervised algorithms, which are Naïve Bayes, Support Vector Machines and Genetic Algorithms. Similar to the other supervised classification methodologies; bag-of-words approach was used while choosing the features of the data instances (Pang et al., 2002). However, not every token in each of these reviews are used. A pre-processing step was performed to reduce the size of the reviews since the reviews that are found in Yelp Dataset are long and complex. Govindarajan (2014) then selected three random data groups among the whole data instances. The data instances that are not selected for any of these three samples were used as the test data. On the three selected groups, Govindarajan (2014) trained NB, SVM and GA models respectively and performed prediction on the same test data for each of them. The dominant class of a test data instance was chosen as its predicted class. He achieved 92.44% accuracy in the task of classifying sentiments (classifying the reviews according to their stars).

On the other hand, by using the Yelp Academic Dataset, Fan et al. (2014) aimed to predict the overall business stars of a restaurant by analyzing its textual reviews using supervised approaches. As one of the important points of their research, Fan et al. provided an example indicating that the two reviewers of the same restaurant gave different stars although both of them indicated that they liked the restaurant in their reviews. One of these reviewers gave 3 stars while the other gave 5 stars to the same business. This finding of Fan et al. (2014) on the Yelp Dataset has become source of motivation for the methodology used in the opinion identification step of this thesis.

In addition to the opinion classification, extraction of the reasons behind those opinions was also considered as a problem that might be solved by using supervised
approaches (Hu & Liu, 2006). Since supervised classification requires labeled data, reviews to perform this task were collected from the websites that states the pros and cons of the product along with the review text itself. Hu & Liu (2006) used maximum entropy classifier on the customer reviews by identifying the unigrams, bigrams and trigrams; using positional features of these terms; and using pre-identified opinion words. Their model is tested in two different data sets which are mp3 reviews and restaurant reviews. With their approach they were able to achieve 61% F-score in the category of cons (Hu & Liu, 2006).

Although the accuracy in supervised learning is more than 80% most of the time, there are some problems associated with that approach. Supervised learning requires a huge amount of labeled training data. Although too much can cause over fit of the model, as the number of labeled reviews increases, supervised learning algorithms perform better (Tsytsarau & Palpanas, 2011). It is possible to find many reviews but it is not easy to tag (as positive, negative or neutral) thousands of reviews manually. Moreover, supervised learning requires quality and clean data for achieving higher accuracy, which is not possible for a free-text review format on the Web. Aside from these, the most important deficiency with supervised learning models is that they are not much suitable for identifying multiple opinions in a text or review. All of these problems with supervised learning led to the research of unsupervised learning approaches in the field of opinion mining.
2.2. Unsupervised Approaches for Opinion Classification

Unsupervised learning approaches do not learn from training data that is labeled beforehand. In fact, they use the current information hidden in the data to classify the reviews. This hidden information is called the semantic orientation of words such as parts of speech and the order of words (Chaovalit & Zhou, 2005). Unsupervised opinion classification usually reveals feature-opinion pairs, which might be thought as summarization of the text (Dave et al., 2003). In general, the first step in unsupervised opinion classification is to extract the words that represent features of the reviews and the second step is to extract their associated opinions.

Unsupervised learning techniques are generally evaluated with precision, recall and F1 measures rather than the accuracy since their output is in the form of feature-opinion pairs, not in single opinion classes. From a general point of view, precision is the number of correctly classified instances divided by the total number of instances whereas recall is the number of correctly classified instances divided by the total number of instances that belongs to the same class. F1 is a measure that is equivalent to the harmonic mean of precision and recall.

Unsupervised learning methods used for opinion classification are generally domain specific because the features and opinion words for each domain different than the other. Hu and Liu (2004), proposed a model to identify opinions on the features of electronic products such as camera, software and picture that are stated in a review. The aim was to identify feature opinion pairs found in the reviews. To identify the feature terms, they extracted most frequently occurring nouns and noun phrases from their
corpus. Then they eliminated the unnecessary ones by use of heuristic methods based on the order and occurrence of the terms in a single sentence. The remaining nouns and noun phrases are used as features. For the opinion words, they extracted the adjectives that are located just before the identified features and tagged them as either negative or positive by using WordNet dictionary. WordNet is a lexical dictionary for English that was developed by a group of researchers from Princeton University (Miller, 1995; Miller et al., 1990). Kim & Hovy (2004) also used WordNet for their unsupervised opinion extraction model. WordNet provides synonyms and antonyms of the words, if they have any (Miller, 1995). By using the opinions stated by synonyms or antonyms of an unlabeled opinion word, it is possible to decide on the opinion of that word. Since there is usually more than one synonym or antonym for a word, Kim and Hovy (2004) compared probabilities of possible opinions that can be assigned to an opinion word. In Hu and Liu’s (2004) model, a similar approach that considers the dominancy of opinions was used while identifying the opinions stated by adjectives.

After opinion word identification, Hu and Liu (2004) reevaluated the possible feature terms that were eliminated in the feature identification step. They considered them as feature terms again, if they appear after an opinion word which is also used with a frequent feature term in the review corpus. In the end, with their feature-opinion pairs, their model achieved 80% recall and 72% precision on average, which is considered successful for an unsupervised sentiment analysis technique.

A year later, Gamon et al. (2005) developed a system that visualizes the features and opinions for a specified car make and model. Their system combined both supervised
and unsupervised techniques to achieve sentiment classification. For the first step, they extracted car makes and models from the review data and then they selected the sentences that contain a make and an associated model. To identify whether a sentence is positively or negatively dominant, they tagged some of the randomly chosen sentences in the corpus and created feature vectors with the relevant terms in each sentence. They trained a Naïve Bayes classifier to identify opinion polarity of the remaining review sentences (Gamon et al., 2005). To identify the features mentioned in the review sentences, they developed an unsupervised clustering algorithm. A list containing the terms that are relevant in the car domain, and another list containing the terms that are irrelevant in the car domain were created. Then, the clusters were created by using term frequency (tf), count of words in the relevant words list and a measure used in document classification applications called tf-idf (Gamon et al., 2005). Tf-idf is a term weighting measure that increases as the number of occurrences of a term in a document increases and decreases as the number of words in the corpus increases (Paltoglou & Thelwall, 2010). At the end, clustering assigned a feature for each sentence, which was then visualized with the opinion assigned to that sentence. Since positive reviews comprised almost 63% of the dataset, they achieved more than 90% recall on positive reviews. However, they achieved 88% precision on the negative reviews and 65% precision on the neutral reviews though both had very low recall.

Feature-opinion extraction task can also be considered as a text summarization problem since the aim is to extract the important information from the text while ignoring the rest of it. Blair-Goldensohn et al. (2008) developed a text summarization approach
which they call aspect-based summarization to identify the aspect that is being mentioned in the local service reviews. The system that they came up with was consisting of three main steps. At first, the sentences that state an opinion are extracted by use of positively and negatively oriented seed terms, WordNet library (Miller, 1995) and a maximum entropy classifier which is trained on some pre-labeled sentences. For the aspect identification, they again used combination of supervised and unsupervised models and as the final step they aggregated these sentiments with their corresponding aspects (Blair-Goldensohn et al., 2008). Although their system was successful as a text summarization system which is not strictly domain specific, their approach assumed that a sentence can only mention one aspect of the local service.

Thus far, the sentiment analysis models described above have been used for sentence level reviews. In each of the models, it was assumed that a sentence mentions only one feature and opinion. However, in free-text format a sentence may contain more than one feature as well as more than one opinion. To tackle with this problem, Zhu et al. (2011) developed an algorithm that segments sentences according to the features they are mentioning and gets the opinions associated with them. They developed their model in the Chinese restaurant reviews domain. For each review the task was to identify whether it mentions food, service, price, staff or ambiance aspects as well as the associated opinions with them. Their model consists of three steps: aspect related words extraction, sentence segmentation and opinion extraction.

In the first step, frequent nouns, verbs, adjectives, adverbs and phrases were extracted from the review corpus as candidate feature terms. Seed dictionaries were also
created by hand for each of the aspects (features). Considering the similarity between the
terms found in dictionary and the terms not found in dictionary, they expanded the
dictionaries. They calculated similarity measures by using RlogF metric (Riloff, 1996),
which is based on the frequency of the terms. For the sentence segmentation, Zhu et al.
(2011; 2012) developed an algorithm that decides on the optimal segmentation points in a
sentence such that each segment represents a different aspect of a restaurant. For their
initial segmentation, they used the fact that single aspect segments are likely to be
separated by a comma or semicolon, if there are multiple aspects in a sentence. They
compared their “multi aspect segmentation (MAS)” model with the ordinary review
segmentation, which treats a sentence as a segment representing only one aspect. They
discovered that the MAS model has higher precision, recall and F1 values. After
segmenting each review into single aspect segments, they identified the opinions stated
by each segment by using an existing Chinese opinion words dictionary. They evaluated
the opinion classification accuracy for each of the segmentation models. Results showed
that reviews that are segmented with the MAS model achieved the highest accuracy,
which is around 76%.

There are similar approaches to Zhu’s that deal with the reviews which contain
multiple aspects (Marrese-Taylor et al., 2013; Xu et al., 2013; Bagheri, 2013). Marrese-
Taylor et al. (2013) used the orientation of words in a review in addition to the frequency
metrics for the task of aspect identification. In addition to this, Marrese-Taylor and his
team depicted the fact that “but” is an important conjunction which could be used to
identify multiple opinions in reviews. Another model is developed by Bagheri et al.
(2013) and it deals with extracting aspects from the electronic product reviews retrieved from www.Amazon.com. The aspects that they were trying to extract were battery life, sound quality, size, phone, signal and weight. In their model, rather than selecting the seed sets manually for those aspects, they performed the initial selection by using an unsupervised method called A-score (Bagheri et al., 2013). The A-score metric was based on the frequency of the terms that are used in the review corpus and the mutual information of the terms with each other, where the mutual information refers to a measure which is based on the probability of occurrence of the two random terms together. Then, they also designed an iterative algorithm on the remaining candidate terms by using the same A-score metric to classify them into their corresponding seed lists (Bagheri et al., 2013). Their algorithm also includes a pruning step where they tried to get rid of the redundant terms that are classified as aspects. In the pruning step, they used heuristics that are based on the occurrence of the terms with opinion terms. Their model achieved 87.5% precision and 65% recall on the product reviews for the task of extracting the aspect related terms (Bagheri et al., 2013).
3. PROBLEM STATEMENT AND THESIS OBJECTIVE

3.1. Problem Statement

Customer reviews that are found on the Web are generally in the form of paragraphs. They do not have a specific paragraph structure but they consist of several unstructured sentences. Each sentence in these paragraphs may or may not mention more than one feature. Although the opinion extraction models explained above, before the Zhu et al. (2011)’s model are all for sentence level reviews and usually deal with just one feature, the models developed by Zhu et al. (2011) and Bagheri et al. (2013) considers the occurrence of more than one feature in a sentence. As it was stated before, Zhu et al. (2011)’s model segments each sentence such that at the end of the segmentation, adjacent segments refer to different aspects. Their model also depicts the fact that people often separate features in the same sentence with a comma or semicolon. Thus, in Zhu et al. (2011)’s model initial segmentation is based on comma and semicolon. A simple example is illustrated with a review sentence 3.1:

\[ \text{Pizza was tasty, service was not very impressive, but it was good.} \quad (3.1) \]

Initial segmentation:

Segment 1: \textit{Pizza was tasty}

Segment 2: \textit{service was not very impressive}

Segment 3: \textit{but it was good}

After the application of multi aspect segmentation model:

Segment 1: \textit{Pizza was tasty}

Segment 2: \textit{service was not very impressive, but it was good}
For this type of sentence, their model worked well. It is obvious that segment 1 is about a food aspect while segment 2 is about a service aspect. However, people often use punctuation and conjunction words such as “and” and “but” interchangeably while writing in free-text format. If the first comma were replaced with “and” in the above sentence, the results would be different according to Zhu et al.’s (2011) model. A simple example is illustrated with the modified review sentence 3.2:

*Pizza was tasty* and *service was not very impressive, but it was good.*  

(3.2)

Initial segmentation:

Segment 1: *Pizza was tasty and service was not very impressive*

Segment 2: *but it was good*

After the application of the multi aspect segmentation model:

Segment 1: Pizza was tasty and service was not very impressive, but it was good.

For this type of sentence, Zhu et al.’s model (2011) is insufficient to segment sentences correctly, which may lead to derivation of incorrect or missing aspect-opinion pairs from the reviews.

