Sentiment Analysis of Turkish Political News

Mesut KAYA[1,2], Güven FİDAN[2], Ismail H. Toroslu[1]

Department of Computer Engineering
Middle East Technical University, Ankara, Turkey
{e1502509, toroslu}@ceng.metu.edu.tr

R&D Department
AGMLab, Ankara, Turkey
{mesut.kaya, guven.fidan}@agmlab.com

Abstract— **In this paper, sentiment classification techniques are incorporated into the domain of political news from columns in different Turkish news sites. We compared four supervised machine learning algorithms of Naïve Bayes, Maximum Entropy, SVM and the character based N-Gram Language Model for sentiment classification of Turkish political columns. We also discussed in detail the problem of sentiment classification in the political news domain. We observe from empirical findings that the Maximum Entropy and N-Gram Language Model outperformed the SVM and Naïve Bayes. Using different features, all the approaches reached accuracies of 65% to 77%.**

Keywords- Sentiment Analysis, Turkish, Machine Learning, News Domain, NLP.

I. INTRODUCTION

The Internet has become a global forum where people express their opinions. With the recent explosive growth of the social media content on the Web, people post reviews of products on merchant sites and express their views about almost anything in their personal blogs, pages at social network sites like Facebook, Twitter, and Blogger. People are also eager to know what individual journalists, and columnists are thinking or feeling about subjects such as politicians, political parties and social issues. With the rapid growth in news groups it is possible to analyze opinions and feelings in news domain, too. For these reasons, sentiment analysis aiming to determine the attitude (sense, emotion, opinion etc.) of a speaker or a writer with respect to a specified topic has become a major area of interest in the field of NLP.

Various statistical and linguistic methods have been developed for the Sentiment Analysis of English texts for different domains. In the case of Turkish and other morphologically rich languages, however, sentiment analysis is a new field of research and not much work has been published in the field.

In this paper we examine the effectiveness of using machine learning techniques on Turkish sentiment analysis of news data (analyzing columns of Turkish columnists to understand whether they support or criticize a political party, politician, or social issue). A challenging aspect of this research is that sentiment analysis of the news domain is a difficult task (See "Section III"). Another challenging aspect is that statistical methods may not perform well on morphologically rich languages like Turkish. Therefore, in addition to presenting the results we obtained from different machine learning techniques, we try to discuss what more can be done in the sentiment classification of Turkish news data.

The contents of this paper are as follows: In Section II we review previous work related to our study. In Section III we explain the data we have used and in Section IV we describe the machine learning techniques used with the different features used. In Section V, we explain the experimental setup and the evaluation metrics used, and give results of the evaluations. Finally in Section VI we conclude the paper by discussing results and future work to be done.

II. RELATED WORK

Early research on sentiment classification of documents is based on cognitive linguistics models [1], [2], or on the construction of discriminant-word lexicons [3], [4], [5].

Some research on sentiment classification focuses on classifying the semantic orientation of words or phrases [6], [7].

Research that aims at distinguishing the author's polarity on certain topics from document level [8], [9], [10] to sentence level [11], [12] has been performed.

In the recent studies, subjectivity extraction has become the area of interest. Sentence-level subjectivity classifiers are trained and it is shown by using selected subjective sentences that only sentiment analysis gets better quality results [13].

Most previous works focus on the sentiment analysis of highly subjective texts, such as product or movie reviews. In this work we try to focus on the sentiment classification of columnists' work which is in news domain and which has received less attention, although some initial work on sentiment analysis in the news domain has been conducted recently [14], [15]. Mihelcea and Strapparava [16] tried to classify newspaper titles according to their emotion, and Godbole et al. [17] presented a system to assign scores indicating positive or negative opinion to each distinct entity in a text corpus collected from blogs and news.

One of the most major pieces of research on sentiment classification of news texts was conducted by Balahur and Steinberger [18]. The researchers tried to define the scope of the task; separating good from bad news content. They also tried to analyze clearly marked opinions that were explicitly expressed in news texts.

978-0-7695-4880-7/12 $26.00 © 2012 IEEE
DOI 10.1109/WI-IAT.2012.115

Tony Mullen and Robert Malouf [19] carried out statistical tests on political discussion group postings. This work was a preliminary work in the sentiment analysis of Informal Political Discourse.

