Review Classification Using Semantic Features and Run-Time Weighting

Chung-chi Huang[a], Meng-chiech Lee[b], Zhe-nan Lin[b], and Jason S. Chang[b]

[a]Institute of Information Systems and Applications, National Tsing Hua University, HsinChu, Taiwan 300, R.O.C.
u901571@gmail.com
[b]Department of Computer Science, National Tsing Hua University, HsinChu, Taiwan 300, R.O.C.
{iamrabbit37, rexkimta, jason.jschang}@gmail.com

Abstract. We introduce a method for learning to assign suitable sentiment ratings to review articles. In our approach, reviews are transformed into collections of n-gram and semantic word class features aimed at maximizing the probability of classifying them into accurate ratings. The method involves automatically segmenting review articles into sentences and automatically estimating associations between features and sentiment ratings via machine learning techniques. At run-time, a simple weighting strategy is performed to give extra weights to features in potential evaluative sentences (e.g., the first, the last sentences and sentences with adverbs) from others. Experiments show that word class information alleviates data sparseness problem facing higher-level n-grams (e.g., bigrams and trigrams) and that our model using *both* training-time n-gram and semantic features *and* run-time weighting mechanism outperforms a strong baseline with surface n-gram features by 2.5% relatively.

Keywords: sentiment, semantic orientations, classification, sentiment ratings, machine learning techniques.

1 Introduction

With the accessibility to the Internet in recent years, any Web user can easily be an article reader or a content provider. Web contents can be coarsely categorized into two groups: ones on knowledge and ones experience. Knowledge of any discipline is provided on on-line forums or on certain Web pages (e.g., WIKIPEDIA[1]) by the collaborative effort of a number of domain experts. On the other hand, more and more Internet users author their personal experiences on the Web. Experiences of using a high-tech product, dining in a restaurant, watching a movie and so on are documented. These publicly-available remarks are especially valuable in terms of marketing and recommendation.

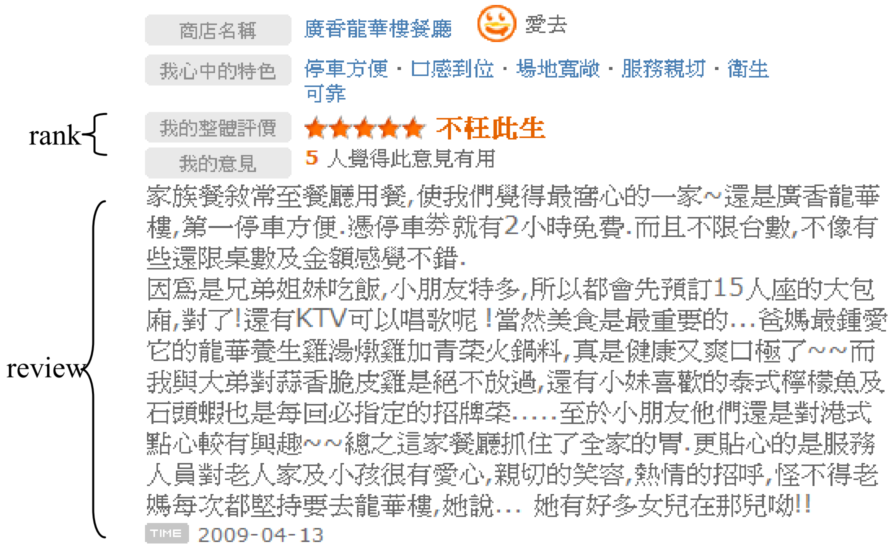
Take Yahoo!奇摩生活+[2] (Yahoo! Kimo Life Style) for example. It is a platform where Web users can author their experiences with or opinions toward the service-providing businesses including restaurants, hotels, amusement parks and etc. Figure 1 shows a restaurant review on Yahoo! Kimo Life Style. In this application, the overall evaluation on a service provider is rated via a number of stars. More stars, more positive the evaluation.