3.2. Thesis Objective

Segmenting each sentence according to the aspects of the domain that they are related to is a successful method to identify aspect-opinion pairs from the reviews that mention about one or more aspects. However, the most important deficiency with the existing segmentation model (Zhu et al., 2011) is that initial segmentation is based on a comma or semicolon as it was stated before. The objective of this thesis is to develop a better multi-aspect sentence segmentation model such that aspects separated by...
conjunction words such as “and” and “but” will also be considered. Thus, the objective is to increase the accuracy in aspect-opinion matching. Additionally, the existing segmentation model and the system to extract the aspect-opinion pairs were developed on Chinese restaurant reviews whereas the proposed model will consider English restaurant reviews. Therefore, the opinion identification phase will also differ since this step will be performed by using English opinion words dictionary. According to the literature review that has been done so far, there is not much research on multi-aspect segmentation models on English restaurant reviews which also considers aspect and opinion related term identification steps. Therefore, the proposed model will have a significant contribution to English text mining applications with its aim and approach. The approach that is used in this thesis can also be categorized in text summarization branch of text mining. In text mining, summarization is the task of extracting the important or necessary information from the text in order to shorten the length, reduce the complexity of that text and put that text into structured format (Gaikwad et al., 2014). That is, converting the unstructured reviews into a set of aspect-opinion pairs can also be defined as summarization.
4. THE PROPOSED METHODOLOGY

In this section, overview of the complete system that is developed for extracting aspect-opinion pairs and each of the system components that are necessary to achieve this task are provided. Zhu et al.’s (2011) methodology was the main influence for the development of the system. System components including data collection, database normalization and the three main models implemented for aspect-opinion extraction are explained in this section. Algorithms that are developed for aspect related term extraction, segmentation and opinion identification (polarity analysis) are discussed in detail.

4.1. System Overview

The methodology of this thesis consists of five main steps, which are data collection, database normalization, aspect related term identification, aspect-based segmentation and polarity analysis of opinions. The flow chart of the methodology is shown in Figure 1 and the detailed information for each of these steps is provided under the corresponding subtitles.
4.2. Data Collection

Yelp Phoenix Academic Dataset is used throughout this research. Yelp is a local business search website; it has applications for mobile devices as well. It allows people to view reviews or ratings of businesses such as hotels, restaurants and shops. At the same time Yelp users can write reviews and rate businesses according to their experiences, and can do check-ins at these businesses. Yelp Academic Dataset is open to the public for
academic use, which can be found at Yelp’s official website (Yelp Academic Dataset, 2013).

The raw dataset consists of four different data files, which are review data, user data, businesses data and check-in data. For this thesis, only business and review data is used. Both business and review data is in JSON (JavaScript Object Notation) format, which is a common format for online resources. Business data consists of the following columns which represent the features of a business: business ID, full address, open/closed, categories, city, review count, name, neighborhoods, longitude, state, stars, latitude, stars and type. On the other hand, reviews data consists of the following columns that represent the features of a review: votes, user ID, review ID, stars, date, text, type and business ID. Raw Business data has 11,537 numbers of instances while reviews data has 229,907 numbers of instances. Below are two examples from the raw business and review data (Yelp Academic Dataset, 2013):

Business Data Instance: {
"business_id": "PzOgRohWWw7F7YEPBz6AubA",
"full_address": "6520 W Happy Valley Rd\nSte 101\nGlendale Az, AZ 85310",
"open": true,
"categories": ["Food", "Bagels", "Delis", "Restaurants"],
"city": "Glendale Az",
"review_count": 14,
"name": "Hot Bagels & Deli",
"neighborhoods": [],
"longitude": -112.200264,
"state": "AZ",
"stars": 3.5,
"latitude": 33.712797000000002,
"type": "business"
}

Review Data Instance: {
"votes": {
"funny": 0,
"useful": 1,
"cool": 0
},
"user_id": "0hT2KtfLiobPvh6cDC8JQg",
"review_id": "IESLBzqUCLdSzSqm0eCSxQ",
"stars": 4,
"date": "2012-06-14",
"text": "love the gyro plate. Rice is so good and I also dig their candy selection :)",
"type": "review",
"business_id": "6oRAC4uyJCsJ11X0WZpVSA"}
4.3. Database Normalization

Database normalization involves the transformation of the raw data set, which is initially in JSON data format. JSON data format is transformed into the csv data format in order to be used during the development and testing of the proposed models. An example of both the JSON format and the csv format are given below. A random review data instance from the dataset is used for demonstration purposes.

As seen in Table 1, outer string fields of JSON data instance (Yelp Academic Dataset, 2013) such as votes, user_id, review_id, starts, date, text, type and business_id are mapped as the column titles of the corresponding csv file. On the other hand, the values of those string fields are mapped as the instances of the corresponding column titles in the csv file. This process is done for all the instances of both the reviews file and the business file. As an output, two csv files called “Reviews.csv” and “Business.csv” are created. After the mapping phase, columns that are not needed for the model construction and experimentation are removed from each of the files. From the “Reviews.csv” file, votes, user_id, date and type columns are removed. From the “Business.csv” file, open, categories, city, review_count, neighbours, longitude, latitude and state columns are removed. However, before removing these columns from “Business.csv” file, only the businesses that are tagged as restaurants, or cafes in their “categories” column are considered. Because we are working on the restaurants domain, the rest of the data instances that have the category tags like hotels, education, health etc. are eliminated since they are considered out of scope for this thesis.
Review JSON Data Instance (Yelp Academic Dataset, 2013): 
{"votes": {"funny": 0, "useful": 1, "cool": 0}, "user_id": "0hT2KtfLiobPvh6cDC8JQg", "review_id": "IESLBzqUCLdSzSm0eCSxQ", "stars": 4, "date": "2012-06-14", "text": "love the gyro plate. Rice is so good and I also dig their candy selection :)", "type": "review", "business_id": "6oRAC4uyJCsl1X0WZpVSA"}

Table 1

<table>
<thead>
<tr>
<th>votes</th>
<th>user_id</th>
<th>review_id</th>
<th>business_id</th>
<th>date</th>
<th>text</th>
<th>type</th>
<th>stars</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0hT2K...</td>
<td>IESLBzqUC...</td>
<td>6oRAC4u...</td>
<td>6/14/2012</td>
<td>love</td>
<td>review</td>
<td>4</td>
</tr>
</tbody>
</table>

As the next step, each review instance in “Reviews.csv” is matched with its corresponding business in “Business.csv” with the help of a program that is written in Python programming language. Recall that in both of the files, the business_id column is common. Therefore, matching process is performed by using the business_id column. At the end of review and business data matching process, 158,431 instances are found in the dataset. The columns that represent features of the data instances are text (review), review stars, review votes, business id and business stars, since the rest had already been eliminated.

For the experiments, 50 reviews that include sentence connections with the conjunctions “and” or “but”; and 50 reviews that do not contain neither of the conjunctions “and” and “but” are chosen randomly among the data instances. A program written in Python separated the dataset into two as the ones that contain either “and” or “but”; and the ones that do not contain any of these. The program further selected 50 instances from each of these data sets. These 100 chosen data instances are separated and
used as the test data as shown in Table 2. Test data instances are limited to 100 due to the workload of manual tagging. Test data will be used to evaluate the systems created using the proposed and the existing segmentation models. The same test data will also be used for the individual evaluation of aspect related term identification, segmentation and opinion extraction steps. In addition to these 100 reviews, 970 reviews are chosen randomly among the reviews that were not selected for the previous step. Again, the selection is done with the help of a simple Python script. During the identification of aspect related terms step, 100 reviews that were selected for the test phase and these 970 reviews will be used together as shown in Table 2. In the segmentation and opinion identification steps, only the 100 selected test instances will be used for testing purposes.
Table 2

Distribution and Type of Data Instances in the Selected Dataset

<table>
<thead>
<tr>
<th>Total Number of Review Data Instances</th>
<th>Number of Test Data Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Distribution and Type of Data Instances (Used in Aspect-Related Term Extraction Step)</td>
<td></td>
</tr>
<tr>
<td>Number of data instances that contain “and” or “but”</td>
<td>50</td>
</tr>
<tr>
<td>Number of data instances that do not contain “and” and “but”</td>
<td>50</td>
</tr>
<tr>
<td>Total</td>
<td>100</td>
</tr>
<tr>
<td>Total</td>
<td>1070</td>
</tr>
</tbody>
</table>

4.4. Aspect Related Term Extraction in Restaurants Domain

In the aspect related term identification step, the aim is to extract the terms associated with the food, service, price and/or ambiance aspects of a restaurant from the textual reviews. The first step is the identification of candidate terms in the reviews that may indicate one of these aspects. Although all of the terms/words in the corpus have the potential of indicating an aspect, only the nouns are considered as aspect indicating candidates. There are two main reasons for considering only the nouns as the aspect related terms. Firstly, the algorithm may be more inclined to consider meaningless or redundant words as aspect related terms, if there were not any limitations on the terms
that might indicate an aspect. Secondly, WordNet library (Miller, 1995), which provides core functionality to the implementation of the proposed algorithm, does not support similarities between the adjectives or adverbs. It is only capable to measure the similarity between the nouns and between the verbs on an individual basis. However, verbs alone are not good indicators of aspect related terms. Therefore, only nouns are considered as the initial candidates for aspect related terms.

Candidate terms identification and indexing are done at the same time. Indexing is the efficient way of storing the terms in files in usable formats. The format that is going to be used is dependent on the needs of the application. In the aspect related terms identification step, indexing is needed to store the candidate nouns in which each of the candidates are assigned an ID and stored with their number of occurrences in the review corpus. In addition to the nouns, reviews are indexed as well. That is, each review is assigned an ID and stored with the nouns and their counts in that review. Overall, aspect related term extraction is achieved by indexing followed by the categorization process as shown in Figure 2. Detailed information on the indexing methodology performed in aspect related term identification step is given in 4.4.1, while the detailed information on the categorization step is given in 4.4.2.
4.4.1. Indexing

This section briefly mentions the steps that are completed during the indexing phase. The indexed data format is also illustrated in this section. As the first step of indexing, each review text is extracted from the data set and each of the reviews is assigned an ID. Each review is further separated into sentences. Each term in each of these sentences is extracted. For each and every review, a text file (named as its assigned ID) that contains the tokenized words in that review, is created and stored under a directory.

At the time of the segmentation of reviews into the sentences and the tokenization of those sentences, each token is assigned its part of speech (POS) tags. The terms that are identified as nouns are further processed. If a term is identified as a noun or part of a noun phrase, it is put into its base form with the lemmatization process. Basically, lemmatization is putting a term into its simplest form. For example, the term “apples” could be lemmatized to the term “apple”; while the term “families” could be lemmatized
to the term “family”. WordNet library (Miller, 1995) has readily available lemmatization functions, which are utilized during the indexing phase as well. After the lemmatization step, an index file called “Nouns.txt” is created for keeping track of the nouns in the corpus. Each lemmatized noun type has been assigned an ID and stored with the corresponding noun itself and its overall count in the review corpus. Table 3 below is a demonstration of the index structure. The table demonstrates the randomly chosen three consecutive lines that are stored in Nouns.txt file. Additionally, pseudocode given in Figure 3 illustrates the process of indexing.

Table 3

Storage of Indexed Nouns in Nouns.txt File

<table>
<thead>
<tr>
<th>ID</th>
<th>Noun</th>
<th>Count in corpus</th>
</tr>
</thead>
<tbody>
<tr>
<td>230</td>
<td>india</td>
<td>3</td>
</tr>
<tr>
<td>231</td>
<td>honey</td>
<td>17</td>
</tr>
<tr>
<td>232</td>
<td>miele</td>
<td>4</td>
</tr>
</tbody>
</table>

1. Open the csv file that refers to the complete data file (The csv file with 1070 data instances)
2. Create an empty nouns list
3. For each line of the csv file
   3.1. Get the review ID
   3.2. Get the textual review
   3.3. If all terms list exists, empty the list. Else, create an all terms list
   3.4. Segment the review into the tokens and POS tag each token
   3.5. For each token, tag pair
       3.5.1. Clean the token // Lemmatize it and eliminate the punctuations from the beginning and end of the token if any
       3.5.2. If the tag indicates a noun, store it in the nouns list in addition to the all terms list. Else, store it only in the all terms list.
   3.6. Into the corresponding review file (identified by the review ID) put the terms that are stored in the all terms list
4. Into the “Nouns.txt” file put the nouns that are accumulated in the nouns list

*Figure 3. Algorithm - Indexing*
4.4.2. Categorization

Mainly, in the categorization process, nouns that are indexed and stored during the indexing step are placed into their corresponding seed list in order to create the final list of food, service, ambiance and price related terms. In the categorization step, WordNet’s (Miller, 1995) similarity functionality is utilized. However, there is a significant issue that needs to be taken care of before the categorization. Some of the terms that are identified as nouns are not always found in the WordNet dictionary. That is, some of them may be misspelled or some of them may not have a corresponding entry in the WordNet dictionary. For this reason, elimination among those terms is the initial step that is done prior to the categorization of terms.

Categorization consists of a total of five rounds, two of which are called pruning rounds. The relation between these five rounds, their inputs and outputs is shown in Figure 4. The reason that there are five categorization rounds is to maximize the accuracy of noun categorization (placing the nouns into the correct aspect lists). Basically, the first round does the initial placement of nouns. The second round is performed in order to eliminate the terms that are misclassified in the first round. The third round is performed in order to classify the nouns that could not be placed any of the lists during round one. Fourth round, which is same as the second round, is performed in order to eliminate the terms that are misclassified in the third round. Finally, the fifth round is performed in order to categorize the remaining uncategorized terms at the end of the third round. Each of these rounds is discussed in detail under the corresponding subsections.
4.4.2.1. Categorization Round-1

Categorization Round 1 processes the complete list of nouns that are extracted from the review corpus. It places the nouns to their corresponding aspect list according to their similarity with the terms that are already placed in those aspect lists. Initial aspect lists, which are also called the seed lists, are chosen for each of the aspects as shown in Table 4. Please note that $ is also added to price list. It is not being shown in initial aspect list since it is a marker which does not affect the rest of the aspect-related term extraction process.
During the categorization, the similarity of each uncategorized noun with each of the food, service, price and ambiance seed lists is measured. Each uncategorized noun is placed into the list that has the highest similarity score over a certain threshold. When a noun is placed into a list, it is tagged as categorized. Threshold values differ for each of the seed lists because these values are defined according to the possibility of placement into the lists. Therefore, adjusting and tuning the threshold values are important to place the relevant terms into the relevant dictionaries. If the threshold values are too low, most of the redundant or neutral terms may be placed into seed lists that they are not corresponding to. On the other hand, if the threshold values are too high, most of the relevant terms may not be categorized and might be identified as neutral terms that do not indicate an aspect.