Our work is new since no previous research has been conducted in the sentiment analysis of news texts for Turkish and there are just a few initial studies conducted in languages other than English. Besides, due to the lack of research, there are no results indicating how well different machine learning methods could perform in Turkish sentiment analysis.

III. NEWS DOMAIN

For our research we choose to work in the News domain, in the sentiment classification of political columns. Our motivation for applying sentiment analysis to news data is that there is not much previous work in the area. Most of the studies mentioned in the previous section used reviews (movie reviews, product reviews etc.), since the writers often summarize their sentiments in their reviews. Besides, it becomes easy to annotate training data as positive or negative since most of the web sites accepting user reviews also provide a rating system, i.e. a 0-5 scale star system. Moreover, reviews are short and relatively easy to analyze.

For news articles, although support or criticism is sometimes expressed, journalists or columnists express their opinions indirectly [18]. In most news data journalists use a clear language to give an impression of objectivity, and for the same reason they state their opinions indirectly by embedding statements in a more complex discourse or argument structure [20]. The main reason why we choose columnists' work is that, unlike most of the news data, in their columns they use subjective language to express their opinions. This results in either support or criticism about *a politician, a political party, a social issue* and so on. However, there are still some difficulties and problems encountered in the task of the sentiment classification of political columns.

Difficulty with annotating political columns as positive or negative is a challenging job since in the political news domain a sentence can be positive for one annotator but negative for another. For example consider the following Turkish sentence: "*Oysa 'A partisi' boyun eğmeye hiç niyetli görünmüyor*." (But, 'Party A' does not seem to give up.). This may be annotated as positive by an annotator supporting the political views of 'Party A', and negative by another annotator not supporting the party's views.

Another difficulty with using political columns is that columnists most of the time put positive criticism and negative criticism together. For example, a columnist supporting the political views of 'Party A' rather than 'Party B', writes positive criticism about 'Party A' and negative criticism about 'Party B' in the same column. Therefore, while annotating the columns we look at the overall sentiment.

Concerning the difficulties explained, we tried to collect columns demonstrating positive (support) or negative criticism about a person, political party, popular topic etc. Our data came from 6 different Turkish newspapers. We collected a total of 400 columns, 200 positive and 200 negative. The collected data was annotated by three native speakers of Turkish. The 400 collected and annotated columns are the ones for which the three annotators agreed about whether they showed positive or negative criticism, or none. During the data collection period we observed that most of the time columnists express criticism rather than support. Therefore, finding documents containing positive criticism was a difficult task. Another observation was that columnists sometimes write about 2-3 unrelated topics in the same column. Since we were looking at the overall sentiment of the column; we did not use such columns in our data set.

IV. MACHINE LEARNING METHODS

Our aim in this work is to examine the sentiment classification of columnists' columns by using machine learning techniques. We experimented with four different algorithms: the Support Vector Machine, the Naïve Bayes Classification, Maximum Entropy Classification and the N-gram based character Language Model.

For Naïve Bayes, SVM and Maximum Entropy Classification, we use the following bag-of-words framework:

Let $\{f1, f2, \dots, fm\}$ be a predefined set of m features that can appear in a document d. Let $ni(d)$ be the number of times fi occurs in a document d, then each document is represented as follows: $d = \big(n1(d), n2(d) \dots, nm(d)\big)$. For N-Gram, the technique used is explained in section "N-gram based character language model".

A. Naïve Bayes Classification

This method is often used in text classification because of its simplicity and speed. Basically, Naïve Bayes makes the assumption that features (words) are generated independently of word position. It assigns a given document d to the class:

$c^* = argmaxcP(c|d)$. We derive the Naïve Bayes Classifier as follows:

$$PNB(c|d) = \frac{P(c) \times (\prod_{i=1}^{m} P(fi|c)^{ni(d)})}{P(d)} \quad (1)$$

where fi's are features that appear in the document and $ni(d)$'s are the number of times features occur in the document.

B. SVM

We used the SVM[light1] package for training and testing, and the default parameters of the package. The idea behind SVM is as follows: in the training phase to find a hyper plane separating the document vectors in one class from those in the other and making the margin or separation as large as possible. In the testing phase the aim is to classify

[1] http://svmlight.joachims.org/

test instances based on which side of the hyperplane they fall on. Let us say that the hyperplane is represented as follows:

$$w = \sum_j \propto j \times cj \times dj, \qquad \propto j \geq 0, \tag{2}$$

w is a vector that represents the hyperplane. $cj \in \{-1,1\}$ (-1 for negative, 1 for positive) is the correct calls of document dj . $\propto j$'s are obtained by solving a dual optimization problem.