We present a model that automatically learns to rate review articles according to their semantic orientations. Reviews are ranked from one to five with larger integers meaning

[1] http://en.wikipedia.org/wiki/Wiki
[2] http://tw.lifestyle.yahoo.com/

reviewers' attitude toward the service is more positive and with smaller ones meaning reviewers' experience with the enterprise is more negative. During training, our model leverages machine learning techniques (e.g., maximum entropy or conditional random fields) to estimate associations between various ratings (i.e., one to five) and features, including surface n-gram features and semantic features. Semantic classes of words are exploited in our model in order to alleviate data sparseness problem, which is a serious problem if n-grams of higher degree (e.g., bigrams and trigrams) are employed. We describe our training process in more detail in Section 3.

At run-time, our model firstly transforms a review article into a collection of features and then ranks the review based on the features and the trained machine learning model. Additionally, features may be weighed according to some characteristics such as whether or not the features appear in the first or last sentence of the article and whether or not the features occur in the sentence containing adverbs which are usually indicators of the existence of statements expressing sentiments and often intensify the positive/negative semantic orientation. For example, the adverb "很" (very) in the sentence "人氣 很 高" (it is very popular) means that the restaurant is not just popular, but very popular.



商店名稱 廣香龍華樓餐廳 愛去

我心中的特色 停車方便・口感到位・場地寬敞・服務親切・衛生可靠

我的整體評價 ★★★★★ 不枉此生

我的意見 5 人覺得此意見有用

家族聚敘常至餐廳用餐,使我們覺得最窩心的一家~還是廣香龍華樓,第一停車方便.憑停車劵就有2小時免費.而且不限台數,不像有些還限桌數及金額感覺不錯.
因為是兄弟姐妹吃飯,小朋友特多,所以都會先預訂15人座的大包廂,對了!還有KTV可以唱歌呢 !當然美食是最重要的...爸媽最鍾愛它的龍華養生雞湯燉雞加青菜火鍋料,真是健康又爽口極了~~而我與大弟對蒜香脆皮雞是絕不放過,還有小妹喜歡的泰式檸檬魚及石頭蝦也是每回必指定的招牌菜.....至於小朋友他們還是對港式點心較有興趣~~總之這家餐廳抓住了全家的胃.更貼心的是服務人員對老人家及小孩很有愛心,親切的笑容,熱情的招呼,怪不得老媽每次都堅持要去龍華樓,她說... 她有好多女兒在那兒呦!!

TIME 2009-04-13

Figure 1: A restaurant review on the Web.

2 Related Work

Recently, various parties have paid a myriad of attention to the research of sentiment analysis for many reasons. Sentiment analysis provides organizations, candidates, political parties and hosts of certain events with the opportunity to automatically identify the subjective remarks toward them or even compile the overall favorability or unfavorability for them. In terms of marketing, damage control, and risk management, it is of great importance.

Some of the research on sentiment analysis focuses on predicting semantic orientations for words (in terms of polarity directions and intensities), especially adjectives, which are good indicators of subjective statements (Hatzivassiloglou and Mckeown, 1997; Hatzivassiloglou and Wiebe, 2000; Wiebe, 2000; Turney and Littman, 2003). Some, however, focuses on identifying the sentiments of collocations (Wiebe *et al.*, 2001), of phrases containing adjectives and adverbs (Turney, 2002), and of phrases marked with polarity (Wilson *et al.*, 2005).

Past research utilizes lexicons (hand crafted in (Huettner and Subasic, 2000) and automatically mined in (Yang *et al.*, 2007a)) or n-gram features (Pang *et al.*, 2002) in automatic analysis on documents' sentiments, such as sentiment categorization of reviews or mood classification of Web blogs. The sentiment classification of documents may be approached by first analyzing their sentences' semantic orientations (Yang *et al.*, 2007b) or sequential sentiments of their sentences (Mao and Lebanon, 2006) or by treating the sentences within as a whole (Mishne, 2005).