Before starting the categorization, there were approximately 4884 terms that are identified as nouns in the review corpus. 4884 number of terms were split into 16 groups, with each of them having approximately 300 words. Threshold values are chosen with the experiments that are performed only on a group of terms consisting of 300 words.
However, after the threshold values are identified, the categorization process is performed on each of the 16 groups of nouns as seen in Figure 5.

Figure 5. Flow Chart - Categorization Round 1

For the initial categorization, the best similarity threshold values were found to be 0.70 for food dictionary, 0.75 for price, 0.75 for service and 0.75 for ambiance dictionaries as shown in Table 5. However, threshold values are not necessarily always used to place the terms. That is, if a term is found to be similar (over a certain similarity score which is defined as 0.92) with a term in either of the dictionaries, it is directly placed into that dictionary. Therefore, similarity score and similarity threshold are used as two different measures. Similarity threshold and similarity score values shown in Table 5 are identified based on the best categorization performance of the classification program written in Python.
Table 5

*Similarity Thresholds and Similarity Scores for Each of the Seed Lists*

<table>
<thead>
<tr>
<th>Aspect Lists</th>
<th>Similarity Threshold ($t_i$)</th>
<th>Similarity Score ($z_i$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Food List ($A_1$)</td>
<td>0.70</td>
<td>0.92</td>
</tr>
<tr>
<td>2 Service List ($A_2$)</td>
<td>0.75</td>
<td>0.92</td>
</tr>
<tr>
<td>3 Price List ($A_3$)</td>
<td>0.75</td>
<td>0.92</td>
</tr>
<tr>
<td>4 Ambiance List ($A_4$)</td>
<td>0.75</td>
<td>0.92</td>
</tr>
</tbody>
</table>

The methodology that has been used for the assignment of the similarity score between two terms is based on the synsets of the terms. The term “synset” stands for the term “synonym”. WordNet library (Miller, 1995), which is integrated with the Natural Language Toolkit (Loper & Bird, 2002) in Python programming language, provides methods to get the synsets of a term if there are any. To find the similarity between two terms (called A and B), the synsets of the first term A is compared with the synsets of the other term B. The highest similarity score that is found during this comparison is utilized as the similarity score between the two terms. To find the similarity of a term with a seed list, the synsets of the uncategorized term are compared with the synsets of the terms that are already found in that seed list. If any one of the similarity scores between the uncategorized term and a term from that specific list is over 0.92, the uncategorized term is placed into that specific list. Otherwise, the similarity scores between the uncategorized term and each of the terms in that specific list are summed up. Then the average value of this sum is used as the similarity of the uncategorized term to which it is being compared. This process is demonstrated with an example below.
Let us have the uncategorized term *pizza*. To categorize it, we find its similarity score for each term in the seed dictionaries, as shown in Table 6. The similarity score between the term *food* and *pizza* is found to be 0.95, which is over 0.92. Therefore, *pizza* is added to the food list directly. Please note that the similarity scores indicated in Table 6 and Table 7 may not be the same with the scores achieved during the runtime of the system. For demonstration purposes, these scores are used.

### Table 6

_Categorization of the Term “pizza” to Its Related Aspect List_

<table>
<thead>
<tr>
<th></th>
<th>Terms in the seed lists and their similarity scores with the term <em>pizza</em></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Food (A₁)</strong></td>
<td><em>food</em>: 0.95  <em>drink</em>: 0.70  <em>sweet</em>: 0.30  <em>fruit</em>: 0.40  <em>vegetable</em>: 0.50</td>
</tr>
<tr>
<td></td>
<td><em>bread</em>: 0.50  <em>flavour</em>: 0.34</td>
</tr>
<tr>
<td><strong>Service (A₂)</strong></td>
<td><em>clock</em>: 0.20  <em>waiter</em>: 0.10  <em>attitude</em>: 0.15  <em>server</em>: 0.50</td>
</tr>
<tr>
<td></td>
<td><em>assistance</em>: 0.40  <em>behavior</em>: 0.24</td>
</tr>
<tr>
<td><strong>Price (A₃)</strong></td>
<td><em>cost</em>: 0.20  <em>price</em>: 0.20  <em>bill</em>: 0.30  <em>fee</em>: 0.23  <em>money</em>: 0.25</td>
</tr>
<tr>
<td><strong>Ambiance (A₄)</strong></td>
<td><em>atmosphere</em>: 0.25  <em>concept</em>: 0.30  <em>activity</em>: 0.40  <em>design</em>: 0.45</td>
</tr>
<tr>
<td></td>
<td><em>decor</em>: 0.15  <em>fun</em>: 0.12</td>
</tr>
</tbody>
</table>

After the categorization of the term *pizza*, let us categorize the term *check*. This time, similarity scores are given in Table 7 below. As it is seen in the Table 7, none of the terms has a similarity score with the term *check* that is equal to 0.92 or over. Therefore, the similarities in each of the seed lists are summed up and their average is calculated. The results are illustrated in the rightmost column of the Table 7. Price seed list has the
highest score, which is 0.75 and is equal to the similarity threshold of the price list. Therefore, the term \textit{check} was categorized into the price list.

Table 7

\textit{Categorization of the Term “check” to Its Related Aspect List}

<table>
<thead>
<tr>
<th>Term</th>
<th>Terms in the seed lists and their similarity scores with the term \textit{check}</th>
<th>Average Scores</th>
</tr>
</thead>
<tbody>
<tr>
<td>Food (A₁)</td>
<td>\textit{food}: 0.40  \textit{drink}: 0.35 \textit{sweet}: 0.30 \textit{fruit}: 0.20 \textit{vegetable}: 0.30 \textit{pizza}: 0.30 \textit{bread}: 0.50 \textit{flavour}: 0.25</td>
<td>0.33</td>
</tr>
<tr>
<td>Service (A₂)</td>
<td>\textit{clock}: 0.20 \textit{waiter}: 0.80 \textit{attitude}: 0.15 \textit{server}: 0.50 \textit{assistance}: 0.40 \textit{behavior}: 0.24</td>
<td>0.38</td>
</tr>
<tr>
<td>Price (A₃)</td>
<td>\textit{cost}: 0.75 \textit{price}: 0.75 \textit{bill}: 0.85 \textit{fee}: 0.65 \textit{money}: 0.75</td>
<td>0.75</td>
</tr>
<tr>
<td>Ambiance (A₄)</td>
<td>\textit{atmosphere}: 0.25 \textit{concept}: 0.30 \textit{activity}: 0.40 \textit{design}: 0.45 \textit{decor}: 0.15 \textit{fun}: 0.12</td>
<td>0.28</td>
</tr>
</tbody>
</table>

This categorization process runs over the uncategorized nouns list continuously until the system reaches the idle state. The idle state occurs when there are not any uncategorized nouns that are eligible for the placement in any of the seed lists. The methodology used in the first round of the categorization is visualized with the flow chart shown in Figure 6.
Figure 6. Flow Chart - Categorization Round 1 Execution
The algorithm used in the categorization process is also illustrated with the pseudocode shown in Figure 7. Recall that $t_1, t_2, t_3$ and $t_4$ refers to the similarity thresholds for food, service, price and ambiance respectively; while $z_1, z_2, z_3$ and $z_4$ refers to the similarity scores for food, service, price and ambiance respectively. Additionally, let $U$ be the list that contains the uncategorized nouns shown by $U_i$ where $i$ is the $i^{th}$ noun; and let $s_{ij}$ be the similarity of the term $U_i$ with the seed list denoted by $A_j$. Also, let $S$ be the similarity function between any two terms denoted by $S(x, y)$. The output of the $S$ function refers to the highest similarity score between $x$ and $y$.

1: Continue until the idle state
2:   For $U_i$ in $U$
3:     For $A_j$ in $A$
4:         $s_{ij} \rightarrow 0$
5:     For $A_{jk}$ in $A_j$
6:         If $S(U_i,A_{jk}) \geq z_j$, add $U_i$ to $A_j$ & go to 2
7:         Else $s_{ij} = s_{ij} + S(U_i,A_{jk})$
8:         $s_{ij} = s_{ij}/\text{size}(A_j)$
9:     Sort $s_{ij}$'s in descending order
10:    If $\max(s_{ij}):j = 1 \& \& \max(s_{ij}) \geq t_1$
11:       Place $U_i$ into $A_1$
12:    Else if $\max(s_{ij}):j = 2 \& \& \max(s_{ij}) \geq t_2$
13:       Place $U_i$ into $A_2$
14:    Else if $\max(s_{ij}):j = 3 \& \& \max(s_{ij}) \geq t_3$
15:       Place $U_i$ into $A_3$
16:    Else if $\max(s_{ij}):j = 4 \& \& \max(s_{ij}) \geq t_4$
17:       Place $U_i$ into $A_4$
18:    Else
19:       Place $U_i$ into $A_5$

*Figure 7. Algorithm - Categorization Round 1*
However, it is observed that the most of the nouns were not classified at the end of these steps. The reason is that the initial sizes of the seed dictionaries were very small so the program could not find enough similarity between these seed terms and the uncategorized terms. Therefore, as part of the categorization round 1 two more steps are performed by using the same algorithm shown in Figure 7 in order to categorize more terms. This time the extended seed lists are used since using these seed lists can increase the chance of categorization for an uncategorized term. The algorithm in Figure 7 is first run with the similarity threshold of 0.75 for each of the aspect lists but the similarity scores are ignored (are kept as 1). The algorithm in Figure 7 is again run with the similarity threshold of 0.70 for each of the aspect lists but the similarity scores are ignored (are kept as 1). Aforementioned, these values are assigned according to the performance of the noun classification program written in Python programming language. Additionally, all of these processes are all automated within the system.

According to the observations, at the end of the initial categorization there were still a lot of nouns that are not classified in any of the aspects. At the same time, there were a lot of nouns that were misclassified especially in the ambiance and service aspect lists. For these aspects, the classification accuracy was observed to be less than 50%. To eliminate these misclassified nouns and place them into their correct seed lists a pruning step (categorization round 2) is performed.

4.4.2.2 Categorization Round-2 (Pruning Round)

The second round of the categorization is also called the pruning round, since the classification is done over the terms that are already classified in the first categorization
step. In other words, the second categorization is for justification purposes and eliminates (recategorizes) the terms that are classified in a wrong way. Pruning is crucial, since the lists that are achieved at the end of this process will be used as inputs in the next processes. Therefore, accuracy of aspect related term extraction is highly dependent on the accuracy of the lists that are output in this step.

Recall that the first classification was done over all the nouns in the review corpus. Unlike the first categorization, second classification (pruning) is performed separately on the food, service, price and ambiance lists that are created as the outputs of the first categorization round. This process is summarized in Figure 8.
The second categorization uses the algorithm in Figure 7 as same as the first categorization. The seed lists used in this round are the same as the ones used in the round one shown in Table 4. However, similarity thresholds used for each of the lists are different than the first categorization round. For the second round, similarity thresholds and similarity scores are defined separately for each of the input lists: food, service, price and ambiance. A significant point is that similarity thresholds are increased as in Table 8, since this time the aim is to recategorize the words that are already classified. In other words, the aim is to increase the accuracy as much as possible and it is observed that as the threshold values increase, the categorization accuracy increases while the number of terms that are categorized decreases. For this round our main concern is the accuracy rather than the number of terms that are classified. Additionally, this time the similarity score for each of the aspect lists differs as shown in Table 9. Again, these values are chosen by experimentation. Using these values, the first categorization step is performed again as part of the second round and the initial seed lists are expanded.
Table 8

*Similarity Thresholds Used in the Second Categorization Round*

<table>
<thead>
<tr>
<th>Aspect List ($A_i$) /Similarity Threshold ($t_i$)</th>
<th>Food ($t_1$)</th>
<th>Service ($t_2$)</th>
<th>Ambiance ($t_3$)</th>
<th>Price ($t_4$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Food List ($A_1$)</td>
<td>0.80</td>
<td>0.85</td>
<td>0.85</td>
<td>0.85</td>
</tr>
<tr>
<td>Service List ($A_2$)</td>
<td>0.85</td>
<td>0.80</td>
<td>0.85</td>
<td>0.85</td>
</tr>
<tr>
<td>Ambiance List ($A_3$)</td>
<td>0.85</td>
<td>0.85</td>
<td>0.80</td>
<td>0.85</td>
</tr>
<tr>
<td>Price List ($A_4$)</td>
<td>0.80</td>
<td>0.80</td>
<td>0.80</td>
<td>0.80</td>
</tr>
</tbody>
</table>

Table 9

*Similarity Scores Used in the Second Categorization Round*

<table>
<thead>
<tr>
<th>Aspect List ($A_i$) /Similarity Score ($z_i$)</th>
<th>Food ($z_1$)</th>
<th>Service ($z_2$)</th>
<th>Ambiance ($z_3$)</th>
<th>Price ($z_4$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Food List ($A_1$)</td>
<td>0.90</td>
<td>0.95</td>
<td>0.95</td>
<td>0.95</td>
</tr>
<tr>
<td>Service List ($A_2$)</td>
<td>0.90</td>
<td>0.90</td>
<td>0.90</td>
<td>0.90</td>
</tr>
<tr>
<td>Ambiance List ($A_3$)</td>
<td>0.90</td>
<td>0.90</td>
<td>0.90</td>
<td>0.90</td>
</tr>
<tr>
<td>Price List ($A_4$)</td>
<td>0.90</td>
<td>0.90</td>
<td>0.90</td>
<td>0.90</td>
</tr>
</tbody>
</table>

After the pruning process, most of the misclassified terms are eliminated and put into their corresponding aspect list. However, it is observed that almost 14 out of 16 of
the noun groups that are found in the review corpus are not yet classified. To classify those, another round of categorization is performed.

