Feature Selection in Sentiment Analysis

Olena Kummer — Jacques Savoy

Institut d'informatique, Université de Neuchâtel
Rue Emile Argand 1, 2000 Neuchâtel, Suisse
{Olena.Zubaryeva, Jacques.Savoy}@unine.ch

ABSTRACT. *In this article, we propose a new method for feature selection and sentiment classification. To identify the most salient features belonging to the specific categories, we use the Z score measure. Based on this score, we can identify confident features and use the Information Gain (IG) measure to obtain scores for terms appearing in the neighborhood of the confident features. Based on this information, we propose a new weighting scheme to perform sentiment classification. The proposed feature selection and classification method was evaluated on two publicly available datasets using various text representation schemes. Accuracy rates over 10 fold cross-validation indicate that the proposed approach performs on the same level, and sometimes outperforms, other schemes like SVM and Naïve Bayes.*

RÉSUMÉ. *Dans cette communication, nous proposons une nouvelle méthode pour la sélection des termes et la classification automatique de sentiments. Pour déterminer les caractéristiques les plus adéquates d'une catégorie, nous nous appuyons sur le score Z. Cette mesure nous permet de définir les termes pertinents et, recourant à la mesure du gain d'information, nous pouvons également évaluer les termes dans le voisinage des termes pertinents. Sur ces deux éléments, nous proposons un nouveau modèle de classification. Notre modèle a été évalué en recourant à deux collections tests et sur la base de plusieurs représentations. La performance de notre modèle (validation croisée) s'avère égale voire supérieure à des approches connues basées sur le modèle SVM ou Naïve Bayes.*

KEY WORDS: *Text Classification, Feature Selection, Evaluation, Sentiment Analysis.*

MOTS-CLÉS: *Catégorisation de textes, sélection des termes, évaluation, analyse de sentiments.*

1. Introduction

In this article we deal with two related text classification problems. First, faced with a short review, we want to automatically classify it as opinionated or not (factual information). In a second step, when faced with an opinionated text, we want to classify it into two possible categories, namely positive or negative. The

results of such analysis could be potentially important and profitable in identifying trends for consumer research, market analysis, and other organizations. We define the task of text classification by the presence or absence of an opinion as *opinion classification* (OC), while the determination of the text polarity (positive vs. negative) will be called *sentiment classification* (SC).

The suggested approach is based on a supervised learning scheme that uses feature selection techniques and weighting strategies to classify sentences into two categories (opinionated vs. factual or positive vs. negative). Our main objective is to design and implement a new classification scheme able to achieve comparable classification effectiveness to popular state-of-the-art approaches such as Support Vector Machines (SVM) or Naïve Bayes. We also want a classification model able to provide a decision that can be understood by the final user (instead of justifying the decision by considering the distance difference between selected examples).

The rest of this article is organized as follows. First, we present the state-of-the-art approaches in Section 2. Section 3 describes the suggested method. Section 4 gives an overview of the experimental setup, datasets and evaluates our method. In Section 5 we discuss our experimental results and expose a failure analysis.

2. Related Work

Feature selection is often integrated as the first step in machine learning algorithms like SVM, Neural Networks, k-Nearest Neighbors, etc. The main goal of the feature selection is to decrease the dimensionality of the feature space and thus computational cost. As a second objective, feature selection will reduce the over-fitting of the learning scheme to the training data. During this process, it is also important to find a good tradeoff between the richness of features and the computational constraints involved when solving the categorization task.

Several studies expose the feature selection question. Forman (2003) reports an extensive evaluation of various schemes in text classification tasks. Dave *et al*. (2003) give an evaluation of linguistic and statistical measures, as well as weighting schemes to improve feature selection. Liu *et al*. (2007) study the effect of various feature selection schemes on imbalanced data.

In Pang *et al*. study (2002) several machine learning algorithms were analyzed on a movie review dataset, together with different feature selection techniques. Features are usually words, or bigrams of words, that could have been somehow pre-processed, for example, stemmed or lemmatized. Pang & Lee (2004) achieved the best result using a SVM method based on words (unigrams) while the Naïve Bayes method gave slightly lower accuracy. Pang & Lee (2004) propose to first separate subjective sentences from the rest of the text. They assume that two consecutive sentences would have similar subjectivity labels, as the author is inclined not to change sentence subjectivity too often. Matsumoto *et al*. (2005) used word sub-sequences (n-grams) and dependency trees of sentences to calculate the frequency

patterns in the word usage across different sentences. Using an SVM model, they achieved an accuracy of 88.1% with the language-independent features on a more recent version of the data set used by Pang & Lee (2004).