In this paper, the goal of our model is to automatically classify reviews into five sentiment ratings using article-level surface n-gram and semantic word-class features. At run-time, we further leverage a simple weighting strategy to give extra weights to features in sentences likely to contain sentiments or evaluative expressions.

3 The Method

3.1 Problem Statement

We focus on ranking review articles using five-star rating scheme. Reviews' sentiment ratings are returned as the output of the system and can be used as rank suggestion to reviewers. We now formally state the problem that we are addressing.

Problem Statement: We are given a general purpose machine learning model *ML* (e.g., maximum entropy model), a semantic word-class thesaurus *WC* (e.g., Chinese synonym thesaurus), and a review article *RE*. Our goal is to assign the most probable sentiment rating *r* (*r* is an integer between one and five) to the review *RE* via *ML*. For this, we transform *RE* into a collection of feasible features, F_1, …, F_m, such that the correct rank of *RE* is likely to be obtained.

In the rest of this section, we describe our solution to this problem. First, we define a strategy for transforming review articles into collections of features (Section 3.2). These feature collections are then utilized to train a machine learning method regarding the associations between features and different sentiment ratings. Finally, we show how our model rates a review article at run-time by applying highly-tuned rank-feature associations and feature weighting (Section 3.3).

3.2 Learning Rank-Feature Associations

We attempt to find transformations from review articles into effective feature collections that consist of terms (or features) expected to assist in rank determination. Our learning process is shown in Figure 2.

(1) Preprocess review articles in training data
(2) Transform review articles into feature collections
(3) Estimate associations between ranks and features
(4) Output the trained model

Figure 2: Outline of the process used to train our model.

Preprocessing Review Articles. In the first stage of the learning process (Step (1) in Figure 2), we word segment and PoS tag the review articles in training data. For example, we segment and tag the Chinese restaurant review "好吃又便宜！真棒！值得推薦" as "好吃/VH 又/Db 便宜/VH ！/Fw 真/DFVH 讚/bb ！/Fw 值得/VHDb 推薦/VC" (English translation: The food is good and not expensive! Goody! Worth recommending to friends). Moreover, we heuristically divide reviews into "sentences" by using punctuation marks as delimiters. Take the above article for instance. It comprises three sentences: "好吃又便宜！", "真棒！", and "值得推薦". Note that, in these review articles, some Chinese characters are "abbreviated" by their sound-like shorthands. To avoid the possibility of degrading the performance of a Chinese word segmenter, we replace popularly used shorthands with their corresponding regular form. See Table 1 for examples.

Table 1: Shorthands for some Chinese characters.

Shorthand	Regular Form	Shorthand	Regular Form
ㄌ	了	ㄅ	吧
ㄉ	的	ㄋ	呢
ㄚ	啊	ㄇ	麼

Transformation from Reviews into Features. In the second stage of the training (Step (2) in Figure 2), we transform review articles into collections of features. Figure 3 shows the transformation algorithm.

```
procedure TransformationToFeatureCollections($RE$, $WC$)
(1) $FeatureCol$=“” //NULL
      for each sentence $s$ in the review $RE$
         for each word $w_i$ in $s$
(2)           add $w_i$ to $FeatureCol$
(3)           add ($w_{i-1}$, $w_i$) to $FeatureCol$
(4a)          $wordClass_{i-1}$=WordClassLookUp($w_{i-1}$, $WC$)
(4b)          $wordClass_i$=WordClassLookUp($w_i$, $WC$)
(4c)          $wordClass_{i+1}$=WordClassLookUp($w_{i+1}$, $WC$)
(5)           if $wordClass_i$ is not NULL
(6)              add $wordClass_i$ to $FeatureCol$
                 if $wordClass_{i-1}$ is NULL
(7)                 add ($w_{i-1}$, $wordClass_i$) to $FeatureCol$
                 else
(8)                 add ($wordClass_{i-1}$, $wordClass_i$) to $FeatureCol$
(9)              if $wordClass_{i+1}$ is NULL
                    add ($wordClass_i$, $w_{i+1}$) to $FeatureCol$
      return $FeatureCol$
```

Figure 3: Transformation algorithm.