4.4.2.3 Categorization Round-3

In categorization round 3, rather than the initial seed lists (Table 4), food, service, price and ambiance lists that are created at the end of the second categorization (pruning) round are used as the seed lists. This round is performed only on the rest of the nouns that remained uncategorized after the first round of categorization, which corresponds to 14 groups of nouns, each having approximately 300 nouns. An overview of the process is shown in Figure 9.

![Flow Chart - Categorization Round 3](image)

*Figure 9. Flow Chart - Categorization Round 3*

Although the algorithm that is run in the third round (Figure 10) looks similar to the algorithm run in the first round (Figure 7), they are slightly different. Algorithm used in round one (Figure 7) runs over the uncategorized nouns list several times until the uncategorized nouns list satisfies the idle condition. However, round three runs for only one time over the uncategorized list to eliminate redundancy. Additionally, the similarity
between a category term and an uncategorized term is identified without considering the synsets of the uncategorized term. Recall that in round 1 the similarity between the two terms was found by considering both of the term’s synsets. As another difference, in this step the uncategorized nouns that do not have more than one occurrence in the review corpus are discarded since they may increase the redundancy.

1: For $U_i$ in $U$
2:   For $A_j$ in $A$
3:     $s_{ij} \rightarrow 0$
4:   For $A_{jk}$ in $A_j$
5:     If $S(U_i, A_{jk}) \geq 1$, add $U_i$ to $A_j$ & go to 1
6:     Else $s_{ij} = s_{ij} + S(U_i, A_{jk})$
7:     $s_{ij} = s_{ij}/\text{size}(A_j)$
8:   Sort $s_{ij}$’s in descending order
9:   If $\max(s_{ij}): j = 1 \& \& \max(s_{ij}) \geq t_1$
10:   Place $U_i$ into $A_1$
11:   Else if $\max(s_{ij}): j = 2 \& \& \max(s_{ij}) \geq t_2$
12:   Place $U_i$ into $A_2$
13:   Else if $\max(s_{ij}): j = 3 \& \& \max(s_{ij}) \geq t_3$
14:   Place $U_i$ into $A_3$
15:   Else if $\max(s_{ij}): j = 4 \& \& \max(s_{ij}) \geq t_4$
16:   Place $U_i$ into $A_4$
17:   Else
18:     Place $U_i$ into $A_5$

*Figure 10. Algorithm - Categorization Round 3*

In the third round, similarity thresholds are defined as 0.85 for each of the aspect lists and the similarity scores are defined as 1 as shown in Table 10. That is, the similarity scores are ignored. According to the observations, use of similarity scores decreases the accuracy of the classification performed in the third round, since the seed lists that are
used in round three contains more nouns than the initial seed lists used in round one. The pseudocode of this step is shown in Figure 10.

Table 10

Categorization Round 3 - Similarity Thresholds and Similarity Scores for Each of the Seed Lists

<table>
<thead>
<tr>
<th>Aspect Lists</th>
<th>Similarity Threshold ($t_i$)</th>
<th>Similarity Score ($z_i$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Food list ($A_1$)</td>
<td>0.85</td>
<td>1.00</td>
</tr>
<tr>
<td>2 Service list ($A_2$)</td>
<td>0.85</td>
<td>1.00</td>
</tr>
<tr>
<td>3 Price list ($A_3$)</td>
<td>0.85</td>
<td>1.00</td>
</tr>
<tr>
<td>4 Ambiance list ($A_4$)</td>
<td>0.85</td>
<td>1.00</td>
</tr>
</tbody>
</table>

4.4.2.4 Categorization Round-4 (Pruning Round)

After the third round, the same pruning round that is discussed in 4.4.2.2 is performed on the outputs of round three in order to get rid of the miscategorizations (Figure 11).
Threshold values and similarity scores that are chosen based on experiments for this pruning step are shown in Table 11 and Table 12 respectively.

**Table 11**

*Similarity Thresholds Used in the Fourth Categorization Round*

<table>
<thead>
<tr>
<th>Aspect List (A_i) /Similarity Threshold (t_i)</th>
<th>Food (t_1)</th>
<th>Service (t_2)</th>
<th>Ambiance (t_3)</th>
<th>Price (t_4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Food List (A_1)</td>
<td>0.90</td>
<td>0.97</td>
<td>0.97</td>
<td>0.97</td>
</tr>
<tr>
<td>Service List (A_2)</td>
<td>0.95</td>
<td>0.90</td>
<td>0.95</td>
<td>0.95</td>
</tr>
<tr>
<td>Ambiance List (A_3)</td>
<td>0.95</td>
<td>0.90</td>
<td>0.90</td>
<td>0.90</td>
</tr>
<tr>
<td>Price List (A_4)</td>
<td>0.90</td>
<td>0.95</td>
<td>0.95</td>
<td>0.90</td>
</tr>
</tbody>
</table>
Table 12

*Similarity Scores Used in the Fourth Categorization Round*

<table>
<thead>
<tr>
<th>Aspect List ($A_i$) /Similarity Score ($z_i$)</th>
<th>Food ($z_1$)</th>
<th>Service ($z_2$)</th>
<th>Ambiance ($z_3$)</th>
<th>Price ($z_4$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Food List ($A_1$)</td>
<td>0.90</td>
<td>0.97</td>
<td>0.97</td>
<td>0.97</td>
</tr>
<tr>
<td>Service List ($A_2$)</td>
<td>0.95</td>
<td>0.90</td>
<td>0.95</td>
<td>0.95</td>
</tr>
<tr>
<td>Ambiance List ($A_3$)</td>
<td>0.95</td>
<td>0.90</td>
<td>0.90</td>
<td>0.90</td>
</tr>
<tr>
<td>Price List ($A_4$)</td>
<td>0.90</td>
<td>0.95</td>
<td>0.95</td>
<td>0.90</td>
</tr>
</tbody>
</table>

4.4.2.5 Categorization Round-5

Categorization round 5 is the last step in the aspect related terms identification step. After this step the lists for each of the aspects are finalized. Example of the words found in each of these lists is shown in Appendix A. The overview of categorization round 5 is illustrated with Figure 12.

*Figure 12. Flow Chart - Categorization Round 5*
Round 5 is again similar but different than the other rounds. The main idea in this step is that the top five similarity scores between an uncategorized term and the terms in each of the seed lists are summed up. The averages of these sums are calculated for each of the seed lists.

The average value is defined as the similarity between the uncategorized term and the corresponding seed list. This average value calculation step is demonstrated considering only the food list on the flow chart shown in Figure 13. Each uncategorized term is placed into the list that has the highest similarity score, unless this highest score is below the similarity threshold. Similarity thresholds used in round 5 are given in Table 13. Similarity scores are ignored for round 5 as well, which means that they are set to 1.
Figure 13. Flow Chart – Similarity Assignment between the Food List and with the Next Noun in Uncategorized Nouns List
Table 13

*Categorization Round 5 - Similarity Thresholds and Similarity Scores for Each Seed List*

<table>
<thead>
<tr>
<th>Aspect Lists</th>
<th>Similarity threshold ($t_i$)</th>
<th>Similarity score ($z_i$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Food list ($A_1$)</td>
<td>0.70</td>
<td>1.00</td>
</tr>
<tr>
<td>2 Service list ($A_2$)</td>
<td>0.85</td>
<td>1.00</td>
</tr>
<tr>
<td>3 Price list ($A_3$)</td>
<td>0.65</td>
<td>1.00</td>
</tr>
<tr>
<td>4 Ambiance list ($A_4$)</td>
<td>0.75</td>
<td>1.00</td>
</tr>
</tbody>
</table>

The algorithm that is used during the categorization is shown by the pseudocode in Figure 14. Recall that $t_1, t_2, t_3$ and $t_4$ refer to the similarity thresholds for food, service, price and ambiance respectively. Additionally, let $U$ be the list that contains the uncategorized nouns shown by $U_i$ where $i$ denotes the $i^{th}$ noun and let $P_{ij}$ be the list that contains the top 5 similarity scores between the term $U_i$ and the seed list $A_j$. Let $s_{ij}$ be the average score of the top 5 similarity scores between the term $U_i$ and the seed list $A_j$.

For each $i$ there are four $j$, which are $s_{i1}, s_{i2}, s_{i3}$ and $s_{i4}$. Also, let $S$ be the similarity function between any two terms denoted by $S(x,y)$. The output of the $S$ function is the highest similarity score between $x$ and $y$. Finally, assume that minElement is a function that returns the smallest element in a list.
4.5 Aspect Based Segmentation

Aspect based segmentation aims to parse the reviews into the aspects they mention. Basically, the model tries to find the longest segments in the reviews that mention the same aspect of a restaurant. A review may contain more than one segment that refers to the same aspect, or it may not contain any segment for one or more of the aspects (Zhu et al., 2011). Two segmentation models, one of which makes use of Zhu et al.’s segmentation approach (2011), and our proposed segmentation model are both implemented with our system in order to make a comparison between two segmentation models. The implementation details of these two different models are discussed in this section.

```plaintext
1: For $U_i$ in $U$
2:   For $A_j$ in $A$
3:     $P_{ij} \rightarrow []$
4:   For $A_{jk}$ in $A_j$
5:     If $S(U_i, A_{jk}) \geq \min \text{Element}(P_{ij})$, replace $\min \text{Element}(P_{ij})$ with $S(U_i, A_{jk})$
6:     $s_{ij} \leftarrow \text{Average of values in } P_{ij}$
7:     Sort $s_{ij}$’s in $s_i$ in descending order
8:     If $\max(s_{ij}) \cdot j = 1 \& \& \max(s_{ij}) \geq t_1$
9:       Place $U_i$ into $A_1$
10:    Else if $\max(s_{ij}) \cdot j = 2 \& \& \max(s_{ij}) \geq t_2$
11:       Place $U_i$ into $A_2$
12:    Else if $\max(s_{ij}) \cdot j = 3 \& \& \max(s_{ij}) \geq t_3$
13:       Place $U_i$ into $A_3$
14:    Else if $\max(s_{ij}) \cdot j = 4 \& \& \max(s_{ij}) \geq t_4$
15:       Place $U_i$ into $A_4$
16:    Else
17:       Place $U_i$ into $A_5$
```

Figure 14. Algorithm - Categorization Round 5
4.5.1 Aspect Based Segmentation Model – MAS

Zhu et al. (2011) developed a segmentation model, which they called multi-aspect segmentation (MAS). Their model aims to identify multiple aspects in a review. Their segmentation model is adapted and integrated with the aspect related terms identification step discussed in 4.4. The algorithm that is used to implement the MAS approach is discussed below.

In the MAS approach, the initial step is to clean and prepare the reviews for the segmentation phase. For this purpose, reviews are parsed into the tokens. Each of the tokens is then cleaned. That is, if the token itself is not a type of punctuation, it is pruned from the punctuations that may be attached near it and put into its simplest form by using lemmatization. For example; the term like- would be transformed into the term like by removing the punctuation attached to it. However, if the token is found to be punctuation itself, only the removal of the blank space is applied. For example, the token “:” remains as it is. After the cleaning, each of the cleaned tokens is put together (separated by a tab) to construct the review text again. An example raw review instance is provided to illustrate the process.

Raw Review: “Really good steaks, a little pricey but good. Very friendly staff which always rocks!” (Yelp Academic Dataset, 2013)

According to the above example, the output achieved after the cleaning process is demonstrated below.

Cleaned Review: really good steak , a little pricey but good
. very friendly staff which always rock !
After that, the next step is to segment the reviews into their smallest pieces. The MAS algorithm (Zhu et al., 2011) starts with parsing each review into sentences. Then, each of the sentences is further parsed each time a comma or semicolon is encountered in the sentence. The process is illustrated below with the same example:

**Cleaned Review:** really good steak , a little pricey but good . very friendly staff which always rock !

**Initial Segments:**

Segment 1: really good steak ,

Segment 2: a little pricey but good .

Segment 3: very friendly staff which always rock !

At the end of these steps, for each review, a file which contains the initial pruned segments is created. The number of lines in each of these files refers to the number of initial segments in that review. For the above example the number of initial segments is three so the number of lines in the file that refers to this review is also three.

