Representing and Classifying User Reviews

Denis D. Mauá[1] **and Fabio G. Cozman**[1]

[1]Escola Politécnica – Universidade de São Paulo (USP)
CEP 05508-900 – São Paulo – SP – Brazil

{denis.maua,fgcozman}@usp.br

Abstract. *A large number of user reviews in the internet contains valuable information on services and products; for this reason, there is interest in automatically understanding such reviews. Sentiment Classification labels documents according to the feelings they express; instead of classifying a document into topics (sports, economics, etc), one attempts to tag the document according to overall feelings. Compared to the accuracy of traditional text categorization methods, sentiment classifiers have shown poor performance. We argue that such bad results are due to an improper representation of reviews. We describe a weakly supervised method that converts raw text into an appropriate representation, and show how techniques from information retrieval can acquire labeled data and process data using Markov logic. We report results on sentence classification and rating prediction that support our claims.*

1. Introduction

Review sites such as *Yelp (http://www.yelp.com)* and *Amazon (http://www.amazon.com)* encourage users to post reviews describing their opinions on products and services. These opinions can be used by other users to make informed decisions concerning purchases. Businesses can take advantage of reviews by obtaining consumer feedback on their products and services. Despite these potential benefits, the overwhelming number of reviews leads to an information overload that prevents users from fully exploiting the data.

Previous work on mining opinions from reviews have tried to summarize a document by its overall sentiment as a way to avoid information overload [Pang et al. 2002, Turney 2001, Pang 2005]. The focus has been on classification of texts according to sentiment indicators that can be binary or categorized. When compared to traditional text categorization [Manning and Schütze 1999], work on sentiment classification have reported poor classification performance.

We argue that these poor results are due to the lack of a proper representation for reviews. We propose a new representation that is better suited to such data (Section 3), and show how to obtain the representation from raw text using a weakly supervised method (Section 4). To do so, we use information retrieval techniques to acquire labeled data and Markov logic to specify classifiers [Richardson and Domingos 2006]. We report results for sentence classification and review rating prediction in Section 5.

2. Background

Information Retrieval (IR) techniques aim to retrieve relevant information from a large collection of (text) documents. The first step of an IR system is to index all documents by building a *co-occurrence matrix*. Let $\{D_i\}$ be a collection of N documents and F a M-sized set containing the vocabulary for collection $\{D_i\}$, i.e. the set of all distinct words appearing in some document. Hence, the co-occurrence matrix C is a $M \times N$ matrix, whose rows represent term distributions and columns represent documents in the bag-of-words representation. Each cell $C_{i,j}$ is equal the number of occurrences of the ith term in F in the jth document of the collection. A heuristic that has been reported to improve the accuracy of document retrieval is **tf-idf** weighting that takes an element of matrix C to be $C_{i,j} = tf_{i,j} \times idf_i$, where tf is the term frequency given by $tf_{i,j} = C_{i,j}/\sum_i C_{i,j}$, and idf is the inverse document frequency given by $idf_i = \log N - \log \sum_j C_{i,j}$.

Documents are retrieved by means of a *query* document. A query is simply a vector of length M where the terms in F that we want to search for are set to one. Documents are then ranked by their similarity to the query. The most common similarity metric used in the *cosine* given by $sim(v_1, v_2) = (v_1 \cdot v_2)/(|v_1||v_2|)$, where v_1 and v_2 are two documents in a proper vector representation, and $|v|$ denotes the norm of a vector v.