Positional information of the words in text can also be taken into account. In this perspective, Raychev & Nakov (2009) use multinomial Naïve Bayes, together with the position information in the feature set. They conducted their experiments on the movie dataset achieving an 89% accuracy using unigrams and bigrams, which is a slight improvement over the performance reported by Pang & Lee (2004).

Another variation of the SVM method was adopted by Mullen & Collier (2004) who used WordNet syntactic relations together with topic relevance to calculate the subjectivity scores for words. They reported an accuracy of 86% on the Pang & Lee's movie review dataset. Zaidan *et al*. (2007) used SVM and so-called "rationales" corresponding to words and phrases explaining a particular classification decision (decisions annotated by humans). Whitelaw *et al*. (2005) also employed SVM and a lexicon created with a semi-automated technique, which was then improved manually. Recently, Paltoglou & Thelwall (2010) suggested using BM25 and *tf idf* weighting schemes coupled with the SVM classifier. This solution achieved a significant improvement over the previous SVM-based approaches.

3. Proposed Method

The proposed method was built upon the previous work in feature selection strategies for textual classification (Yang & Pedersen, 1997; Forman, 2003; Zubaryeva & Savoy, 2009). First, we represent each sentence as a vector of features. We define a feature as a unit of text that could be a word, a stemmed word, a punctuation mark, or a bigram of two consecutive words in a text. Our classification method is composed of several steps. First we estimate a category score for every feature using the Z score as explained in Section 3.2. This Z score value will be used to identify the confident features. Afterwards, we evaluate the discriminative power of terms appearing in the neighborhood of these confident features using the information gain ratio (Section 3.3). Based on these scores, we compute the overall score of the sentence over the classification categories.

3.1. Selecting Features

After the tokenization of a sentence or a short review, we remove all word types having an overall occurrence frequency of three or less in the corpus. We think that word types having a small occurrence frequency convey more noise than pertinent information. This pruning heuristic also allows us to eliminate specific slangs that could be hard to classify without any additional lexical vocabulary.

To represent a text, we adopted the bag-of-words assumption in which each word is stemmed according to Porter's method. Such a sequence of words corresponds to our unigram model. When considering two consecutive words to form a feature, we generate a bigram model. Of course, we can mix the two representations and thus we represent a sentence by a set of stems (unigrams) and sequences of two stems (bigrams). We do not use any POS tagging information in order to reduce the language-specific pre-processing of the corpus.

3.2. Z Score

The second step is to determine the terms belonging clearly to one category. To achieve this, various selection functions have been suggested, and in our study, we have selected the Z score technique. To describe this function, we can regroup the various frequencies in a contingency table. An example is given in Table 1.

In this table, the letter *a* indicates the number of occurrences of the feature *f* in the subset S (e.g., corresponding to all positive sentences). The letter *b* denotes the number of features of the same feature *f* in the rest of the corpus, while $a + b$ is the total number of occurrences in the entire corpus C. Similarly, $a + c$ indicates the total number of features in S, including *f* and all other features (denoted $\neg f$).

	S	C-	C = S ∪ C-
f	*a*	*b*	$a + b$
$\neg f$	*c*	*d*	$c + d$
	$a + c$	$b + d$	$n = a+b+c+d$

Table 1. *Example of a contingency table*

The frequencies shown in Table 1 could be used to estimate various probabilities. For example, we might calculate the probability of the occurrence of the feature *f* in the entire corpus C as $P(f) = (a+b) / n$ or the probability of finding a feature belonging to the subset S as $P(S) = (a+c) / n$.

To define the discrimination power of feature *f*, we define a weight according to Muller's method (Muller, 1992). We assume that the distribution of the feature *f* follows a binomial distribution with the parameters P(*f*) and *n'*, where parameter P(*f*) represents the probability of drawing the feature *f* from the corpus C (estimated as $(a+b) / n$). If we repeat this drawing $n' = a+c$ times, we can expect having $P(f) \cdot n'$ times the feature *f* in the subset S. On the other hand, Table 1 indicates that we observe *a* times the feature *f* in S. A large difference between *a* and the product $P(f) \cdot n'$ is clearly an indication that the distribution of the term *f* differs in the subset S and in C-.