Each of these segments is assigned an aspect based on the tokens it contains as well as these token’s pre-identified categories. To achieve this, a score is assigned for each of the food, service, ambiance and price aspects for each and every segment. Assignment of the initial scores is performed according to the number of occurrences of the aspect related terms. If a segment is either the first or the last segment in the review, then it is assigned a score of 1 for being neutral as well, since it is observed that the first and the last segments in the reviews usually mention a general opinion about the restaurant. Considering this fact, the above mentioned neutral assignment to these
segments is performed. In Table 14, the expected score assignments of each segment are
given for the above example to illustrate the process.

Table 14

<table>
<thead>
<tr>
<th>Cleaned Segments</th>
<th>Food</th>
<th>Service</th>
<th>Ambiance</th>
<th>Price</th>
<th>Neutral</th>
</tr>
</thead>
<tbody>
<tr>
<td>really good steak ,</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>a little pricey but good .</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>very friendly staff which</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>always rock !</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For each of the segments, initial scores are sorted. With these sorted scores and
the segmentation rules (heuristic rules) that are pre-identified, segments are assigned to
their corresponding aspects. These pre-identified heuristic rules are given in the section
below:

1. If the highest scores are assigned both for being neutral and for another aspect, then
   segment aspect is assigned to that other aspect.

2. If the highest score is assigned for being neutral;
   2.1. If the segment is the first or the last segment in the review, the aspect of that
        segment remains neutral.
   2.2. If the conjunction before that segment does not indicate end of sentence, the
        aspect of that segment could be same with the aspect of the segment that comes
before it. Therefore, the segment should be assigned to the aspect of the segment that comes immediately before it.

2.3. If the conjunction before that segment indicates end of sentence, the segment aspect should remain neutral.

3. If all of the aspects including the neutral aspect have a score of 0;

3.1. If the previous segment does not indicate end of sentence, the aspect of the segment should be assigned to its previous segment’s aspect.

3.2. If the previous segment indicates end of sentence;

3.2.1. If the conjunction that connects the segment to its next segment does not indicate end of the sentence; the segment should be assigned to the aspect of its next segment.

3.2.2. If the conjunction that connects the segment to its next segment indicates end of the sentence; the segment should be assigned to be neutral.

4. If the highest score is unique and assigned either to food, service, price or ambiance aspects, the segment aspect should be assigned to the aspect that has the highest score.

5. If there is more than one aspect that has the same highest scores;

5.1. If the previous segment does not indicate end of sentence;

5.1.1. If previous segment’s aspect is among these aspects, the segment should be assigned to the previous segment’s aspect.

5.1.2. If previous segment’s aspect is not among these aspects, next segment should be considered.
5.1.2.1. If the conjunction that connects the segment to its next segment does not indicate end of the sentence, the segment should be assigned to the aspect of its next segment.

5.1.2.2. If the conjunction that connects the segment to its next segment indicates end of the sentence, the segment should be assigned to be neutral.

5.2. If the previous segment indicates end of a sentence;

5.2.1. If the conjunction between the segment and its next segment does not indicate end of sentence, the segment should be assigned to the aspect of its next segment.

5.2.2. If the conjunction between the segment and its next segment indicates end of sentence, the segment should be assigned to be neutral.

The process used in this segmentation step is described with the pseudocode shown in Figure 15. Assume that each of the initial segments is shown by \( U_i \) where \( i \) is the \( i^{th} \) segment; and \( p_i \) is a list that contains the scores of the \( i^{th} \) segment for each of the possible aspect \( j \). \( S \) is the function that assigns initial scores to segments for each of the aspects, while \( R \) is another function that does the final aspect assignment to a segment based on its initial scores and the pre-identified rules based on the previous and next segment information. Meanwhile, final aspect assignment of a segment \( i \) is shown by \( F_i \).
After the aspect assignment to each of these segments, the consecutive segments that have the same aspect are combined. At the end of the combination, the number of segments may be equal or less than the number of segments before the segment combination. Pseudocode for segment combination is shown in Figure 16. Note that \( S_a \) in Figure 16 refers to the final \( a^{th} \) segment and \( Segment(i) \) refers to the corresponding segment of \( F_i \).

\[
\begin{align*}
1: & \quad a = 0 \\
2: & \quad \text{For} \ F_i \ \text{in} \ F \\
3: & \quad \text{If} \ F_i == F_{i-1} \\
4: & \quad S_a = S_a + Segment(i) \\
5: & \quad \text{Else} \\
6: & \quad a = a + 1 \\
7: & \quad S_a = Segment(i)
\end{align*}
\]

*Figure 16. Algorithm - Segmentation Process (Continues)*

Aspect assignment for each of the segments and the segment combination process are illustrated in Table 15 and Table 16 respectively over the same example used in Table 14.
Table 15

*Initial Segments and Their Expected and Assigned Aspects*

<table>
<thead>
<tr>
<th>Segment</th>
<th>Assigned Aspect (According to Table 14)</th>
<th>Expected Aspect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Really good steaks,</td>
<td>food</td>
<td>food</td>
</tr>
<tr>
<td>a little pricey but good.</td>
<td>price</td>
<td>price/food</td>
</tr>
<tr>
<td>very friendly staff which always rocks!</td>
<td>service</td>
<td>service</td>
</tr>
</tbody>
</table>

Table 16

*Final Segments According to their Aspects*

<table>
<thead>
<tr>
<th>Segment</th>
<th>Aspect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Really good steaks,</td>
<td>food</td>
</tr>
<tr>
<td>a little pricey but good.</td>
<td>price</td>
</tr>
<tr>
<td>very friendly staff which always rocks!</td>
<td>service</td>
</tr>
</tbody>
</table>

4.5.2 Proposed Aspect Based Segmentation Model - IMAS

The proposed aspect based segmentation model is inspired by Zhu et al.’s (2011) model. The initial segmentation approach differs from the model developed by Zhu and his team. The proposed approach, named improved multi-aspect segmentation (IMAS), initially segments the reviews through the selected conjunctions and punctuations. It aims to have a better segmentation of reviews in terms of the aspects that they mention. The algorithm for the new segmentation model is discussed in this section.
Review sentences that are cleaned for 3.4.1 are also used in the proposed segmentation model. However, to create the smallest segment pieces, the proposed algorithm starts with parsing each cleaned review into the sentences. Then, each of the sentences is further segmented when a comma, colon, semicolon, ellipsis, or either of the conjunction such as “and” and “but” is encountered. With this approach the simplest segments are created. The process is illustrated with a simple example (Yelp Academic Dataset, 2013) below:

**Raw Review:** “This place is old but the food is good. The compliments aren’t that great...such as the basil and bean sprouts they give out. Their soy milk is made fresh and does not come from a Yeo's can.” (Yelp Academic Dataset, 2013)

**Cleaned Review:** this place is old but the food is good . the compliment aren't that great ... such as the basil and bean sprout they give out . their soy milk is made fresh and does not come from a yeo can.

**Initial Segments:**

Segment 1: this place is old but

Segment 2: the food is good .

Segment 3: the compliment aren't that great ...

Segment 4: such as the basil and

Segment 5: bean sprout they give out .

Segment 6: their soy milk is made fresh and

Segment 7: does not come from a yeo can .
Each of these initial noun based segments is assigned an aspect based on the tokens they contain and these token’s pre-identified categories. For each segment, a score is assigned for each of the food, service, ambiance and price aspects. By using the assigned scores and the pre-defined rules, final aspects are identified. This process is performed in the same way as it is performed for MAS model. After the segmentation step, the output for this specific example (Yelp Academic Dataset, 2013) should be like the one in Table 17 and Table 18. The results shown in Table 18 are the expected final outputs.

Table 17

*Segments and Their Associated Aspects*

<table>
<thead>
<tr>
<th>Segment</th>
<th>Aspect</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>This place is old but</em></td>
<td>ambiance</td>
</tr>
<tr>
<td>the food is good.</td>
<td>food</td>
</tr>
<tr>
<td><em>The compliments aren't that great...</em></td>
<td>food</td>
</tr>
<tr>
<td>such as the basil and</td>
<td>food</td>
</tr>
<tr>
<td><em>bean sprouts they give out.</em></td>
<td>food</td>
</tr>
<tr>
<td><em>Their soy milk is made fresh and</em></td>
<td>food</td>
</tr>
<tr>
<td><em>does not come from Yeo’s can.</em></td>
<td>food</td>
</tr>
</tbody>
</table>
### Table 18

**Final Segments According to their Aspects**

<table>
<thead>
<tr>
<th>Segment</th>
<th>Aspect</th>
</tr>
</thead>
<tbody>
<tr>
<td>This place is old but</td>
<td>ambiance</td>
</tr>
<tr>
<td>the food is good. The compliments aren't that great...such as the basil and bean sprouts they give out. Their soy milk is made fresh and does not come from Yeo’s can.</td>
<td>food</td>
</tr>
</tbody>
</table>

#### 4.6 Opinion Identification of each Aspect-Based Segment

In order to extract the feature-opinion pairs from the reviews, a final step that assigns opinions to those identified features is performed. Basically, the opinion identification step assigns opinions to each of the final segments created in step 4.5 since each of these segments mentions different aspect. We assume that each final segment can only have one associated opinion with it since for most of the reviews this is the case. The methodology that is used in the opinion identification step is discussed in this section.

It is observed that most of the reviews in the dataset that have 1 or 2 stars are completely negative oriented while the reviews that have 5 stars are completely positive oriented. Findings of Fan et al. (2014) on the same Yelp Dataset was a trigger for doing an observational analysis on our review corpus. As the observations suggest, it is...
assumed that the aspects mentioned by the reviews that have 1 or 2 stars are all negative; while the aspects mentioned by the reviews that have 5 stars are all positive.

However, the reviews that have 3 or 4 stars constitute almost half of our dataset as shown in Figure 17. For those reviews, SentiWordNet 3.0 library is utilized. SentiWordNet is a well-known, readily available library that is in the form of a .txt file which contains some of the words and their sentiment scores for being negative or positive (Baccianella et al., 2010). The implementation of SentiWordNet is based on pre-identified positive and negative seed lists and their extension by using WordNet library (Baccianella et al., 2010; Miller, 1995). Recall that WordNet library was also used in our system for the aspect related term identification step.

*Figure 17. Distribution of Review Stars in the Review Corpus*
Each of the final segments detailed in section 4.5, is assigned an opinion based on the words they contain. However, not all of the words in these segments are considered. The words that are not needed during the opinion identification step are removed from those final segments and saved in different folders. Only the adjectives, conjunctions such as “and” and “but” and some of the verbs such as “like”, “love” and “hate” are kept. Additionally, the words like “not”, “dont”, “arent”, “wasnt”, “isnt”, “doesnt”, “werent”, “hasnt”, “havent”, “n’t” and “no” that may indicate contradiction are also kept in these segments. The reason is that the words “like”, “love”, “hate”, “not”, “dont”, “arent”, “wasnt”, “isnt”, “doesnt”, “werent”, “hasnt”, “havent”, “n’t” and “no” do not have an entry in SentiWordNet library. However, it is assumed that “like” and “love” indicate positive opinion; while “hate” indicates negative opinion. It is also assumed that contradiction words change the opinion of the adjective that comes after it (Marrese-Taylor et al., 2013). That is, if the adjective or the opinion verb is positive, the contradiction that comes before it makes the adjective negative. Again, if the adjective or the opinion verb is negative, the contradiction that comes before it makes the adjective positive.

Opinion polarity of an adjective is assigned based on its SentiWordNet score. For an adjective, only the scores (either positivity or negativity score) that are over 0.50 are considered. The reason is that if an adjective has negativity or positivity score over 0.50, it is likely to be negative or positive respectively. On the other hand, if the negativity or positivity score is less than 0.50 for an adjective, one cannot assume that it is negative or positive respectively. Therefore, if an adjective has more than one entry in the
SentiWordNet dictionary, the average value of its scores that are over 0.50 is calculated. If there is not any score entry over 0.50 for an adjective, it is assigned to be neutral. If the negativity score is higher than the positivity score of an adjective, the adjective is assigned to be negative and increases the negativity score of the segment in which it resides by one. If the positivity score is higher than the negativity score, the adjective is assigned to be positive and increases the positivity score of the segment it resides in by one.

Not only the adjectives but also the conjunctions that come before the final segments are also taken into consideration during the opinion identification. It is observed that when “but” comes before a segment, it creates a contradiction; while “and” and “,” do not (Marrese-Taylor et al., 2013). Therefore, if the conjunction that comes before the segment is “but” and if the previous segment is found to be positive, the negativity score for the current segment is increased by one. If the previous segment is found to be negative, the positivity score for the current segment is increased by one. On the other hand, if the conjunction that comes before the segment is “and” or “,”, the same sentiment dominant in the previous segment’s is increased by one for the current segment. By considering all of these, opinion assignment process of a segment is demonstrated with the flow chart shown in Figure 18.
Figure 18. Flow Chart - Opinion Assignment to a Segment
Based on the positivity and negativity scores achieved at the end of these steps, the segments are assigned an opinion for being positive, negative or neutral. If there is more than one segment that mentions same aspect but different opinions, the dominant opinion is assigned to that aspect. If negativity and positivity scores are identical, the opinion of that aspect is assigned to be positive. Please note that if there are two different opinions for the same aspect within a review, the dominant opinion is assigned to the aspect. An example review data instance (Yelp Academic Dataset, 2013) is processed in Table 19 to illustrate the relation of opinions with the segments and their assigned aspects.