In Step (1) of the transformation algorithm we define an empty collection, *FeatureCol*, for gathering article-level n-gram or semantic word-class features. For each word in the sentences of the given review *RE*, we include it into *FeatureCol* as unigram feature (Step (2)). We also consider bigram feature (Step (3)).

In Steps (4a), (4b) and (4c) we look up the (semantic) word classes for words w_{i-1}, w_i and w_{i+1}, and denote them as $wordClass_{i-1}$, $wordClass_i$ and $wordClass_{i+1}$, respectively. Word class features incorporated into our model aim at alleviating data sparseness problem and reducing out-of-vocabulary encounters at run-time. If there is a word class for w_i in the semantic thesaurus *WC*, it is added into the feature collection (Steps (5) and (6)), referred to as class-based unigram feature, **CBuni** for short. On the other hand, we deal with the class-based bigram feature, **CBbi** for short, from Step (7) to (9). Compared to the surface bigram features in Step (3), the semantic class-based bigrams are not necessary bigrams of word classes. Words can be included in the class-based bigram features if one of their adjacent words can be labeled with semantic classes. (Steps (7) and (9)). In the end, this procedure returns the feature collection of the review article.

Although we only consider n-gram and semantic features, one can integrate other feasible features into this transformation algorithm, such as dependencies of words.

Association Estimation. In the third and final stage of the learning algorithm, we exploit a machine learning technique, maximum entropy model, to estimate the associations between sentiment ratings (i.e., from one to five) and features. Recall that, in our model, types of features include unigram, bigram, class-based unigram and class-based bigram.

Once we transform review articles into feature collections as previously described, these features with corresponding ranks specified by reviewers are fed to train maximum entropy model (MaxEnt for short). MaxEnt derives its set of system parameters via

$$\theta^* = \arg\max_{\theta} \prod_{RE} \Pr\left(rank\left(RE\right) \middle| features\left(RE\right), \theta\right) \tag{1}$$

where θ denotes any possible set of system parameters, *features*(*RE*) the features provided with the review *RE*, and *rank*(*RE*) the rank, or the sentiment rating, of *RE*. Moreover, the probability in Eq. 1 is estimated by

$$\exp\left(\sum_i \lambda_{i,r} f_{i,r}\left(features\left(RE\right), rank\left(RE\right)\right)\right) \tag{2}$$

in which the θ in Eq. 1 is factored into a set of $\lambda_{i,r}$'s, standing for the feature weights of the binary-valued feature functions (i.e., $f_{i,r}$'s). $f_{i,r}$(*features*(*RE*), *rank*(*RE*)) returns 1 if r equals to *rank*(*RE*) and *features*(*RE*) contains the feature which f_i takes note of, or represents, and 0 otherwise. For instance, given a review, ranked two-star, feature function of $f_{\text{"dislike"},2}$ yields 1 if the unigram "dislike" exists in the feature collection of this review, and 0 if not. Furthermore, a larger $\lambda_{i,r}$ means that the feature f_i represents is considered to be a strong indicator for the rank r. Provided with rank-annotated reviews, MaxEnt manages to tune the feature weights to capture suitable associations, between distinct feature and rank (see Eq. 1).

Note that words may be repeated in a review article. For example, a review may mention the *great* food served by the restaurant, the great service provided by the staff in the restaurant, the *great* atmosphere in the restaurant, and the *great* personality of the restaurant owner at the same time. Using binary-valued feature functions in Eq. 2, unfortunately, may not correctly reflect the overall enjoyable dining experience in terms of the food, the service, the atmosphere, the personality of the restaurant owner. Therefore, the feature functions ($f_{i,r}$'s) in Eq. 2 are re-defined to return the frequency, observed in a review, of the corresponding features.