C. N-gram based character Language Model

The N-gram based character language model is derived from the N-gram language models. Instead of words, this model takes characters as the basic unit in the algorithm instead of words [21].

The model provides $p(s)$ defined for strings $s \in \sum^*$ over an alphabet of characters $\sum$. For a character c and string s the chain rule is:

$$p(sc) = p(s) \times p(c|s) \tag{3}$$

The N-gram Markovian assumption restricts the context to the previous $n-1$ characters, taking:

$$p\left(c_n \middle| s_{c_1 \dots c_{n-1}}\right) = p(c_n|c_1 \dots c_{n-1}) \tag{4}$$

Therefore the maximum likelihood estimator for N-grams is:

$$p(cs) = \frac{C(sc)}{\sum_c C(sc)} \tag{5}$$

where $C(sc)$ is the number of times the sequence sc observed in the training data and $\sum_c C(sc)$ is the number of single-character extensions of sc.

In this research, we used the Lingpipe [2] DynamicLMClassifier to test the effect of using the N-gram based character language model. It depends on a language model with a form of Witten-Bell smoothing [21]. We took N=8 in the N-gram statistical model. The results for taking N from 1 to 8 are given in Table III. The results are discussed in detail in Section V.

D. Maximum Entropy

We use Apache OpenNLP[3], which is a machine learning based toolkit for the processing of natural language text that has Maximum Entropy implementation.

To assign a given document d to a class c. Maximum Entropy uses the following exponential form:

$$P_{ME}(c|d) = \frac{1}{Z(d)} exp\left(\sum_i \lambda_{i,c} F_{i,c}(d,c)\right) \tag{6}$$

$Z(d)$ is a normalization function. $\lambda_{i,c}$'s are feature-weight parameters that are set so as to maximize the entropy of the induced distribution. The main goal is to choose the model making the fewest assumptions about the data while remaining consistent with it. Ten iterations of the improved iterative scaling algorithm [22] are performed for parameter training.

$$F_{i,c}(d,c') = \begin{cases} n_i(d), n_i(d) > 0 \; and \; c' = c \\ 0 \qquad\qquad\qquad otherwise \end{cases} \tag{7}$$

V. EVALUATION

A. Experimental Setup

We used documents from Turkish news columns as described in Section III. To prepare the documents we removed all HTML tags automatically from the original document format. No stopwords removal was used since most of the stopwords carry strong sentiment. To investigate the effects of stemming we used a Turkish stemmer developed under Zemberek [4], which is an open source natural language processing (NLP) framework for Turkic languages. In Zemberek, for stemming, a morphological parser basically finds the possible root and suffixes of a given word. We use the possible roots as features in the experiments. We use the token itself if the stemmer cannot find a possible stem. Notice that punctuation marks are not eliminated; they are treated as separate lexical items.

We conducted K-fold-cross-validation [23] in the experiments, adopting K to be 3. 200 positive and 200 negative news items are used to make a 3-fold cross validation in the data experiments. The data were partitioned randomly into three folds. On each round of experiments, we used 2-folds as the training data and the remaining fold as testing data.

For creating a baseline we use the claim [9] that there are certain words people tend to use to express strong sentiments, so that it might suffice to simply produce a list of such words to classify texts. We chose good indicators for positive and negative columns, 197 positive and 300 negative indicators.

Some of our selections are shown in Table I. We converted these words into simple decision procedures counting the number of the proposed positive and negative words in a given document. The accuracy for a simple baseline classifier technique is shown in Figure 1. It gives **59%** accuracy which might be considered as a baseline. We

[2] http://alias-i.com/lingpipe/

[3] http://incubator.apache.org/opennlp/

[4] http://code.google.com/p/zemberek/

TABLE I. TURKISH HUMAN EFFECTIVE WORDS, THEIR ENGLISH MEANINGS AND BASELINE RESULTS

Proposed Effective Words	Accuracy
Positive: cesaret (courage), teşekkür (appreciation), minnet (gratitude), gönülden (lief), kutluyorum (celebrating), emek (labour), kazanmak (to win), çabalarını (their efforts), işbirliği (collaboration), inşa (construction), soğukkanlı (cool)… **Negative:** yasak (forbidden), yüzsüz (impudent), arsız (sassy), zulüm (cruelty), vahim (desperate), inkarcı (denier), bedel (forfeit), tasfiye (clearance), utanç (shame), yalan (lie), suç (crime), taciz (abuse), cahillik (ignorance), ötekileştirme (otherization), yanıltıcı (misleading), yoksun (void of), hesaplaşma (reckoning), zorluklar (difficulties)…	59%

used this preliminary experiment as baseline for other experiments.