To obtain a clear decision rule, we suggest computing the standardized Z score attached to each feature *f* as shown in Equation [1], where P(*f*)·*n'* is the mean of a binomial distribution and P(*f*)·(1-P(*f*))·*n'* is its variance.

$$Zscore(f) = \frac{a - n' \cdot P(f)}{\sqrt{n' \cdot P(f) \cdot (1 - P(f))}} \qquad [1]$$

Table 2 depicts the ten highest Z score values for negative and positive categories for the movie review dataset. As shown in this table, usually these features are quite descriptive and salient to each category analyzed.

	Positive Feature	**Z score**	**Negative Feature**	**Z score**
1	powerful	5.864	no	6.740
2	entertaining	5.611	boring	6.599
3	touching	5.562	or	6.289
4	enjoyable	5.536	so	6.188
5	best	5.519	feels	5.831
6	culture	5.492	worst	5.790
7	with	5.277	only	5.571
8	solid	5.179	like	5.561
9	film	5.148	tv	5.521
10	both	5.010	heavy	5.390

Table 2. *Top ten highest Z score values for both positive and negative categories*

Analysing the feature list above for each of the categories, we can notice the terms "with", "or", "so" that do not carry any sentiment by themselves. Traditionally in the IR this kind of terms are removed from the corpus. We let the classification model take care of these terms. If the frequency of these terms is high in both categories, their scores will not highly influence the classification accuracy.

In our opinion the presence of these words is due to two reasons. First, they represent a part of phraseological expressions or constructions overused in a specific category, for example "agree with", "satisfied with". Second, since we use the calculation of the statistical scores when training our model on two classes, very frequent elements in a category will receive higher Z scores even if the percentage of all frequencies distributed over the two categories is the same.

3.2. Classification Model

Based solely on confident features, we have a simple text representation that ignores term neighbors. For example, the meaning of the expressions "*give"* and

"*give up"* is clearly different. To take into account this local proximity, we extract the neighbors of each confident feature (two terms before and two after). The bigram indexing scheme does not always capture these expressions since a lot of times they may contain modifiers or other words in-between. For example: "*It acknowledges and celebrates their cheesiness as the reason why people get a kick out of watching them today*." Here "*kick out*" is an idiomatic expression that has a different meaning and sentiment polarity than the word "*kick*". To capture expressions related to a confident feature, we use the Information Gain (IG) ratio (also called *expected mutual information*). Similar to Table 1, we can introduce a contingency table for two features f_{conf}, f_n as shown in Table 2.

	f_n	$\neg f_n$	
f_{conf}	a	b	$a + b$
$\neg f_{conf}$	c	d	$c + d$
	$a + c$	$b + d$	$n = a+b+c+d$

Table 3. *Example of a contingency table for two features f_{conf} and f_n*

Using the above notation we can estimate $P(f_{conf}, f_n) = a / n$, $P(f_{conf}) = (a+b) / n$ or $P(\neg f_{conf}) = (c+d) / n$. To compute the IG ratio between two features, we use the following formula:

$$IG(f_{conf}, f_n) = \sum_{f_r \in \{f_{conf}, \neg f_{conf}\}} \sum_{f_s \in \{f_n, \neg f_n\}} P(f_r, f_s) \cdot \log_2 \left(P(f_r, f_s) \Big/ P(f_r) \cdot P(f_s) \right) \quad [2]$$

When the IG ratio value is close to zero, we cannot detect a significant association between the two features. A positive value tends to indicate an association between the two terms.

To compare the probability distributions of the term in the two classification categories respectively, we use the variation of the Kullback-Leibler (KL) divergence that we calculate using the term probability distribution in the whole corpus C and in the specific category S:

$$D_{KL}(C \,||\, S_{\{pos,neg\}}) = \sum_k P(k) \ln \left(\frac{P(k)}{Q(k)} \right) \quad [3]$$

Based on this information, we can then compute the three following quantities for each sentence and for the two possible categories (opinionated vs. factual or positive vs. negative).

$$\sum_{f_i \in Conf} Z\,score(f_i) \quad \text{The sum of Z score of all confident features}$$
$$\sum_{f_j \in Neighbor(f_i)} IG(f_j, f_i) \quad \text{The sum of IG scores for neighbors of confident features} \qquad [4]$$
$$\sum_{f_i \in \neg Conf} Z\,score(f_i) \quad \text{The sum of Z score of all non confident features}$$

When computing these elements, we need to take into account negations. If the preceding word is a negation modifier, such as *not*, *no*, or *but*, we add the term weight to the opposite category.