3.3 Run-Time Rank Classification

Once the associations between features and sentiment ratings are tuned, for a given review article, our model determines its most probable rank r^*, satisfying $\arg\max_r \Pr\left(r \middle| features(RE), \theta^*\right)$ where the probability is estimated via Eq. 2. Recall that *features*(*RE*) is acquired using the review-to-feature transformation algorithm in Figure 3.

Inspired by (Yang *et al.*, 2007b), suggesting that the last sentence plays an important role in determining the overall emotion of a blog document, at run-time we take some sentential characteristics into account. Specifically, we attempt to distinguish sentences with these characteristics from those without.

Intuitively, the first sentence and the last sentence of a review as well as the sentences with adverbs may carry more sentiment information than those that are not. A reviewer may start with or conclude by a positive or negative statement on the overall personal experience with the service provider. Also, adverbs in sentences tend to put extra stress on the modified adjectives, posing a stronger indication toward the sentiments of the adjectives than the adjectives alone (e.g., the "very" in the phrase "very nice" implies the semantic orientation is quite positive).

In this paper, three types of sentences, the first (**firstS**) and the last (**lastS**) sentence and the sentences with adverbs (**advS**), are considered to be more informative in determining the sentiment rating of a review. Therefore, at run-time, these sentences are given more weights by doubling the n-gram and semantic features therein while features of others not belonging to these three types are not duplicated. For instance, we include the unigram features "very" and "good", and the bigram feature "very good" *twice* for the sentence with an adverb, "very good". On the other hand, for the sentence, "good", not belonging to **firstS**, **lastS**, and **advS** has single unigram feature "good".

4 Experiments

Our model was designed to find the most likely sentiment rank for a review article. As such, it will be trained and evaluated on reviews collected from the Web. In this section, we first present the details of training our model for the evaluation (Section 4.1). Then, Section 4.2 reports the results of the experiments using different combinations of features mentioned in Section 3. Finally, discussion concerning error types and potential improvements to the system is made in Section 4.3.

4.1 Training Our Model

We used a set of 31,500 restaurant reviews for training, obtained from querying "餐廳" (restaurant) in Yahoo!奇摩生活+ (Yahoo! Kimo Life Style). Each review was provided with a rank, from one-star to five-star, by its author. Notice that the collected reviews were uniformly distributed over ranks, that is, 6,300 reviews per rank. After word segmentation, our training data consisted of approximately 2M Chinese words.

On the other hand, our semantic word classes were based on 同義詞詞林 (Chinese synonym thesaurus), and we employed Zhang's MaxEnt toolkit[3] to tune the feature weights described in the training procedure. Following table shows some example words in two semantically related topics from the Chinese synonym thesaurus. Words in group "Ga01A" express the concept of "happiness" while words in "Gb10A" express the concept of "dislike".

Group	Word	Group	Word
Ga01A	高興	Gb10A	討厭
Ga01A	開心	Gb10A	厭惡
Ga01A	歡愉	Gb10A	嫌惡

4.2 Evaluation Results

In this subsection, we first examine the difficulty of our problem: classifying reviews into five different semantic ratings (i.e., from one-star to five-star). We asked one graduate and one PHD student to classify 50 randomly sampled reviews, uniformly collected from five semantic ranks, without knowing the original semantic ratings given by authors, which were considered the gold standards. Table 2 summarizes the accuracy of these two human annotators. The low accuracy in Table 2 suggests that even humans find this classification task difficult and that words and phrases, though evidence to semantic orientations, might be associated with different semantic ranks from person to person. Inconsistent understanding of evaluative expressions

[3] http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html

(words or phrases) resulting mostly from human's subjective judgement makes it hard to find an accurate classifier.

Table 2: Results of human annotators.