B. Performance Evaluations

To evaluate the performance of the sentiment classification of news data with four different Machine Learning methods, we adopted Accuracy, Precision and Recall metrics that are generally used in text categorization. These metric can be calculated according to the values in Table II and the following formulas:

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \quad (8)$$

$$Precision = \frac{tp}{tp + fp} \quad (9)$$

$$Recall = \frac{tp}{tp + fn} \quad (10)$$

C. Results

In this study we adopted four supervised machine learning algorithms from Naïve Bayes, SVM, Maximum Entropy and the character based N-Gram Language Model to Turkish political news columns.

We adapted several experiments, using different features:

1) Unigram: Unigrams are single words occuring in training and test data documents. For example, "demokrasi (democracy)", "yanıltıcı (misleading)" are example unigrams used in the experiments.

2) Stemmed Unigrams: The possible roots of single words in training and test data. To have the stems of the unigrams we use Zemberek framework's Turkish stemmer as explained in Experimental Setup section.

3) Unigram + Adjectives: Adjectives are treated as separate features and used as additional features with unigrams.

4) Unigram + Effective Words: Effective words shown in Table I are treated as separate features and used with unigrams.

TABLE II. CONTINGENCY TABLE FOR PERFORMANCE EVALUATIONS

	Actual Class (Expectation)	
Predicted Class (Observation)	tp (true positive)	fp(false positive)
	tn (true negative)	fn(false negative)

5) Bigram: Contiguous sequence of 2 words. For example, "adil değil(not fair)", "olumlu şekilde (positively)".

We also performed tests with frequency vs. presence of the used features and with stem of unigrams.

In the bag-of-words framework we represented a document as follows: $d = \left(n1(d), n2(d) \dots, nm(d)\right)$ for m different features. For presence we simply convert $ni(d)$ to 1 if it is greater than 0, otherwise it remains as 0.

Notice here that for the N-gram character based language model the usage of different features and frequency, and the presence of the features are different than for the 3 other algorithms. The steps followed are as follows:

- While creating the language models, input text is tokenized. For presence each different token is used only once, for frequency all the tokens are used.
- For stemmed unigrams, the stems of the tokens are given to language models as input data.
- For adjectives and unigrams, each token is identified as adjective, effective word or not. If they are adjectives or effective words the token is given to the language model twice.

Note that we took N=8 in the case of the N-Gram model. We tested this using unigrams as features to compare different values of N. The results can be found in Table 1. We can observe that by taking N=8 we obtained the best performance. It can be seen that after 7 it makes no difference in performance. According to the results presented in Table III we took N=8 in the remaining experiments.

The overall accuracies of four algorithms using different features are indicated in Table IV. Table VI shows the precision values of the experiments and in Table VII we show the Recall values. Figure 1 and Figure 2 indicate the graphics of accuracy values for frequency and presence information.

1) Feature frequency vs. presence: In order to investigate the effect of the frequency and presence information, we ran all the experiments twice, once with the presence of the features only and once with the frequencies of the features only.

As can be seen from the results, frequency information does not have a positive effect on the sentiment classification of the columns except for the Maximum Entropy Classification.

TABLE III. ACCURACY VALUES BY USING DIFFERENT N IN N-GRAM LANGUAGE MODEL

	N=1	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9	N=10	N=11
Accuracy	58.31	62.34	72.28	73.70	74.17	74.40	76.31	**76.54**	76.30	75.59	75.74

TABLE IV. ACCURACIES IN PERCENTAGE

	Features	frequency or presence?	NB	ME	N-gram Language Model	SVM
(1)	unigram	frequency	71.81	**75.85**	73.93	71.12
(2)	unigram	presence	72.05	74.88	**76.54**	74.88
(3)	bigram	frequency	70.86	69.92	**72.77**	64.48
(4)	bigram	presence	**72.05**	69.44	71.80	66.81
(5)	unigram+adjective	frequency	71.81	**75.59**	73.45	72.95
(6)	unigram+adjective	presence	72.29	74.88	76.30	**76.31**
(7)	unigram+effective	frequency	71.81	**76.31**	73.22	73.70
(8)	unigram+effective	presence	72.05	76.06	**76.78**	74.65
(9)	unigram(stemmed)	frequency	67.77	**76.78**	73.45	74.65
(10)	unigram(stemmed)	presence	67.99	74.88	75.35	**76.31**

In almost all of the experiments adopted with other algorithms, better performance was achieved by accounting feature presence not feature frequency. Although the difference is not significant in the case of algorithms other than Maximum Entropy, presence information provides better results.