Table 19

*Expected Segments, Aspects and Their Associated Opinions*

<table>
<thead>
<tr>
<th>Segment</th>
<th>Aspect</th>
<th>Opinion</th>
</tr>
</thead>
<tbody>
<tr>
<td>This place is old but the food is good. The compliments arent that great...such as the basil and bean sprouts they give out. Their soy milk is made fresh and does not come from Yeo’s can.</td>
<td>ambiance</td>
<td>NEG</td>
</tr>
<tr>
<td></td>
<td>food</td>
<td>POS</td>
</tr>
</tbody>
</table>
5. EXPERIMENTATION AND RESULTS

In this chapter, the results that are achieved after the aspect related term extraction, aspect based segmentation and opinion identification steps are evaluated and the findings are discussed. Complete system accuracies are compared using both of the systems which are developed with Zhu et al.’s segmentation approach and the proposed segmentation approach respectively.

5.1 Preparation of Test Data

There are three main test data sets created for each of the three system components. Recall that there are 100 data instances that were separated for testing purposes. Testing of the three steps is achieved with these 100 data instances.

The first test data is created for evaluating the performance of the aspect related term identification step. For all of the test data 100 instances, 100 files are created, each of which contains the nouns in the corresponding reviews. For example; for the first test data instance, a file which contains the nouns in the review that corresponds to that data instance is created. Noun extraction from the reviews is achieved by the POS tagging functionality of NLTK (Loper & Bird, 2002). Each of these nouns in each of these created files is tagged manually either as food, service, price, ambiance or neutral according to the implied meanings in their corresponding review. The tagging process is demonstrated in Table 20 on the raw review shown below (Yelp Academic Dataset, 2013).
Raw Review: “This place is old but the food is good. The compliments aren't that great...such as the basil and bean sprouts they give out. Their soy milk is made fresh and does not come from a Yeo's can.” (Yelp Academic Dataset, 2013).

Table 20

*Tagged Nouns According to Their Aspects*

<table>
<thead>
<tr>
<th>NOUN</th>
<th>ASPECT TAG</th>
</tr>
</thead>
<tbody>
<tr>
<td>place</td>
<td>Ambiance</td>
</tr>
<tr>
<td>food</td>
<td>Food</td>
</tr>
<tr>
<td>compliment</td>
<td>Food</td>
</tr>
<tr>
<td>basil</td>
<td>Food</td>
</tr>
<tr>
<td>bean</td>
<td>Food</td>
</tr>
<tr>
<td>sprout</td>
<td>Food</td>
</tr>
<tr>
<td>soy</td>
<td>Food</td>
</tr>
<tr>
<td>milk</td>
<td>Food</td>
</tr>
<tr>
<td>yeo</td>
<td>Neutral</td>
</tr>
</tbody>
</table>

The other test data is created for the segmentation phase. There are two types of segmentation that were implemented: base segmentation model developed by Zhu et al. (2011) and the improved model based on Zhu et al.’s (2011) segmentation model. For both of them, the minimal segments generated in the segmentation phase are used. For each of the test data instances, these segments are put in files where each line refers to one segment. Since segmentation methods have different outputs, two separate test data
sets are created for each segmentation type. After that, each of these segments for each of
the reviews is tagged according to their implied aspects. For the above example (Yelp
Academic Dataset, 2013), the segment’s test data is provided in Table 21 and Table 22.

Table 21
*MAS - Segments and Their Expected Aspects*

<table>
<thead>
<tr>
<th>SEGMENT</th>
<th>TAGGED ASPECT</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>This place is old but the food is good.</em></td>
<td>Ambiance/Food</td>
</tr>
<tr>
<td><em>The compliment aren't that great... such the basil and bean sprout they give out.</em></td>
<td>Food</td>
</tr>
<tr>
<td><em>Their soy milk is made fresh and does not come from Yeo’s can.</em></td>
<td>Food</td>
</tr>
</tbody>
</table>

Table 22
*IMAS - Segments and Their Expected Aspects*

<table>
<thead>
<tr>
<th>SEGMENT</th>
<th>TAGGED ASPECT</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>This place is old but the food is good.</em></td>
<td>Ambiance</td>
</tr>
<tr>
<td><em>the food is good.</em></td>
<td>Food</td>
</tr>
<tr>
<td><em>The compliment aren't that great... such the basil and bean sprout they give out.</em></td>
<td>Food</td>
</tr>
<tr>
<td><em>Their soy milk is made fresh and does not come from Yeo’s can.</em></td>
<td>Food</td>
</tr>
</tbody>
</table>

Final test data is created for testing the complete system after the opinion identification of finalized segments. A single file is created which contains the ID’s of the 100 test data instances and opinion tags for food, service, price and ambiance aspects. The segments in each review and their associated opinion are assigned manually again. Opinions can take positive (POS), negative (NEG) and neutral (NEU) values. In the test data set, the line that corresponds to above example is shown below. If an aspect does not have a corresponding opinion, it means that the aspect is not mentioned in that review. Additionally, if an aspect implies more than one opinion within the same review, it means that the aspect mentions the dominant opinion.

1 ambiance: NEG food: POS service: -- price: --

5.2 Results of Aspect Related Term Identification Step

The aspect related term identification step is evaluated by the average precision, average recall and the F1 score that is calculated by using the average precision and recall scores. Average precision and average recall values are calculated by summing up the precision and recall values of all 100 data instances respectively and dividing these sums by the total number of data instances, which is 100.

Individual precision and recall scores are calculated for each review (data) instance. Precision is defined as the number of correctly categorized nouns over the total number of nouns that are retrieved from the corresponding review. On the other hand, recall is defined as the number of correctly categorized nouns over the total number of
nouns that are correctly assigned an aspect. According to the Table 23 precision and recall formulations used for evaluation are shown below.

Table 23

**Precision & Recall Matrix**

<table>
<thead>
<tr>
<th>True Aspect Assignment</th>
<th>False Aspect Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>True Positive (TP)</td>
<td>False Positive (FP)</td>
</tr>
<tr>
<td>False Negative (FN)</td>
<td>True Negative (TN)</td>
</tr>
</tbody>
</table>

**Precision** = \( \frac{TP}{TP+FP} \)  \hspace{1cm} (6.2.1)  

**Recall** = \( \frac{TP}{TP+FN} \)  \hspace{1cm} (6.2.2)

An example that illustrates the use of precision and recall measures in the aspect related term identification step is provided below.

**ReviewID: 40**


dennys: true positive
chain: true positive
restaurant: false positive
life: true positive
chain: true positive
breakfast: false negative
**egg**: *true positive*

**Precision = 5/6   Recall = 5/6**

For the aspect identification step, average precision is found as 0.63, while the average recall is found as 0.93. As mentioned before, recall is basically the probability of the correctly retrieved nouns over all the correctly categorized nouns (whether they are retrieved or not retrieved). Since recall does not deal only the retrieved documents, it is not surprising to achieve a score of 0.93. The precision value is more important than the recall value in the aspect identification step. However, evaluating these two metrics as one significant value provides a better understanding for our analysis. For this reason, F1 is calculated and found as 0.75 according to the formulation shown below.

\[
F1 = \frac{2\times P\times R}{P+R} \tag{6.2.3}
\]

In text mining, one would not expect values more than 90% since a noun may be classified as food in one sentence while the same noun may be classified as service in another sentence. An F1 value of 0.75 indicates that the most of the assigned aspects for each of the nouns are correct and can be used in the next processes.

5.3 Results of Aspect Based Segmentation

The segmentation step is evaluated by the accuracy measure. Accuracy measures are expected to be close to each other for both of the segmentation models. The reason is that the accuracy is defined as the fraction of the total number of correct aspect assignments to the segments over the total number of segments. One might expect that the number of correct assignments would be higher in the new segmentation model. Although this is the case, the total number of segments is higher in the new segmentation
model as well. Therefore, the accuracy measures for both of the segment types approach to each other. The accuracy measure that is used during the segmentation phase is defined as below.

Let $S_n$ be the number of segments in $n^{th}$ review and let $a_k$ be a constant value that denotes the correctness of an aspect assignment to the $k^{th}$ segment in a review. That is, $a_k$ can take value of 0 if the segment is not assigned to a correct aspect; and take value of 1 if the segment is assigned to a correct aspect.

$$a_k = \begin{cases} 
1, & \text{if the } k^{th} \text{ segment is correctly classified} \\
0, & \text{if the } k^{th} \text{ segment is not correctly classified} 
\end{cases}$$  \hspace{1cm} (6.3.1)

$$\text{Accuracy} = \frac{\sum_{n=1}^{S_n} \sum_{k=1}^{S_n} a_k}{\sum_{n=1}^{S_n} S_n}$$  \hspace{1cm} (6.3.2)

Given the accuracy measures, test results showed that the old segmentation model achieved 50% segmentation accuracy whereas the new one achieved 51% segmentation accuracy. Recall that as the last step of the segmentation phase the consecutive segments that mention the same aspect were combined. However, the comparison, which is done to measure the accuracy of the aspect based segmentation step, is performed over the minimal segments, not over the combined segments. The reason is that it is almost impossible to achieve identical segment combinations. The situation is illustrated with the new segmentation model over a review data instance (Yelp Academic Dataset, 2013) shown below.

Raw review: “3.5 stars Fast friendly service and the food was good. I ordered Hung Pao rice bowl, while it was not huge and loaded with things it was plenty for a lunch size. The food tasted good. We ordered it Sunday evening for dinner. Orange chicken, General Tso
chicken and chicken fried rice. ALL with white meat. While I really like the fact you can get all white meat, it was a little dry. Perhaps the way it was sliced and not in small cubes? All the food was good but not the best I have had. I would order again.” (Yelp Academic Dataset, 2013)

In Table 24, actual aspect taggings of the initial segments are provided for this data instance (review). The following Table 25 shows the combined (final) segments for the same data instance. On the other hand, Table 26 and Table 27 show the results that were achieved after running our new aspect assignment model on the same example. The evaluation is based on the segments that were tagged manually and the ones achieved by the segmentation model. When the initial segments are used for the evaluation, a comparison can be performed between the tagged and the test data. In this case, the accuracy is 0.67 for this example review. However, if we use the combined segments, we cannot make a one-to-one comparison since the number of final segments in tagged and test data are not the same, as it is shown in Table 24 and Table 26.
Table 24

Manually Tagged Data Instance – Initial Segmentation

<table>
<thead>
<tr>
<th>Segment</th>
<th>Aspect</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5 stars Fast friendly service and</td>
<td>service</td>
</tr>
<tr>
<td>the food was good.</td>
<td>food</td>
</tr>
<tr>
<td>I ordered Hung Pao rice bowl,</td>
<td>food</td>
</tr>
<tr>
<td>while it was not huge and</td>
<td>food</td>
</tr>
<tr>
<td>loaded with things it was plenty for a lunch size.</td>
<td>food</td>
</tr>
<tr>
<td>The food tasted good.</td>
<td>food</td>
</tr>
<tr>
<td>We ordered it Sunday evening for dinner</td>
<td>food</td>
</tr>
<tr>
<td>Orange chicken,</td>
<td>food</td>
</tr>
<tr>
<td>General Tso chicken and</td>
<td>food</td>
</tr>
<tr>
<td>chicken fried rice.</td>
<td>food</td>
</tr>
<tr>
<td>ALL with white meat.</td>
<td>food</td>
</tr>
<tr>
<td>While I really like the fact you can get all white meat,</td>
<td>food</td>
</tr>
<tr>
<td>it was a little dry.</td>
<td>food</td>
</tr>
<tr>
<td>Perhaps the way it was sliced and</td>
<td>food</td>
</tr>
<tr>
<td>not in small cubes?</td>
<td>food</td>
</tr>
<tr>
<td>All the food was good but</td>
<td>food</td>
</tr>
<tr>
<td>not the best I have had.</td>
<td>food</td>
</tr>
<tr>
<td>I would order again.</td>
<td>food</td>
</tr>
</tbody>
</table>
### Table 25

**Manually Tagged Data Instance – Final Segmentation**

<table>
<thead>
<tr>
<th>Segment</th>
<th>Aspect</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5 stars Fast friendly service and the food was good. I ordered Hung Pao rice bowl, while it was not huge and loaded with things it was plenty for a lunch size. The food tasted good. We ordered it Sunday evening for dinner. Orange chicken, General Tso chicken and chicken fried rice. ALL with white meat. While I really like the fact you can get all white meat, it was a little dry. Perhaps the way it was sliced and not in small cubes? All the food was good but not the best I have had. I would order again.</td>
<td>service</td>
</tr>
</tbody>
</table>
Table 26