	Accuracy (%)
Human A	44
Human B	46

In experiments, to inspect the performance of our model, a test data set, made up of 5,000 restaurant reviews, was allocated. Since testing data were uniformly distributed among sentiment ratings, the expected precision of random guessing was 20%. Table 3 shows the accuracy of our MaxEnt baseline using surface n-gram features (Note that using feature presence, instead of feature frequency, (see Eq 2 in Section 3.2) did not improve the accuracy of our baseline). As we can see, our MaxEnt baseline achieved comparable results with humans' (see Table 2). In Table 3, the result of conditional random fields[4] (CRFs) fed with the same features is listed for comparison. Although there was a noticeable difference between the performances of two machine learning techniques, these two substantially outperformed the method of random guessing.

Table 3: Results of different machine learning techniques.

	Accuracy (%)
CRFs	36.56
Baseline	45.18

Table 4 summarizes the performance of our model using class-based unigram (**CBuni**) or class-based bigram (**CBbi**) features. In Table 4, we find that our model benefited from using higher-degree class-based n-grams ((3) vs. (2)), and that our model with semantic unigram and bigram features improved approximately 1% over the baseline ((3) vs. (1)), suggesting the baseline probably suffered from data sparseness problem in that it only took surface word forms into account.

Table 4: Performance on semantic features.

	Accuracy (%)
Baseline	(1) 45.18
+**CBuni**	(2) 45.82
+**CBuni**+**CBbi**	(3) 46.14

On top of n-gram and semantic features, we conducted experiments where, at run-time, features of sentences belonging to **firstS**, **lastS**, and **advS** received larger weights. As suggested in Table 5, replicating features in sentences with adverbs worked better than replicating those in the first and last sentences of reviews. Although doubling features of sentences falling into the categories of **firstS** and **lastS** was not as effective as expected, it might not mean these sentences should not be taken more seriously, but might mean we need a more sophisticated run-time weighting strategy, instead. Encouragingly, giving more weights to sentences containing adverbs by means of replicating their n-gram and semantic features boosted the precision to 46.34, achieving a relative gain of 2.5% over the strong MaxEnt baseline.

[4] We used the implementation provided on http://flexcrfs.sourceforge.net/

Table 5: Performance on different run-time weighting mechanisms.

	Accuracy (%)
firstS	45.56
lastS	45.62
advS	46.34

On the other hand, our model may be used to provide a list of rank suggestion for a review. As such, we leveraged mean reciprocal rank (MRR), the average of the inverse of the rank of the *first* correct answer, to measure the suggestion quality of our model. The MRR of our best model (i.e., baseline+**CBuni**+**CBbi**+**advS**) was 0.6849, indicating that, most of the time, the first and the second sentiment ratings on our suggestion lists would be adopted by review authors.

4.3 Discussion

To analyze errors our system made, we examine the confusion matrix, shown in Table 6, on our testing data. A confusion matrix lists system's results with respect to the gold standard. Take 266 (the bold-faced number in Table 6) for example. It is the number of reviews that were wrongly labeled to 2 by our system, but were labeled to 1 by the authors (i.e., the gold standard). Additionally, the accuracy of our system concerning each semantic rating is also shown (the last column).

As suggested by Table 6, it was not easy for the model to distinguish restaurant reviews between rank 1 and rank 2 and ones between rank 4 and rank 5. The reason is probably because maximum entropy model, our system bases on, does not "understand" that there are strength differences among semantic ratings from one to five. One possible way to avoid this problem is to utilize multi-level classifiers. For example, a top-level classifier first categorizes reviews into negative, neutral, and positive ones. Negative- and positive-labeled reviews are subsequently classified into very negative and negative ones and positive and very positive ones respectively.

Table 6: Confusion matrix of our best system (baseline+**CBuni**+**CBbi**+**advS**).

		Our System					
		1	2	3	4	5	Accuracy (%)
Gold Standard	1	488	**266**	121	35	90	48.80
	2	274	410	242	31	43	41.00
	3	96	194	472	135	103	47.20
	4	28	29	146	306	491	30.60
	5	52	28	79	200	641	64.10

Also, we find that errors regarding the originally neutral reviews were quite spread out. In other words, reviews labeled as 3 in the gold standard might be wrongly classified into 1, 2, 4, and 5 with some proportion. The rationale behind this is that authors probably did not check the suitable semantic ratings at submission because of laziness (default rating is 3) or cultural influence (out of politeness, Chinese people prefer the modest or the neutral choice). This viewpoint is, to some extent, verified by the scattered "errors" made by the two humans concerning the neutral restaurant reviews (bold-faced numbers in Table 7 and Table 8).