A difficulty in comparing documents with the cosine metric is that it matches words that co-occur in two documents, but many different words can be used to describe a same target-information. Also, words often have several meanings, and simple term matching may lead to the overestimation of document relevance [Manning and Schütze 1999]. One way to overcome these problems is to include *term co-occurrence* information in the document representations. If a word w_1 co-occurs often with another word w_2, they are likely to share some relation. Thus a query for w_1 might include documents where w_2 appear and vice-versa. A reasonable way to accomplish this is by producing a *low-rank approximation* of matrix C. The new canonical terms of the matrix can then be understood as latent concepts that are able to generalize the original high dimensional term vector F to a new lower dimensional vector F' of concepts. This technique of dimensionality reduction is known in the IR literature by the name of **latent semantic indexing** (LSI), which contrasts the usual word indexing by the new latent concept indexing . The k-rank LSI representation C_k of co-occurrence matrix C is given by $C_k = U\Sigma_k V$, where U and V are the term and document matrices, respectively, given by the *single value decomposition* of matrix C, and Σ_k is the diagonal matrix of the k greatest singular values of C.

In this paper we build classifiers based on **Markov logic** (ML). Markov Logic is a statistical relational language that uses a first-order logic (FOL) syntax to specify complex Markov networks [Richardson and Domingos 2006]. Formally, a knowledge base (KB) in Markov logic is a set of (implicitly conjoined) weighted first-order formulae. Let x_i be a ground atom with truth value assigned (e.g. $\texttt{HasWord(D,}\texttt{"meal"}) : True$) and $x = \{x_i\}$ an interpretation (i.e., the set of all possible ground atoms with truth values assigned). Then, the probability of a particular interpretation is given by

$$P(x) = \frac{1}{Z} \exp\left(\sum_i w_i f_i\right), \tag{1}$$

where Z is the partition function given by $\sum_{x \in X} \exp\left(\sum_i w_i f_i\right)$, and w_i denotes the weight attached to the ith grounded formula f_i.

For instance, a *MaxEnt* Text Classifier [Manning and Schütze 1999] can be imple-

Review 1. The service was great but the food was terrible.

Review 2. The food was great but the service was terrible.

Figure 1. Examples of user reviews.

mented in Markov Logic by the following model.

$$\begin{array}{rl} w_{\mathtt{w+,a+}} & \mathtt{HasWord(s,w+)} \rightarrow \mathtt{Topic(s,a+)} \\ w_{\mathtt{a+}} & \neg\mathtt{Topic(s,a+)} \end{array}$$

The first formula models how evidences (the presence of a word in a sentence) collaborate to topic discrimination under a naive Bayes assumption that word contributions are independent. The second formula imposes a prior on topic distributions. It represents that in the absence of any evidence no aspect topic should be true. The $+$ marks are syntactic sugars that indicate that these formulas are actually templates. In practice, if W is the number of words in the vocabulary and K the number of topics, then the first formula is turned into $W * K$ formulas, and the second formula becomes K prior formulas, one for each topic. The $(W + 1)K$ weights are then discriminatively learned from data.

For example, a document D with only word *meal* occurring has probabilities

$$P(\mathtt{Topic(D,"food")}|\mathtt{HasWord(D,"meal")}) \propto$$
$$P(\mathtt{Topic(D,"food")}, \mathtt{HasWord(D,"meal")}) = \frac{1}{Z}\exp(w_{\mathtt{meal,food}})$$

of belonging to topic *food* and $P(\neg\mathtt{Topic(D,"food")}|\mathtt{HasWord(D,"meal")}) \propto \frac{1}{Z}\exp(w_{\mathtt{food}})$ of not belonging to, given by Equation (1).

A considerable advantage of Markov Logic modeling is the availability of efficient methods for learning and inference and an opensource implementation (available at http://alchemy.cs.washington.edu).

3. A Proper Representation for Reviews

Sentiment classification methods usually assume that a review can be summarized by a single overall metric. However, as noted in [Snyder and Barzilay 2007] and [Titov and McDonald 2008], opinions expressed by reviewers are multi-faceted and cannot be correctly represented by a single sentiment score. Take the example reviews in Figure 1. Both examples express opposite sentiments on different aspects of an object. A neutral sentiment might be assigned to both reviews which clearly does not represent them well. A more reasonable assumption is that reviews can be summarized by aspect-based sentiments. For instance, we can classify reviews in Figure 1 as being, respectively, positive and negative according to aspect *service* and negative and positive, respectively, according to aspect *food*. We call the task of classifying documents according to the sentiments they express regarding a particular aspect as *aspect-based sentiment classification*.