The classification model is computed in the following way. If at least one confident feature is present in the sentence, we classify the sentence according to the sum of the confident feature scores for each category (see example in Table 4). If we look at the statistics, we can notice that the number of confident scores is much less than the total number of distinct terms per corpus. Thus, we encounter a lot of times a situation where we need to classify a sentence without the evidence of the confident feature set. In this case we use the weighted harmonic mean (*H)* of the Z score and KL divergence and the IG scores for neighbors of confident features (if present) in order to obtain a score for each classification category respectively. The weights attributed to each of the sum terms in Equation [5] are determined empirically. In our experiments we obtained the best results giving more weight to the KL divergence term. The final classification model uses the above variables as described by Equation [5].

$$\pi(c_i) = H\left(\sum_{f_k \in C_i \cap \neg Conf} Z\,Score(f_k), \sum_{f_k \in C_i} D_{KL}(f_k), \sum_{f_k \in C_i \cap Neighbor(f_j)} IG(f_k, f_j) \right) \qquad [5]$$

where c_i denotes the corresponding category. For each sentence, we can compute the $\pi(c_i)$ corresponding to the two possible categories and the final decision is to simply classify the sentence according to the category maximizing this formulation.

	Z score^{CONF}		**Z score**		**KL score**		**IG score**	
Feature/Category	POS	NEG	POS	NEG	POS	NEG	POS	NEG
magnific	-	-	0.38	-2.19	0.66	0.32	-	-
drama	-	-	-1.43	-7.7	3.67	3.22	-	-
well	-	-	-3.39	-8.17	4.62	4.36	0.07	-
worth	1.17	-	1.17	-7.68	2.73	1.91	-	-
track	-	1.03	-2.77	1.03	0.24	0.79	-	-
down	-	-	-7.72	-0.01	2.52	3.32	-	-

Table 4. *Statistical scores computed for each of the features in the example sentence 1 from the movie review dataset*

To illustrate the classification procedure, we present the sentence: "*Magnificent drama well worth tracking down*" from the movie review dataset. As we can see from the table we have two scores for the two confident features found in different categories: "worth" and "track". First, our model checks the sum of the confident scores and therefore classifies the sentence correctly as positive.

4. Experiments

4.1. Experimental Setup and Corpora Used

To evaluate our model, we used two benchmark datasets frequently used in sentiment classification. The first is the movie review polarity dataset[1] containing 5,331 positive and 5,331 negative sentences of movie reviews (Pang & Lee, 2005). The second test-collection is the subjectivity dataset containing 5,000 subjective (opinionated) and 5,000 objective (factual) sentences. It is a challenging task because the reviewers use a lot of metaphors, comparisons, and sometimes unclear language or references to other movies or situations.

In order to evaluate the classification performance, we compute the accuracy of the classifier using the 10-fold cross-validation. In this case, the training examples are never used in the test. This measure takes into account the effectiveness of the classification in both classes, positive and negative, or opinionated and factual, by counting the number of correct decisions divided by the number of cases.

4.2. Preprocessing

To represent each textual unit, we have used different strategies. First, the unigram model is based on a sequence of stems obtained with the Porter's stemmer. We also use the bigram indexing scheme. As a third approach, we noticed that some of the prepositions combined with the previous term in the sentence can change its meaning, and sometimes even its polarity. For example, "*take"* and "*take off"*, "*put"* and "*put up"*. Therefore, together with the experiments with unigram and bigram representations we implemented a new indexing scheme called WiseTokenizer.

This new text representation is obtained using the following procedure. All terms in the sentence are indexed separately, except terms that precede the prepositions that could change the meaning of the verb. These are some examples of the prepositions: *out*, *of*, *back*, *over*, *on*, *at*, *about*, *up*, *in*, *off*, *through*, *along*, *by*,

1. Freely available at http://www.cs.cornell.edu/people/pabo/movie-review-data

behind. This indexing scheme represents a hybrid of the unigram and bigram model.