Figure 1 shows the accuracy values of four different methods by using frequency of the features. We can see that Maximum Entropy performs better than other algorithms.

2) Results of Using Different Features: The classification accuracies resulting from using different features show that all of the experiments clearly surpass the baseline of 59% (See Table I).

In the experiments, in addition to unigrams, we also studied the use of bigrams as features. Lines 3 and 4 of Table IV show that bigram information does not improve performance beyond unigram information. We can see a 5% to 8% decrease for SVM, Maximum Entropy and the N-gram Language Model. In the case of Naïve Bayes there is not much difference between the results from using unigrams and bigrams. We can conclude that bigram information is not as useful as unigram information for the Sentiment Classification of news columns.

We also experimented effect of adjectives and effective words. To investigate the effects of adjectives and effective words carrying strong sentiment, *unigram + adjectives* and *unigram + effective words* were used as features.

One might expect that adjectives or effective words carry a great deal of information about the sentiment of a news column. The results indicate, however, that neither adjectives nor effective words provide a notably better performance.

We can see only a very small percentage improvement in the accuracies for different algorithms used, less than 1%. This may be due to the length of the documents used in the experiments. Unlike short movie reviews, political news columns are quite long. The frequency of adjectives and effective words used as features may be very small in the documents.

Using the stems of the unigrams as features makes no significant difference in the results either.

Comparing four different algorithms used in the experiments it can be observed that the N-gram character based Language Model outperforms other algorithms in the case of unigrams, at over 70%. Maximum Entropy is the second algorithm that performs well, although there is small difference between Naïve Bayes and Maximum Entropy. An interesting observation is that SVM performs worse than other algorithms. It has been shown that for Sentiment Classification of short reviews SVM performs better than Naïve Bayes and Maximum Entropy [9].

Comparing the 4 algorithms according to their runtime performance we observed that SVM and Naïve Bayes perform better than Maximum Entropy and the N-Gram character based Language Model. However, as we can see from the accuracy results, Naïve Bayes performs worse than the other algorithms, and SVM in most of the tests cannot perform as well as Maximum Entropy and the Language Model.

3) Turkish vs. English: In order to make a comparison between English and Turkish we made a further experiment. We collected some English columns from Hurriyet Daily News[5] and annotated 25 support and 25 criticism articles. The reason why we collected columns from this website is that, most of the English columns available there are written by Turkish columnists and they are political columns.

TABLE V. ACCURACIES FOR ENGLISH VS. TURKISH TEST

Test Type	NB	ME	N-gram	SVM
Turkish (Frequency)	**69.90**	64.12	74.07	69.68
English (Frequency)	67.59	**75.69**	**79.66**	**73.61**
Turkish (Presence)	63.66	**65.97**	69.90	67.82
English (Presence)	**71.53**	60.18	**73.83**	**69.91**

[5] http://www.hurriyetdailynews.com/

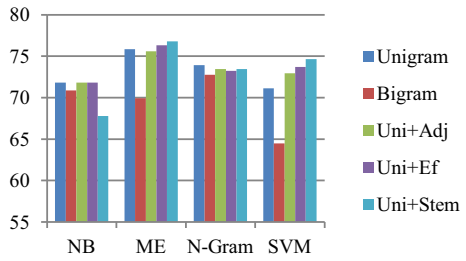

Figure 1. Accuracies for frequency experiments.

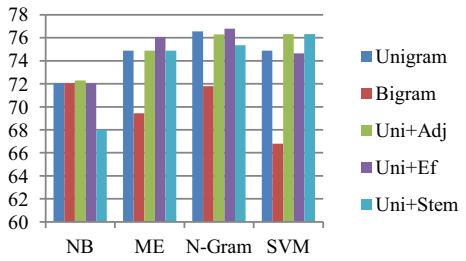

Figure 2. Accuracies for presence experiments.