Test Data Instance – Initial Segmentation

<table>
<thead>
<tr>
<th>Segment</th>
<th>Aspect</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5 stars Fast friendly service and</td>
<td>service</td>
</tr>
<tr>
<td>the food was good.</td>
<td>food</td>
</tr>
<tr>
<td>I ordered Hung Pao rice bowl,</td>
<td>ambiance</td>
</tr>
<tr>
<td>while it was not huge and</td>
<td>ambiance</td>
</tr>
<tr>
<td>loaded with things it was plenty for a lunch size.</td>
<td>ambiance</td>
</tr>
<tr>
<td>The food tasted good.</td>
<td>food</td>
</tr>
<tr>
<td>We ordered it Sunday evening for dinner</td>
<td>food</td>
</tr>
<tr>
<td>Orange chicken,</td>
<td>food</td>
</tr>
<tr>
<td>General Tso chicken and</td>
<td>food</td>
</tr>
<tr>
<td>chicken fried rice.</td>
<td>food</td>
</tr>
<tr>
<td>ALL with white meat.</td>
<td>food</td>
</tr>
<tr>
<td>While I really like the fact you can get all white meat,</td>
<td>food</td>
</tr>
<tr>
<td>it was a little dry.</td>
<td>ambiance</td>
</tr>
<tr>
<td>Perhaps the way it was sliced and</td>
<td>ambiance</td>
</tr>
<tr>
<td>not in small cubes?</td>
<td>food</td>
</tr>
<tr>
<td>All the food was good but</td>
<td>food</td>
</tr>
<tr>
<td>not the best I have had.</td>
<td>food</td>
</tr>
<tr>
<td>I would order again.</td>
<td>ambiance</td>
</tr>
</tbody>
</table>
### Table 27

**Test Data Instance – Final Segmentation**

<table>
<thead>
<tr>
<th>Segment</th>
<th>Aspect</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5 stars Fast friendly service and the food was good.</td>
<td>service</td>
</tr>
<tr>
<td>I ordered Hung Pao rice bowl, while it was not huge and loaded with things it was plenty for a lunch size.</td>
<td>ambiance</td>
</tr>
<tr>
<td>The food tasted good. We ordered it Sunday evening for dinner. Orange chicken, General Tso chicken and chicken fried rice. ALL with white meat. While I really like the fact you can get all white meat,</td>
<td>food</td>
</tr>
<tr>
<td>it was a little dry. Perhaps the way it was sliced and</td>
<td>ambiance</td>
</tr>
<tr>
<td>not in small cubes? All the food was good but not the best I have had.</td>
<td>food</td>
</tr>
<tr>
<td>I would order again.</td>
<td>ambiance</td>
</tr>
</tbody>
</table>

It is important to note that the accuracy of the aspects that are retrieved at the end of the segmentation step is more important than their ordering in the reviews, because we are only interested in the effects of the new segmentation model on the overall aspect–opinion pair extraction task. For example, in the tagged data instance (Table 24), food and service aspects actually exist; whereas in the test data instance (Table 26) the segmentation model retrieved food, service and also ambiance aspects. Therefore, if we make a comparison based on the aspects they retrieve, we achieve 75% accuracy since we also need to consider the price aspect that should not be retrieved. However, this type of evaluation will be utilized during the evaluation of the complete system.
5.4 Overall Results Achieved After the Identification of Opinions

In the aspect based segmentation step, there were two models that were tested on the same restaurant data. In the overall system evaluation, the results achieved after the opinion identification step is evaluated over the two different systems which has two different segmentation models as shown in Figure 19. During the evaluation, accuracy metric is utilized. Examples from the experimentation results is given below. The last line under each review block shows the total number of correct categorizations for each category at the time of the evaluation of that review. Total number of correct categorizations for each category cannot exceed 100 since there are 100 reviews that are used as test data instances.

ReviewID: 97
Tagged: {'food': 'POS', 'price': '', 'ambiance': '', 'service': ''}
Tested: {'food': 'POS', 'price': '', 'ambiance': '', 'service': ''}
Food Accr: 59  Price Accr: 77  Service Accr: 70  Ambiance Accr: 51

ReviewID: 98
Tagged: {'food': 'NEG', 'price': '', 'ambiance': '', 'service': ''}
Tested: {'food': 'NEG', 'price': '', 'ambiance': 'NEU', 'service': 'NEG'}
Food Accr: 60  Price Accr: 78  Service Accr: 70  Ambiance Accr: 51

Figure 19. Overall System Evaluation
5.4.1 Results – Old Model

By adapting the old segmentation model, the system achieved overall accuracy of 0.61. According to the results, food accuracy is found to be 0.57; price is found to be 0.77; service is found to be 0.61 and ambiance is found to be 0.50.

5.4.2 Results – New Model

By using the new segmentation model, the system achieved overall accuracy of 0.66. According to the results, food accuracy is found to be 0.61; price is found to be 0.79; service is found to be 0.71 and ambiance is found to be 0.52.

Overall results on the Yelp Dataset as shown in Table 28 indicates that use of the new segmentation model (IMAS) improved the aspect-opinion extraction accuracy achieved by using the MAS model by 8.2%. Examples of the reviews and the aspect-opinion pairs extracted from those reviews by using IMAS can be found in Appendix B.

Table 28

<table>
<thead>
<tr>
<th>Models</th>
<th>Food</th>
<th>Price</th>
<th>Service</th>
<th>Ambiance</th>
<th>Overall Evaluation (Aspect-Opinion Pairs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAS</td>
<td>0.57</td>
<td>0.77</td>
<td>0.61</td>
<td>0.50</td>
<td>0.61</td>
</tr>
<tr>
<td>IMAS</td>
<td>0.61</td>
<td>0.79</td>
<td>0.71</td>
<td>0.52</td>
<td>0.66</td>
</tr>
</tbody>
</table>
6. CONCLUSIONS AND FUTURE WORK

This research aims to contribute to the text summarization applications, which are considered under text mining. The major intent of this thesis was to improve the accuracy of the extraction of the aspect-opinion pairs by improving the underlying segmentation model, which is considered the most important part of a multi-aspect opinion extraction system.

In the aspect-opinion extraction system proposed in this thesis, there were three steps which are aspect related term identification, aspect-based segmentation and opinion identification of the identified segments. The implementation of the system is performed in Python programming language. Aspect-related term identification step was based on the similarity function provided by the WordNet Library (Miller, 1995) and term frequencies, whereas the segmentation was based on the heuristic rules. On the other hand, opinion identification step was performed by use of the SentiWordNet Library (Baccianella et al., 2010) and heuristic rules. Segmentation methodology, improved multi-aspect segmentation (IMAS), is considered the main contribution to the literature of this research. IMAS is also found to be the most significant step since the segmentation allows the system to identify multiple aspects that are found within a review.

The segmentation model, multi-aspect segmentation (MAS) developed by Zhu et al. (2011), was the major influence for this research. The MAS model, together with the aspect-related term identification and sentiment analysis steps, were used on Chinese restaurant reviews by Zhu and his team. On the other hand, the similar methodology used in this thesis was implemented on English restaurant reviews. To compare effects of the
two different segmentation models, MAS and IMAS, two different systems were developed in parallel. Both of the systems consist of three steps, as stated above: aspect-related term identification; segmentation; and opinion identification. The comparison of the two systems is performed on the restaurant reviews retrieved from the Yelp Academic Dataset. On this dataset, the overall accuracy of the system developed by adapting MAS model is found to be 0.61, while the overall accuracy of the system developed using IMAS model is found to be 0.66. As a result, developed IMAS model performed 8.2% better than the MAS model. In IMAS model, the motivation was that people usually mention different aspects of a restaurant when they use conjunctions like “and” and “but” in addition to the use of comma and end-of-sentence marks. Experimentation results proved this argument too by achieving improved accuracy for IMAS model on English restaurant reviews.

One of the most important limitations of this research is that the proposed system is not domain-independent since the aspect-related term extraction step is developed by considering a restaurant’s domain. Therefore, in the future, the overall system can further be utilized in different domains like electronic products and hotel reviews in order to summarize the customer reviews or extract the feature-opinion pairs from the reviews. Furthermore, since the manual tagging of data takes time, supervised approaches were not used during the aspect related term identification and opinion identification phases. However, these phases can also be improved by using supervised approaches and may lead to an improved overall accuracy. For example, artificial neural networks can be used while identifying the aspect related terms or classifying the opinions of the segments if
there is a significant number of tagged data instances available. The final limitation of this study is the limited capability of the readily available libraries such as WordNet and SentiWordNet which are used in the implementation of the system. As these libraries acquire superior capabilities in the future, the system may achieve better results. To the best of our knowledge, the experts in natural language processing are still working on improving these libraries.

In conclusion, extracting important information from customer reviews is among today’s hot topics, and needs more research to achieve faster and better analysis. Usually a tradeoff analysis between time and accuracy has to be done while developing text-mining systems. In this research, time factors played an important role, and our aim was to make a comparison with an existing unsupervised model, which led us to use an unsupervised methodology. Overall, this research contributed to the text mining area with the proposed aspect-opinion extraction system, and it could be further improved for better performance.
REFERENCES


APPENDIX A: EXAMPLES FROM FINAL ASPECT DICTIONARIES

<table>
<thead>
<tr>
<th>Ambiance</th>
<th>atmosphere</th>
<th>dish</th>
<th>woman</th>
<th>hanging</th>
<th>idea</th>
<th>section</th>
<th>job</th>
</tr>
</thead>
<tbody>
<tr>
<td>ambiance</td>
<td>hell</td>
<td>practice</td>
<td>look</td>
<td>go</td>
<td>market</td>
<td>protection</td>
<td></td>
</tr>
<tr>
<td>concept</td>
<td>layout</td>
<td>game</td>
<td>try</td>
<td>hash</td>
<td>order</td>
<td>training</td>
<td></td>
</tr>
<tr>
<td>activity</td>
<td>fit</td>
<td>outlet</td>
<td>bag</td>
<td>operation</td>
<td>run</td>
<td>beginning</td>
<td></td>
</tr>
<tr>
<td>design</td>
<td>sky</td>
<td>appearance</td>
<td>view</td>
<td>theory</td>
<td>business</td>
<td>search</td>
<td></td>
</tr>
<tr>
<td>decor</td>
<td>location</td>
<td>trouble</td>
<td>set</td>
<td>category</td>
<td>use</td>
<td>feeling</td>
<td></td>
</tr>
<tr>
<td>fun</td>
<td>playing</td>
<td>building</td>
<td>creation</td>
<td>humor</td>
<td>possibility</td>
<td>career</td>
<td></td>
</tr>
<tr>
<td>industry</td>
<td>walk</td>
<td>hood</td>
<td>store</td>
<td>study</td>
<td>smell</td>
<td>play</td>
<td></td>
</tr>
<tr>
<td>thing</td>
<td>art</td>
<td>music</td>
<td>bladder</td>
<td>marketplace</td>
<td>object</td>
<td>shop</td>
<td></td>
</tr>
<tr>
<td>place</td>
<td>hang</td>
<td>way</td>
<td>property</td>
<td>fashion</td>
<td>painting</td>
<td>office</td>
<td></td>
</tr>
<tr>
<td>Food</td>
<td>sugar</td>
<td>grape</td>
<td>cooler</td>
<td>espresso</td>
<td>berry</td>
<td>beet</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>-------</td>
<td>-------</td>
<td>--------</td>
<td>----------</td>
<td>-------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>drink</td>
<td>shake</td>
<td>avocado</td>
<td>cider</td>
<td>tiramisu</td>
<td>cocoa</td>
<td>kiwi</td>
<td></td>
</tr>
<tr>
<td>sweet</td>
<td>milk</td>
<td>dumpling</td>
<td>gum</td>
<td>mate</td>
<td>lettuce</td>
<td>medium</td>
<td></td>
</tr>
<tr>
<td>fruit</td>
<td>sashimi</td>
<td>celery</td>
<td>legume</td>
<td>plantain</td>
<td>oregano</td>
<td>mushroom</td>
<td></td>
</tr>
<tr>
<td>vegetable</td>
<td>toast</td>
<td>fig</td>
<td>rosemary</td>
<td>herb</td>
<td>java</td>
<td>squash</td>
<td></td>
</tr>
<tr>
<td>serving</td>
<td>soup</td>
<td>savory</td>
<td>tea</td>
<td>water</td>
<td>spinach</td>
<td>chocolate</td>
<td></td>
</tr>
<tr>
<td>sangria</td>
<td>bread</td>
<td>pumpkin</td>
<td>candy</td>
<td>compote</td>
<td>taste</td>
<td>olive</td>
<td></td>
</tr>
<tr>
<td>course</td>
<td>cabbage</td>
<td>hip</td>
<td>lentil</td>
<td>onion</td>
<td>draft</td>
<td>flan</td>
<td></td>
</tr>
<tr>
<td>stuff</td>
<td>menu</td>
<td>coffee</td>
<td>okra</td>
<td>mixer</td>
<td>treat</td>
<td>malt</td>
<td></td>
</tr>
<tr>
<td>fennel</td>
<td>cilantro</td>
<td>thyme</td>
<td>truffle</td>
<td>milkshake</td>
<td>basil</td>
<td>tree</td>
<td></td>
</tr>
<tr>
<td>Service</td>
<td>clock</td>
<td>thanks</td>
<td>plant</td>
<td>accommodation</td>
<td>side</td>
<td>rest</td>
<td>box</td>
</tr>
<tr>
<td>------------------</td>
<td>----------</td>
<td>--------</td>
<td>--------</td>
<td>---------------</td>
<td>--------</td>
<td>-------</td>
<td>-----</td>
</tr>
<tr>
<td>waiter</td>
<td>offense</td>
<td>hotel</td>
<td>presentation</td>
<td>spin</td>
<td>power</td>
<td>type</td>
<td></td>
</tr>
<tr>
<td>attitude</td>
<td>court</td>
<td>person</td>
<td>pleasure</td>
<td>pick</td>
<td>stand</td>
<td>staff</td>
<td></td>
</tr>
<tr>
<td>server</td>
<td>service</td>
<td>female</td>
<td>support</td>
<td>break</td>
<td>life</td>
<td>feature</td>
<td></td>
</tr>
<tr>
<td>assistance</td>
<td>comfort</td>
<td>utensil</td>
<td>waitress</td>
<td>term</td>
<td>field</td>
<td>machine</td>
<td></td>
</tr>
<tr>
<td>behavior</td>
<td>resort</td>
<td>pose</td>
<td>heavy</td>
<td>selection</td>
<td>approach</td>
<td>bank</td>
<td></td>
</tr>
<tr>
<td>owner</td>
<td>alarm</td>
<td>pitcher</td>
<td>shot</td>
<td>culture</td>
<td>mill</td>
<td>help</td>
<td></td>
</tr>
<tr>
<td>manner</td>
<td>hour</td>
<td>hand</td>
<td>back</td>
<td>reading</td>
<td>convenience</td>
<td>fixture</td>
<td></td>
</tr>
<tr>
<td>player</td>
<td>time</td>
<td>post</td>
<td>chair</td>
<td>cut</td>
<td>setup</td>
<td>seat</td>
<td></td>
</tr>
<tr>
<td>computer</td>
<td>minute</td>
<td>host</td>
<td>brewery</td>
<td>raise</td>
<td>second</td>
<td>watch</td>
<td></td>
</tr>
<tr>
<td>Price</td>
<td>tab</td>
<td>pile</td>
<td>overcharge</td>
<td>ticket</td>
<td>drag</td>
<td>prize</td>
<td></td>
</tr>
<tr>
<td>-----------------------</td>
<td>------</td>
<td>-------</td>
<td>------------</td>
<td>--------</td>
<td>-------</td>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>cost</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>price</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bill</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fee</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>money</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>collard</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>charge</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>kale</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dollar</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Terms that start with *price* are also added to the price dictionary since most of them could not be retrieved due to their POS tags (adj).
APPENDIX B: EXAMPLE REVIEWS FROM THE DATASET AND THEIR CORRESPONDING OUTPUTS BY APPLICATION OF IMAS