Machine learning sentiment classification methods use the traditional *bag-of-words* model to represent reviews. Such a representation assumes that each document can be represented in the vector space by a function of the number of occurrences of each word in text. A nice way to visualize this representation is to see each document as an unordered list of words; Figure 2 shows the representation for both documents in Figure 1.

The bag-of-words representation assumes that documents regarding different subjects have different word distributions. For example, in sport articles one should expect to service food great terrible was but

Figure 2. Bag-of-words representation for *both* documents in Figure 1.

Review 1 service/*service* food/*food* great/*service* horrible/*food* was/*service* was/*food* but/*other*

Review 2 service/*service* food/*food* great/*food* horrible/*service* was/*service* was/*food* but/*other*

Figure 3. A proper representation for documents in Figure 1. Documents are no longer mapped to the same representation as in Figure 2.

find more occurrences of word “athlete” than in economic articles. Thus, we can predict the topic of a document by merely checking the word occurrences in it. However, this assumption fails when modeling reviews to sentiment classification, because here we are not concerned with the main subject of the review (e.g. restaurants, hotels or electronics) but with the many “micro”-opinions in the text. In fact, one can conceive all reviews as belonging to one same class, the class of reviews, thus expecting all of them to present similar word distributions. This explains the poor performance of sentiment classifiers based on the bag-of-words model.

A better representation for aspect-based sentiment classification is then to assign a tag to each word so as to discriminate with respect to the aspect it refers to. Figure 3 depicts a possible representation of reviews in Figure 1 according to this new model. Note that unlike the bag-of-words representation depicted in Figure 2 this new representation disambiguates between the two documents. A *bag-of-tagged-words* model is a representation model for text documents that extends the bag-of-words model as follows. Let $D = \{w_1, \ldots, w_N\}$ denote a document, where w_i denotes the ith token (word) of the document and N the length of the document. Let also $A = \{a_1, \ldots, a_M\}$ denote the set of topics, where each a_i denotes a distinct topic that a word may refer to.

Definition 1 *The* bag-of-tagged-words *for document D is the* bag-of-words *model for the new document $\bar{D} = \{t_1, \ldots, t_N\}$, where $t_i = (w_i, \{a_j\})$ is an ordered pair of the ith token and a subset $\{a_j\} \in A$ of the topic set.*

The main goal of this work is to develop a method that, given a set of topics A and a document D, generates a new document $\bar{D}$ represented as a bag-of-tagged-words. For reviews, the set of topics A is the set of aspects of an object that a user can comment on. Note that this definition of a bag-of-tagged-words model is a broad definition which fits any text document, not only reviews. In fact, this model can be seen as an instance of the general class of *topic models*, used in information retrieval and NLP to represent text documents [Manning and Schütze 1999].

4. The Method

An overview of the proposed method is depicted in Figure 4. Given a collection of documents D of size N and set of aspects A of size K, the system first produces an indexing of all sentences in the collection. Then, for each aspect $a \in A$ it retrieves the top-k most relevant sentences to a and produces a labeled set by assigning label a to the k sentences retrieved for a. This training dataset is used to learn a Markov logic classifier capable of categorizing sentences according to the aspect they refer to. The classifier is then used to

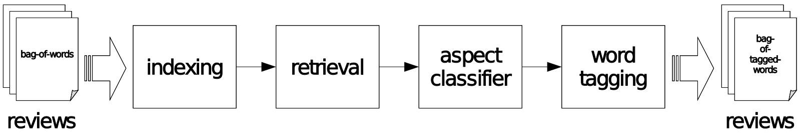

Figure 4. Method Overview. The diagram illustrates the main steps in the process of converting text documents to the *bag-of-tagged-words* representation.