Besides, the topics that are covered are similar to the Turkish political news data used in the previous experiments.

To make a coherent comparison we choose the same amount of Turkish data by using random sampling, 25 support and 25 criticism columns among 200 support and 200 criticism Turkish columns. We just tested for using unigram features for both frequency and the presence of the features. In all experiments 3-fold cross validation is used. The results are indicated in Table V.

It can be observed that in most of the experiments we obtain better accuracies for English. Throughout 8 experiments in only 2 of them accuracy values for Turkish are better, Naïve Bayes with frequency feature by 2% and Maximum Entropy with presence feature by 5%. In the remaining 6 experiments accuracy values for English data outperforms accuracy values for Turkish data by 2% to 11%. The reason may be the morphological differences between two languages. Turkish is a morphologically rich language that makes effective use of linguistic information such as the functional meaning of modal affixes, the semantic scope of negation terms like *no, not,* and the semantic classes of words. Therefore the usage of the frequency or presence features only may not be sufficient for Turkish data. We discuss in the next section what further work can be done in morphologic level.

TABLE VI. PRECISION VALUES IN PERCANTAGE

	Features	frequency or presence?	NB	ME	N-gram Language Model	SVM
(1)	unigram	frequency	70.04	76.31	73.02	**76.90**
(2)	unigram	presence	70.17	75.44	**75.93**	72.81
(3)	bigram	frequency	69.11	77.50	71.98	**77.65**
(4)	bigram	presence	70.05	**74.91**	72.92	64.96
(5)	unigram+adjective	frequency	70.04	**76.12**	72.61	74.92
(6)	unigram+adjective	presence	70.06	75.23	75.46	**75.64**
(7)	unigram+effective	frequency	70.16	**76.26**	72.18	75.93
(8)	unigram+effective	presence	70.08	**77.08**	76.02	72.15
(9)	unigram(stemmed)	frequency	66.73	76.34	71.24	**76.49**
(10)	unigram(stemmed)	presence	62.85	74.53	**74.81**	73.63

TABLE VII. RECALL VALUES IN PERCENTAGE

	Features	frequency or presence?	NB	ME	N-gram Language Model	SVM
(1)	unigram	frequency	76.79	74.92	**77.29**	60.24
(2)	unigram	presence	77.67	73.95	78.22	**80.13**
(3)	bigram	frequency	**75.85**	56.42	75.84	40.34
(4)	bigram	presence	**77.77**	58.81	71.12	73.95
(5)	unigram+adjective	frequency	**76.79**	77.75	76.78	69.25
(6)	unigram+adjective	presence	78.25	74.43	**78.69**	78.24
(7)	unigram+effective	frequency	76.32	**76.82**	76.71	70.16
(8)	unigram+effective	presence	77.27	77.08	78.69	**81.07**
(9)	unigram(stemmed)	frequency	72.04	77.74	**80.10**	71.09
(10)	unigram(stemmed)	presence	73.93	75.83	77.26	**82.01**

VI. DISCUSSION & FURTHER WORK

Although the results produced via machine learning techniques are quite good in comparison to the baseline results of 59% accuracy, our 76-77% accuracy is not good enough when compared to sentiment classification of reviews. For reviews, Pang et al. [9] and Boiy and Moens [24] obtained accuracies of up to 87.40% with SVM, Maximum Entropy and Naive Bayes. LingPipe for the N-Gram character based Language Model reported 81.5% accuracy for reviews by taking N=8. This can be explained by the difficulties explained in News Domain section.

In terms of relative performance, the N-gram based character Language Model and Maximum Entropy performed better than Naïve Bayes and SVM.

Unigram presences with the usage of adjectives turned out to be most effective for Naïve Bayes and SVM. Unigram presences with the usage of effective words were the most effective method for the N-gram Language Model. Unigram frequencies with the usage of stemming turned out to be most effective for Maximum Entropy. Although the usage of adjectives, effective words and stems provided better performance for different algorithms, the amount of improvement was not significant and so we cannot generalize their positive effect on sentiment classification in the news domain.

For future work more training data and test data must be collected in order to generalize the results for Turkish news domain. Different experiments should be conducted to compare the results in a more convenient way.

After extensive experiments, the effects of the Turkish language may be analyzed and different methods may be used for news text classification for Turkish. For example, contextual shifters and subjectivity clues may be used. Morphological Analysis may be used also. Concerning the difficulties and challenges explained in Section 3 related to News Domain, data can be preprocessed by eliminating objective sentences with subjectivity-objectivity detection.