Table 7: Confusion matrix of human A.

		Human A					
		1	2	3	4	5	Accuracy (%)
Gold Standard	1	7	2	1	0	0	70
	2	5	5	0	0	0	50
	3	**1**	**3**	**2**	**3**	**1**	20
	4	0	0	1	3	6	30
	5	0	0	2	3	5	50

Table 8: Confusion matrix of human B.

		Human B					
		1	2	3	4	5	Accuracy (%)
Gold Standard	1	10	0	0	0	0	100
	2	8	1	1	0	0	10
	3	**3**	**0**	**2**	**4**	**1**	20
	4	0	0	1	3	6	30
	5	0	0	0	3	7	70

5 Summary and Future Work

In summary, we have introduced a method for learning to assign sentiment ratings to review articles using n-gram and semantic features. The method involves transforming reviews into collections of features, estimating associations between features and ratings, and weighting features in evaluative sentences by duplication. We have implemented and evaluated the method as applied to restaurant reviews on the Web. In the evaluation, we have shown that the method leveraging training-time semantic features and run-time weighting outperforms the strong baseline with n-gram features. As for future work, we would like to examine other weighting strategies in order to better handle features in the first or the last sentences of review articles, usually believed to correlate highly with authors' semantic orientations.

References

Hatzivassiloglou, V. and K. R. McKeown. 1997. Predicting the Semantic Orientation of Adjectives. *Proceedings of the Annual Meeting of the ACL / European Association of Computational Linguistics*, 174-181.

Hatzivassiloglou, V. and J. M. Wiebe. 2000. Effects of Adjective Orientation and Gradability on Sentence Subjectivity. *Proceedings of the International Conference on Computational Linguistics*, 299-305.

Huettner, A. and P. Subasic. 2000. Fuzzy Typing for Document Management. *Proceedings of ACL Companion Volume: Tutorial Abstracts and Demonstration Notes*, 26-27.

Mao, Y. and G. Lebanon. 2006. Sequential Models for Sentiment Prediction. *Proceedings of ICML Workshop on Learning in Structured Output Spaces*.

Mishne, G. Experiments with Mood Classification in Blog Posts. 2005. *Proceedings of Workshop on Stylistic Analysis of Text for Information Access*.

Pang, B., L. Lee and S. Vaithyanathan. 2002. Thumbs up? Sentiment Classification using Machine Learning Techniques. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 79-86.

Turney, P. D. 2002. Thumbs up or Thumbs down? Semantic Orientation Applied to Unsupervised Classification of Reviews. *Proceedings of the Annual Meeting of the ACL*, 417-424.

Turney, P. D. and M. Littman. 2003. Measuring Praise and Criticism: Inference of Semantic Orientation from Association. *ACM Transactions on Information System*, 21(4).

Wiebe, J. M. 2000. Learning Subjective Adjectives from Corpora. *Proceedings of National Conference on Artificial Intelligence.*

Wiebe, J. M., T. Wilson and M. Bell. 2001. Identifying Collocations for Recognizing Opinions *Proceedings of the Annual Meeting of the ACL / European Association of Computational Linguistics Workshop on Collocation*.

Wilson, T., J. M. Wiebe and P. Hoffmann. 2005. Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 347-354.

Yang, C., K. H.-Y. Lin and H.-H. Chen. 2007a. Building Emotion Lexicon from Weblog Corpora. *Proceedings of the Annual Meeting of the Association of Computational Linguistics*, 133-136.

Yang, C., K. H.-Y. Lin and H.-H. Chen. 2007b. Emotion Classification Using Web Blog Corpora. *Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence*, 275-278.