Table 1. Aspect word sets used. Right column shows queries that retrieve sentences for aspects on the left column.

food	food dish course meal chicken portion taste soup bread pasta salad meat chicken
service	service staff wait waiter hostess host
value	price value cost worth cheap expensive
atmosphere	atmosphere ambiance ambience decor crowded noisy loud comfortable
experience	experience overall place favorite visit fun

build a new *bag-of-tagged-words* representation for documents in D by classifying each sentence in $d \in D$ according to the aspects $a \in A$ and by labeling each word in D by the predicted label(s) of the sentences they are in. The result is a new collection $\bar{D}$ where documents follow the new representation scheme. The next subsections detail each step.

4.1. Indexing

The first step is to build an index of all sentences in the collection. First, all documents are segmented into sentences, forming a new collection S of all sentences of all documents. Because reviews are very noisy,[1] a simple split-by-period procedure yielded best results in sentence segmentation than statistical methods. With each document segmented into sentences a co-occurrence matrix C_k of all sentences is built, followed by the application of tf-idf weighting and latent semantic indexing .

4.2. Retrieval

After indexing the sentences, the next step is to retrieve relevant sentences to compose a training dataset. In order to find sentences that are relevant to a given aspect we need to construct appropriate queries. This is very subjective and is the only part of our method where human supervision is required. An intuitive strategy is to look at text corpora and extract the most common words used by reviewers when commenting on a given aspect.

To facilitate the extraction of relevant words we have designed a bigram filter that returned the most frequent bigrams (pairs of adjacent words) whose first word is an adjective and second word is a noun. Although many other patterns could be investigated, we found this heuristic very satisfactory. We produced a list of approximately 50 words (adjectives and nouns) and manually cluster them into K groups, each group representing a distinct aspect in A. Table 1 shows the set of words used in our experiments with restaurant reviews. The small number of words used to represent each aspect is only feasible because of the LSI transformation applied to the co-occurrence sentence matrix, which makes other non-listed related terms relevant to the query.

[1] By noisy, we mean that simple formal rules such as capitalizing initial words and separating sentences with spaces are often not followed by review authors.

With a set of relevant words for each aspect in hand, we are able to build queries for retrieving relevant sentences. For each aspect $a \in A$ the top-k sentences more similar for the query q_a of relevant words of a are retrieved and labeled as a. The final result is a ($K \times k$)-length training dataset with equally distributed classes, which we can use to learn a sentence classifier.

4.3. Aspect Classifier

In this step we learn a K-topic MaxEnt classifier in Markov Logic, which classifies each sentence as whether it belongs to each of the aspects $a \in A$ as described in Section 2.

4.4. Word Tagging

The last step of the method is to tag each token in a document with aspects so as to produce a new *bag-of-tagged-words* representation. The process is described in pseudocode by Algorithm 1. For each document in the collection, we segment it into sentences as explained in Subsection 4.1 (line 2). Then, a *bag* structure is created to store the new representation (line 3). Each sentence in the original document is classified with respect to each aspect as follows. For each aspect, a binary classifier is run to predict whether the sentence belongs to that aspect (lines 5–9). If the output is positive, every word in the sentence is tagged with the aspect a tag and added to the bag. If not, words are added with no tag. A untagged word can be fitted in our definition of a bag-of-tagged-words (Definition 1) by adding a dummy extra topic to the aspect set and tagging untagged words with this new topic. Finally, the new bag is stored in the collection.

```
1  foreach document in the collection do
2      segment document into sentences
3      create new bag
4      foreach sentence in document do
5          foreach aspect in $A$ do
6              if Classify(sentence,aspect) then
7                  foreach word $w \in S$ do  add word/aspect to bag
8              else
9                  add word to bag
10     document ← bag
```

Algorithm 1: Word Tagging. Given a collection of documents, algorithm generates a collection of documents as a *bag-of-tagged-words*.