Data	**Movie Review Dataset**		**Subjectivity Dataset**	
	Pos.	**Neg.**	**Obj.**	**Subj.**
# of documents	5,331	5,331	5,000	5,000
# of terms	116,080	116,176	129,316	119,069
# of distinct terms	20,370	21,052	22,790	21,651
mean # of terms per sent.	21.77	21.79	25.86	23.81
mean # distinct terms per sent.	20.23	20.3	23.58	22.08
# of confident terms	1186	1031	2105	2415

Table 4. *Corpus statistics for movie review and subjectivity datasets*

5. Evaluation results

5.1. Comparison with Other Methods

Table 5 presents the accuracy rates on both datasets based on the 10-fold cross-validation. From the results depicted in Table 5, we can see that the WiseTokenizer scheme tends to perform better than bigram and unigram indexing schemes. This wiser indexing representation specifically captures possible compound verb constructions. We can also see that our model performed better on the subjectivity dataset than on movie review. According to the analysis of our experiment results, we do not commit an error when at least one of the confident features is present in the sentence. To some extent the number of confident features selected influences the classification accuracy. The performance result, in our opinion, indicates the difficulty of the particular dataset for the proposed model.

Approach	**Movie Review Dataset**	**Subjectivity Dataset**
WiseTokenizer	88.29%	94.64%
Unigram	85.75%	94.08%
Bigram	84.48%	92.92%

Table 5. *Results showing accuracies of the proposed classification model using different indexing schemes with 10-fold cross validation evaluation and stemming*

It is important to note that these results cannot be directly compared as some of the authors use a different number of folds for cross validation or different splits if the number of folds is the same. There are also differences in corpus preprocessing

techniques. Additionally, both datasets have previous versions where a classification unit was a snippet (a short paragraph with an average number of 668 terms per document), which, in our opinion, facilitates the classification task as there is more textual information provided.

Overall, the WiseTokenizer scheme outperformed the unigram and bigram tokenization schemes. The growth in the accuracy rate with the use of the WiseTokenizer could also be the indicator of the particularity of the dataset. As expected from previous experiments on the corpus (Pang & Lee, 2008), the bigram scheme yields in accuracy to the unigram scheme. The new suggested method gives state-of-the-art performance as most of the SVM-based approaches, showing lower accuracy compared with an approach involving human annotation as an added source of information for the learning model (Zaidan *et al*., 2007). In the last several years, the use of methods involving SVM for the opinion detection and sentiment analysis has reached a stable level of performance and does not improve much (Mullen & Collier, 2004, Whitelaw *et al.*, 2005). Moreover, the main drawbacks of the SVM method in comparison to our model are the computational complexity in the training phase and the inability to perform a clear failure analysis to determine where and why the proposed decision is incorrect. The proposed feature selection and classification model provides a simpler way to analyze the feature selection procedure and adapt the classification criteria.

5.2. Failure Analysis

In order to have a better understanding of our underlying classification scheme, we have conducted a failure analysis for the movie review dataset. When inspecting the sentences misclassified by our model, we can see that most of the classification errors are related to the underlying ambiguity of the natural language. In the following examples, we have first presented sentences that were incorrectly labeled as *negative*. In the second case, we can find sentences incorrectly classified as having a *positive sentiment* by our model.

As you can see, these reviews would be difficult for an automatic classification model in several ways. First the use of highly positive or negative words to express or intensify completely opposite polarity, as we can see in the first three examples.

Another concern is when the sentence (e.g., Sentence #4) does not contain any overtly negative features, but nevertheless expresses a negative opinion by the means of the verb *abridged*. The fifth sentence gives a weak clue of negativity with the use of terms *mechanical* and *seeming*, while containing a highly positive *charismatic* feature. The use of slang expressions, such as *bow-wow*, and a negative connotation of the term *promotion*, are also difficult to detect correctly. In this case, the first expression is quite infrequent in the corpus and the latter is mostly neutral by itself in its polarity (Sentence #6). The last sentence represents a subset of misclassified examples from our observation where one part of the sentence displays

an abundance of positive terms, while the other uses several negative terms. In the current case, the not so common phraseological expression *shooting blanks* is the main reason for the misclassification of this review. All these examples demonstrate the complexity of all natural language and the need for developing language specific heuristics to better capture phraseological expressions, contrasting statements, sarcasm, and allusions made by the writer.