Review 1: “Alright we are going to say that this place is Awesome! The service is outstanding and the food wonderful! We came here for Breakfast on Sunday Morning. They were very busy, but, we did not have to wait too long at all. Our server Colleen was great! She greeted us right away and brought our drinks very quickly. The food was very good and just love their Potatoes. The menu is fantastic! Lots to choose from. We like the decor and lots of parking in the lot. Nice Place and we will be back!” (Yelp Academic Dataset, 2013)

Food: POS
Price: POS
Ambiance: POS
Service: POS

Review 2: “Fish & Chips are very good but I did not care for the mac & cheese or the artichoke soup. The mac & cheese reminded me of cheese sauce and the soup had waaay to much lemon. The bar & seating areas are very clean, staff friendly but not bothersome and beer is cold- what's not to like about that?” (Yelp Academic Dataset, 2013)

Food: POS
Price: -
Ambiance: POS
Service: POS

Review 3: "love the food and the services , but you neesd to weight at 350ib to eat the burgers for free, the burgers are big and jucey" (Yelp Academic Dataset, 2013)

Food: POS
Price: -
Ambiance: -
Service: POS

Review 4: "The lunch special is an incredible value. I may never eat fast food again. Dinner is more expensive but is enough food to feed 2 hungry people. Great value, very fresh. The best Chinese food I've had in the east valley. Best for carryout or groups of 4 or less.” (Yelp Academic Dataset, 2013)

Food: POS
Price: POS
Ambiance: POS
Service: -
Review 5: “Visiting family is Arizona and stopped by for a quick bite to eat. The decor was great, restaurant was nice and clean and service was decent. It was our servers first day so it's hard to judge. However the food... AMAZING. Make sure you ask for your pizza well done. We also had the pasta primavera with a tomato cream sauce, best tasting pasta I've had in a while. The sangria wasn't the best I've had but it certainly wasn't terrible. GO, you won't be disappointed!” (Yelp Academic Dataset, 2013)

Food: POS
Price: -
Ambiance: POS
Service: POS
APPENDIX C: PYTHON CODE USED FOR MAS

def initial_segmentation_mas(self, sentences_folder, initial_segments_folder, noun_segments_folder, non_noun_segments_folder):
    reviews=101
    for i in range(1, reviews):
        fr=open(path.join(sentences_folder, str(i) + "_.txt"), "r")
        fw=open(path.join(initial_segments_folder, str(i) + "_.txt"), "a")
        if noun_segments_folder!="":
            fn=open(path.join(noun_segments_folder, str(i) + "_.txt"), "a")
        if non_noun_segments_folder!=""
            fnn=open(path.join(non_noun_segments_folder, str(i) + "_.txt"), "a")
        segmentID=1
        segments=[]
        sentences = nltk.sent_tokenize(fr.read())
        for sentence in sentences:
            x=0
            for segment in sentence.split(", "): x=x+1
            if x==len(sentence.split(", ")):
                segments.append(segment.strip("\n").strip("\t").strip("\n").strip("\t") + "t,"
            else:
                segments.append(segment.strip("\n").strip("\t") + "t,"
        segmentID=1
        for segment in segments:
            tokens=segment.split("\t")
            if (tokens[len(tokens)-1] not in self.conjs):
                tokens.append(".")
            ## Initial Segmentation
            fw.write(str(segmentID) + "t")
            for token in tokens:
                fw.write(token + "t")
            fw.write("n")
            ## Noun based segmentation
            tagged_tokens=nltk.pos_tag(tokens)
            if noun_segments_folder!="":
                fn.write(str(segmentID) + "t")
                for token, tag in tagged_tokens:
                    fn.write(token + "t")
            fw.write("n")
if (tag in self.nouns) or (token=="$") or (tag in self.old_conjs):
    fn.write(token+"\t")
fn.write("\n")

if non_noun_segments_folder!="":

    fnn.write(str(segmentID)+ "\t")
    for token, tag in tagged_tokens:
        if token in self.old_conjs:
            fnn.write(token+"\t")
        elif (token.startswith("like") or token.startswith("love") or token.startswith("hate")
            or (token in self.negation)):
            fnn.write(token+"\t")
        elif (((tag.startswith('J')) or (tag.startswith('R')))
            and (token not in nltk.corpus.stopwords.words('english'))):
            fnn.write(token+"\t")
        else:
            pass
    fnn.write("\n")

    segmentID+=1

fr.close()
fw.close()
APPENDIX D: PYTHON CODE USED FOR IMAS

def initial_segmentation_imas(self, sentences_folder, initial_segments_folder, noun_segments_folder, non_noun_segments_folder):
    reviews=101
    for i in range(1, reviews):
        fr=open(path.join(sentences_folder, str(i)+ ".txt"), "r")
        fw=open(path.join(initial_segments_folder, str(i)+ ".txt"), "w")
        if noun_segments_folder!="":
            fn=open(path.join(noun_segments_folder,str(i) + ".txt"),"a")
        if non_noun_segments_folder!="":
            fnn=open(path.join(non_noun_segments_folder,str(i) + ".txt"),"a")
        segmentID=1
        for sentence in fr:
            tokens=sentence.split("\t")
            if (tokens[len(tokens)-2] not in self.conjs):
                tokens[len(tokens)-1]="."
                tokens.append("\n")
            tagged_tokens=nltk.pos_tag(tokens)
            ## Segmentation
            if noun_segments_folder!="":
                segment=[]
                for token, tag in tagged_tokens:
                    if token in self.conjs:
                        segment.append(token)
                    fn.write(str(segmentID)+ "\t")
                    for t in segment: fn.write(t+"\t")
                    fn.write("\n")
                    segmentID+=1
                    segment[:]=[]
                    elif (tag in self.nouns) or (token=="$"):
                        segment.append(token)
                else:
                    pass
            segmentID=1
            if non_noun_segments_folder!="":
                segment=[]
                for token, tag in tagged_tokens:
                    if token in self.conjs:
                        segment.append(token)
```python
fnn.write(str(segmentID) + "\t")
for t in segment: fnn.write(t +"\t")
fnn.write("\n")
segmentID+=1
segment[:]=[]
elif (token.startswith("like") or
token.startswith("love") or token.startswith("hate")
or (token in self.negation)):
    segment.append(token)
elif (((tag.startswith('J')) or (tag.startswith('R')))
and (token not in nltk.corpus.stopwords.words('english'))):
    segment.append(token)
else:
    pass

segmentID=1
segment=[]
for token in tokens:
    if token in self.conjs:
        segment.append(token)
        fw.write(str(segmentID)+ "\t")
        for t in segment: fw.write(t+"\t")
        fw.write("\n")
        segmentID+=1
        segment[:]=[]
    else:
        segment.append(token)

fr.close()
fw.close()
```
APPENDIX E: PYTHON CODE USED TO ASSIGN ASPECTS TO SEGMENTS

def aspect_segments(self, inital_segments_folder, final_segments_folder):
    reviews=101
    for i in range(1, reviews):
        categorized_segments=[]
        count=-1
        fr=open(path.join(inital_segments_folder, str(i)+".txt"), "r")
        fw=open(path.join(final_segments_folder, str(i)+".txt"), "w")
        prev_conj=""
        file_size=len(fr.readlines())
        line_count=1
        fr.seek(0)
        for line in fr:
            aspects=[[0,"food"],[0,"service"],[0,"price"],[0,"ambiance"],
            [0,"neutral"]]
            if (line_count==1 or line_count==file_size): aspects[4][0]=1
            tokens=line.split("\t")
            if len(tokens[1:-1])==1:
                categorized_segments.append([line.strip("\n"),
                categorized_segments[count][1]])
            else:
                for token in tokens[1:-2]:
                    if token in self.food:
                        aspects[0][0]+=1
                    elif token in self.service:
                        aspects[1][0]+=1
                    elif token in self.price:
                        aspects[2][0]+=1
                    elif token in self.ambiance:
                        aspects[3][0]+=1
                    else:
                        pass
                aspects.sort()
                aspects.reverse()
            if line_count==1:
                next_conj=tokens[-2]
                prev_conj="",""
            else:
                next_conj=tokens[-2]
self.choose_category(aspects, categorized_segments, line.strip("\n"), count, prev_conj, next_conj)
prev_conj=next_conj

line_count+=1
count+=1

self.assign_next_category(categorized_segments)

for segment, group in categorized_segments:
    tokens=segment.split("\t")
    for token in tokens:
        fw.write(token + "\t")
        fw.write(group + "\n")
    fw.close()
fr.close()

# Segment-aspect assignment rules

def choose_category(self, category_scores, categorized_segments, segment, last_segment, prev_conj, next_conj):
    if (category_scores[0][0]==category_scores[1][0] and
category_scores[1][0]>category_scores[2][0]
and(category_scores[0][1]=="neutral" or
category_scores[1][1]=="neutral"):  
        if (category_scores[0][1]!="neutral"):  
            categorized_segments.append([segment, category_scores[0][1]])
        else: categorized_segments.append([segment, category_scores[1][1]])
    elif (category_scores[0][0]==category_scores[1][0] and
category_scores[0][0]==0):
        if prev_conj in self.in_conjs:
            categorized_segments.append([segment, categorized_segments[last_segment][1]])
        else: ## previously neutral
            categorized_segments.append([segment, "neutral"])
    elif (category_scores[0][0]!="neutral" and last_segment==-1):
        categorized_segments.append([segment, category_scores[0][1]])
    elif category_scores[0][1]!="neutral":  
        if prev_conj in self.in_conjs:
            categorized_segments.append([segment, "neutral_signed_in"])
        else:  
            categorized_segments.append([segment, category_scores[0][1]])
    elif (category_scores[0][0]=="neutral" and last_segment==-1):  
        categorized_segments.append([segment, category_scores[0][1]])
    elif category_scores[0][1]=="neutral":  
        if prev_conj in self.in_conjs:
categorized_segments.append([segment, categorized_segments[last_segment][1]])
else:
    categorized_segments.append([segment, category_scores[0][1]])
else:
    categorized_segments.append([segment, category_scores[0][1]])

elif (category_scores[0][0] == category_scores[1][0] and
      category_scores[0][0] != 0):
    if prev_conj in self.in_conjs:
        cats=[]
        for score, cat in category_scores:
            if category_scores[0][0] == score: cats.append(cat)

        if last_segment == -1:
            categorized_segments.append([segment, "neutral_signed_in"])

        elif (categorized_segments[last_segment][1] in cats):
            categorized_segments.append([segment,
                                          categorized_segments[last_segment][1]])
        else:
            if next_conj in self.in_conjs:
                categorized_segments.append([segment, "neutral_signed_in"])
            else:
                categorized_segments.append([segment, "neutral_signed_in"])
    else:
        if next_conj in self.in_conjs:
            categorized_segments.append([segment, "neutral_signed_in"])
        else:
            categorized_segments.append([segment, "neutral_signed_in"])
    else:
        pass

# Segment-aspect assignment rules

def assign_next_category(self, categorized_segments):
    cont=True
    while (cont):
        cont=False
        seg_count=0
        for segment in categorized_segments:
            if (segment[1] == "neutral_signed_in" and
                 (seg_count+1)<len(categorized_segments)):
                segment[1]=categorized_segments[seg_count+1][1]
if segment[1]=="neutral_signed_in":
    cont=True
elif(segment[1]=="neutral_signed_in"):
    segment[1]="neutral"
else:
    pass
seg_count+=1