5. Experiments

We evaluate the proposed method using 6260 restaurant reviews downloaded from the we8there website (http://www.we8there.com). Each review is composed by a short text (on average 90 words) and a set of five ratings on a 1–5 scale regarding aspects *food, service, value, atmosphere* and *overall experience*. We report results on the sentence classification and aspect-based rating prediction tasks.

5.1. Sentence Classification

In this task, we evaluate the performance of the method in extracting useful sentences for training a Markov Logic classifier as well as the classification performance. The objective is for each aspect to classify a sentence as whether it comments on the aspect. We

Table 2. Results for the sentence classification task with different indexing schemes. The numbers report on F1-measure (in %).

Representation	Food	Service	Value	Atmosphere	Experience	**Average**
COUNT	47.80	40.36	**31.82**	19.36	24.72	32.81
TF-IDF	34.30	33.33	15.62	23.33	37.17	28.75
COUNT LSI	**47.84**	**67.53**	21.18	35.51	40.00	42.41
TF-IDF LSI	44.22	61.87	28.92	**39.37**	**40.31**	**42.94**

segmented all reviews into sentences in the dataset resulting in a database of 49662 sentences. We then filtered each sentence by removing low-frequency and function words, ending with a 3402x49662 word-sentence co-occurrence matrix. We extracted 500 sentences from this dataset and manually labeled each sentence, so that each had from zero up to five labels. Then, all sentences were converted into binary vectors: for each sentence i of the dataset a 3402-length vector was created by assigning 1 to the jth position iff term t_j occurs in i, and 0 otherwise. This binary representation helps improve classification accuracy. We use F1-measure to evaluate classification. F1-measure is the harmonic mean of precision and recall, given by F1 $= 0.5(p \times r)/(p + r)$, where p is the precision given by $p =$ (# of correct classified instances)$/$(# of total classified instances) and r is the recall given by $r =$ (# of correct classified instances)$/$(# of total instances belonging to aspect).

Evaluating Indexing Schemes In order to assess the different indexing schemes, we selected the top-500 sentences more relevant according to each aspect, ending with a 2500-length training dataset. Because a sentence may be relevant to more than one aspect, some sentences in the training data may occur more than once. Table 2 presents the results for different schemes according to the F1-measure, with best results for each aspect in bold. The COUNT scheme is the traditional bag-of-words model. The TF-IDF is the result of the tf-idf weighting to this vector. COUNT LSI and TF-IDF LSI are the COUNT and TF-IDF matrices, respectively, after latent semantic indexing being applied. On average, the TF-IDF LSI performed better than others, indicated by its higher score in the last column. Surprisingly, the COUNT scheme had the best result for aspect *value*. A possible explanation is that aspect *value* can actually be regarded as a sub-aspect of aspect *food*. This way, LSI schemes may increase the ratio of noisy by adding many *food* documents to the *value* set, hurting classifier accuracy.

Evaluating k Influence In order to assess the influence of the number of retrieved sentences per aspect in the training set, we performed an experiment with TF-IDF LSI indexing and k varying from 100 to 2000. As Table 3 shows, the best performance according to the average F1-measure happens when $k = 500$. For too small k, there is no sufficient number of training instances to correctly discriminate data. For too large k, the ratio of misclassified data in the training set increases, leading to poor performance.

Evaluating Classifiers We evaluated the performance of the Markov Logic Classifier against two baselines. The first baseline is a procedure that classifies all sentences as belonging to all aspects. Its precision is simply the number of sentences of each aspect over the total. For the second baseline we implemented a Naive Bayes Classifier. Naive Bayes Classifiers have been reported as perform well for text categorization [Manning and Schütze 1999]. Table 4 presents the results obtained with $k = 500$ and TF-IDF LSI indexing. The Baseline refers to the first baseline classifier. Except for as-

Table 3. Results for the sentence classification task with varying k. Numbers report on F1-measure (in %).

k	Food	Service	Value	Atmosphere	Experience	**Average**
100	24.13	56.67	28.13	33.71	41.59	36.84
200	26.81	**65.65**	22.53	34.29	**41.91**	38.24
500	44.22	61.87	**28.92**	**39.37**	40.31	**42.94**
1000	**50.00**	57.33	25.00	37.41	32.47	40.44
2000	46.53	50.73	18.41	29.27	30.04	35.00

Table 4. Results for the sentence classification task with different classifiers. Numbers report on F1-measure (in %).