Positive sentences classified as negative.
1. "Longley has constructed a remarkably coherent, horrifically vivid snapshot of those turbulent days."
2. "Romanek keeps the film constantly taut... reflecting the character's instability with a metaphorical visual style and an unnerving, heartbeat-like score."
3. "Compelling revenge thriller, though somewhat weakened by a miscast leading lady."
Negative sentences classified as positive.
4. "In the book-on-tape market, the film of "the kid stays in the picture" would be an abridged edition."
5. "A mechanical action-comedy whose seeming purpose is to market the charismatic Jackie Chan to even younger audiences."
6. "It's not so much a movie as a joint promotion for the national basketball association and teenaged rap and adolescent poster-boy lil' bow wow.'"
7. "Director Tom Dey demonstrated a knack for mixing action and idiosyncratic humor in his charming 2000 debut shanghai noon, but showtime's uninspired send-up of tv cop show cliches mostly leaves him shooting blanks."

6. Conclusion

In this article we suggest a novel method for overcoming the binary classification problem in sentiment and opinion classification. In the proposed procedure, the extraction and weighting of confident features are based on the Z score model, able to determine term specificity according to two or more categories. Based on the information gain measure, we suggest taking for the neighborhood of confident terms.

Based on two well-known test-collections in the domain (movie review and subjectivity), the suggested model is able to achieve comparable results to more classical methods such as SVM and Naïve Bayes. Based on a simple statistical approach, the proposed classification model was applied with success in two different contexts. Based on terms and their polarity, the decision taken by our model can be explained easily.

7. References

Dave, K., Lawrence, S., & Pennock, D.M. (2003). "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews". In *Proceedings of the Twelfth International World Wide Web conference*, pp. 519-528.

Forman, G. (2003). "An extensive empirical study of feature selection metrics for text classification". *The Journal of Machine Learning Research*, 3, pp. 1289-1305.

Liu, Y., Loh, H.T., Youcef-Toumi, K., & Tor, S.B. (2007). "Handling of imbalanced data in text classification: category-based term weights". *Natural Language Processing and Text Mining*, Springer, London, pp. 171-192.

Matsumoto, S., Takamura, H., & Okumura, M. (2005). "Sentiment classification using word sub-sequences and dependency sub-trees". In *Proceedings PAKDD*, pp. 301-311.

Mullen, T., & Collier, N. (2004). "Sentiment analysis using support vector machines with diverse information sources". In *Proceedings EMNLP'04*, pp. 412-418.

Muller, C. (1992). *Principes et méthodes de statistique lexicale*. Champion, Paris.

Paltoglou, G., & Thelwall, M. (2010). "A study of information retrieval weighting schemes for sentiment analysis", In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 1386-395.

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, vol. 2(1-2), pp. 1–135.

Pang, B., & Lee, L. (2005). "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales". In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pp. 115-124.

Pang, B., & Lee, L. (2004). "A sentimental education: Sentiment analysis using subjectivity summarization based on Minimum Cuts". In *Proceedings of the 42nd Annual Meeting of the Association of Computational Linguistics*.

Pang, B., Lee, L., & Vaithyanathan S. (2002). "Thumbs up? Sentiment classification using machine learning techniques". In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 79-86.

Raychev, V., Nakov, P. (2009). "Language-independent sentiment analysis using subjectivity and positional information". In *Proceedings of the International Conference RANLP-2009*, pp. 360-364.

Whitelaw C., Garg N., & Argamon, S. (2005). "Using appraisal groups for sentiment analysis". In *Proceedings of the 14th ACM CIKM*, pp. 625-631.

Yang, Y., & Pedersen, J.O. (1997). "A comparative study of feature selection in text categorization". In *Proceedings ICML*, pp. 412-420.

Zaidan. O.F., Eisner J., & Piatko, C.D. (2007). "Using annotator rationales to improve machine learning for text categorization". In *Proceedings of NAACLHLT*, pp. 260-267.

Zubaryeva, O., & Savoy, J. (2009). "Investigation in statistical language-independent approaches for opinion detection in English, Chinese and Japanese". In *Proceedings of the Third International Workshop on Cross Lingual Information Access: Addressing the Information Need of Multilingual Societies*, pp. 38-45.