	Food	**Service**	**Value**	**Atmosphere**	**Experience**	Average
Baseline	**45.14**	15.50	5.04	12.80	13.05	18.31
Naive Bayes	38.06	43.55	13.56	34.29	32.43	32.38
Markov Logic	44.22	**61.87**	**28.92**	**39.37**	**40.31**	**42.94**

pect *food*, Markov Logic performs much better than the others. The poor performance for aspect *food* is due to a low recall. This is because most part of the sentences comment on food quality, thus simply classifying sentences as belonging to food leads to good results. In fact, Markov Logic had the highest precision in aspect food (92.86% against 84.62% of Naive Bayes and only 45.14% for the Baseline). On the recall, however, results were far less satisfactory (29.02% against 24.55% for NB and 45.14% for Baseline).

5.2. Aspect-Based Rating Prediction

Aspect-Based Rating Prediction is the task of classifying review according to an aspect in a given pre-defined scale. In our dataset, ratings vary from 1–5. We performed a 80/20 split on the data, ending with 5008 and 1252 instances, respectively, for the training and test sets. A baseline was obtained by training a *MaxEnt* Markov Logic classifier with vectors following the common bag-of-words representation, except that only the presence/absence was stored as information. Then, using the algorithm described in Subsection 4.4, Markov Logic classifiers, and $k = 500$ and TF-IDF LSI scheme for sentence classification, we produce a bag-of-tagged-words representation of the dataset. Results are shown in Table 5. On average and in almost all aspects, the classifier learned with the bag-of-tagged-words (B-O-T-W) dataset performed slightly better than that learned with the common bag-of-words (B-O-W). By comparing the results on aspect *value* with those on Table 4, one can see that the sentence classification for this aspect had the poorest performance, which may explain the bad results in rating prediction.

Table 5. Results for the rating prediction task with different document representations. The numbers report on overall accuracy (in %).

Model	Features	Food	Service	Value	Atmosphere	Experience	**Average**
B-O-W	3402	61.84	54.00	**52.40**	48.32	59.36	55.18
B-O-T-W	18760	**64.16**	**55.20**	51.60	**48.40**	**60.16**	**55.90**

6. Related Work

Sentiment Classification The problem of sentiment classification has been treated often as a binary classification task, where the goal is to predict the overall polarity (positive or negative) of the document. [Pang et al. 2002] provides a detailed analysis of machine learning methods to this task, and reports performance much lower than traditional topic classification ($< 85\%$ for sentiment classification against $> 95\%$ for general text categorization). They conclude that the *bag-of-words* model is one of the main factors of the low accuracy of the classifiers. In [Turney 2001], the author tries to avoid possible limitations of machine learning methods by using mutual information metrics and representing the overall sentiment as the average sentiment of the sentences. He uses a simple PMI estimator which scores each sentence with discriminative words such as ”good” or ”bad”, reporting accuracies in the range $66\% - 84\%$. More recently, [Pang 2005] have allowed a finer-grained description of sentiments. They classify documents in four-classes (0–3) indicating the strength of sentiment, and achieve maximum accuracies of $\sim 66\%$ for a movie review dataset. In [Snyder and Barzilay 2007], this model is extended to allow sentiments to be classified regarding different aspects. They use perceptron-like algorithms and meta-classifiers to include aspect sentiment correlation into their final classifier. They show that explicitly modeling aspect correlation improves performance of aspect-based rating predictors, but do not report on accuracy (they use rank loss to evaluate their method).

Opinion Extraction Research in Opinion Extraction tries to extract passages from text representatives of an opinion, and then group extracted passages by the fine-grained aspect they refer to. This way, a review can be seen as a series of polarized opinions on fine-grained aspects of the assessed object. [Hu and Liu 2004] extract frequent nouns from text by applying an associative rule learner and use a syntactic parser to check for possible modifiers (adjectives) for each extracted word within a sentence. They use a thesaurus based procedure to find the polarity of modifiers, and assign each sentence to the fine-grained (noun) aspect with a polarity tag. A review is then represented by a set of fine-grained (noun) aspects with opinion polarities attached. For example, the reviews 1 and 2 of Figure 1 would be represented by the vectors `[service+`,`food-]` and `[service-`,`food+]`. [Popescu and Etzioni 2005] use PMI estimation, with a priori knowledge of the world, to search for possible candidates to fine-grained aspects in text. They apply relaxation labeling, a common technique from image processing to determine the polarity of aspect noun modifiers (which they extend to adverbs and verbs). They report gains up to 11% with respect to previous work.

Topic Modeling Topic Modeling is concerned with richer representations of text documents. A topic model is a probabilistic model that jointly assign topic distributions to documents and words, and can be seen as a probabilistic version of the more common latent semantic indexing (LSI) method [Manning and Schütze 1999]. Perhaps the work closest to ours is [Titov and McDonald 2008], where the authors propose a probabilistic generative model to represent reviews. Unlike our work, their method is completely unsupervised. Their representation model is also close to ours, but in their model each word is assigned a mixture model over the set of possible aspects. They report improvements on the review rating prediction task when compared to common bag-of-words model and also to other traditional Topic Models such as LDA and PLSA.

7. Conclusion and Future Work

Despite the success in traditional text categorization, machine learning methods have performed poorly on sentiment classification. Our claim is that such a poor performance is due to the lack of a proper representation of reviews. To support our claim, we have presented a novel representation for text documents that is better suited to sentiment classification. Our method relies on information retrieval techniques and Markov Logic to translate documents into this new representation with very little human intervention. We report results on aspect-based rating prediction showing that the proposed method indeed improves the performance. In the future, we plan to investigate unsupervised mechanisms to automate the only step in this process where human intervention is necessary.

References

Hu, M. and Liu, B. (2004). “Mining opinion features in customer reviews”. In McGuinness, D. L. and Ferguson, G., editors, *AAAI*, Proceedings of the Nineteenth National Conference on Artificial Intelligence, Sixteenth Conference on Innovative Applications of Artificial Intelligence, July 25-29, 2004, San Jose, California, USA, pages 755–760. AAAI Press / The MIT Press.

Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA.

Pang, B. (2005). “Seeing stars”. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics - ACL 05 ACL 05*, page 115.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). “Thumbs up?: sentiment classification using machine learning techniques”. In *EMNLP ’02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 79–86, Morristown, NJ, USA. Association for Computational Linguistics.

Popescu, A.-M. and Etzioni, O. (2005). “Extracting product features and opinions from reviews”. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing - HLT 05 HLT 05*, pages 339–346, Morristown, NJ, USA. Association for Computational Linguistics.

Richardson, M. and Domingos, P. (2006). “Markov logic networks”. *Machine Learning*, 62(1-2):107–136.

Snyder, B. and Barzilay, R. (2007). “Multiple aspect ranking using the good grief algorithm”. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 300–307, Rochester, New York. Association for Computational Linguistics.

Titov, I. and McDonald, R. (2008). “Modeling online reviews with multi-grain topic models”. In *WWW ’08: Proceeding of the 17th international conference on World Wide Web*, pages 111–120, New York, NY, USA. ACM.

Turney, P. D. (2001). “Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews”. In *ACL ’02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 417–424, Morristown, NJ, USA. Association for Computational Linguistics.