As future work, Named Entity Recognition might be used to detect which columnists write about which political party or politician and so on. Moreover, columnists can be clustered according to their sentiment (support or criticism) towards topics decided by NER.

REFERENCES

[1] Marti Hearst. 1992. Direction-based text interpretation as an information access refinement. In Paul Jacobs, editor, *Text-based Intelligent Systems*. Lawrence Erlbraum Associates.

[2] Warren Sack. 1994. On the computation of point of view. In *Proc. of the Twelfth AAAI*, page 1488. Student abstract.

[3] Allison Huettner and Pero Subasic. 2000. Fuzzy typing for document management. In *ACL 2000 Companion Volume*: *Tutorial Abstracts and Demonstration Notes*, pages: 26-27.

[4] Sanjiv Das and Mike Chen. 2001. Yahoo! For Amazon. Extracting market sentiment from stock message boards. Alternation. In *Proc. of the 8th Asia Pacific Finance Association Annual Conference* (*APFA 2001*).

[5] Richard M. Tong. 2001. An operational system for detecting and tracking opinions in online discussion. Workshop note, *SIGIR 2001 Workshop on Operational Text Classification*.

[6] Vasileios Hatzivassiloglou and Kathleen McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proc. of the 35th ACL/8th EACL*, pages 174-181.

[7] Peter D. Turney and Michael L. Litnemm. 2002. Unsupervised learning of semantic orientation from a hundred-billion-word corpus. Technical Report EGB-1094, National Research Council Canada.

[8] Peter Turney. 2002. Thums up or thumbs down? Semantic Orientation applied to unsupervised classification of reviews. *In Proc. of ACL*.

[9] Pang, Bo, Lilian Lee, and Shivakumar Vaithyanatham. 2002. Thumbs up? Sentiment classification using machine learning techniques. *In Proceedings of the ACL-2002 conference on Empirical methods in natural language processing*, 10.

[10] Dave, K., S. Lawrance, and D. M. Pennock. 2003. Mining the peanut gallery: Opinion Extraction and semantic classification of product reviews. *In Proceedings of WWW-2003.*

[11] Hu, Minqing and Bing Liu. 2004. Mining and summarizing customer reviews. *In Proceedings of the KDD.*

[12] Kim, S. M., and E. Hovy. 2004. Determining the sentiment of opinions. *In Proceedings of the COLING*.

[13] Yu, H., and Hatzivassiloglou V. 2003. Towards answerring opinion questions: separating facts from opinions and identifying the polarity of opinion sentences. *In Proceedings of EMNLP*, 32.

[14] Fortuna Blaz, Carolina Galleguillos and Nello Cristianini. 2009. Detecting the bias in the media with statistical learning methods Text Mining: Theory and applications Yatlor and Francis Publisher.

[15] Balyaeva Evgenia, Erik van der Goot. 2009. News bias of online headlines across languages. The study of conflict between Russia and Georgia. Rhetorics of the media. *Conference Proceedings (2009) Lodz University Publishing House*.

[16] Strapparava, C. and Mihalcea, R. 2007. Semeval 2007 task 14: Affective Text. *In Proc. of ACL 2007*.

[17] N. Godbole, M. Srinivasaiah, and S. Skiena. 2007. Large-scale sentiment analysis for news and blogs. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*.

[18] Alexandra Belahur and Ralf Steinberger. 2009. Rethinking Sentiment Analysis in the news: from Theory to Practice and back. *WOMSA'09*, pages: 1-12.

[19] Mullen, T., Malouf, R. 2006. A preliminary investigation into sentiment analysis of informal political discourse. *In Proceedings of the AAAI symposium on computational approaches to analyzing weblogs,* pages: 159-162.

[20] Balahur, A., Steinberger, R., Kabadjov, M., Zavarella, V., Goot, E.V.D., & Halkia, M., et al. 2010. Sentiment Analysis in the news. *The 7th Conference on International Language Resources and Evaluation.*

[21] Carpenter, B. 2005. Scaling high order character language models to gigabytes. *In Proceedings of the 2005 association for computational linguistics software workshop,* pages: 1-14.

[22] Stephan Della Pietra, Vincent Della Pietra, and John Lafferty. 1997. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 19(4): 380-393.

[23] Kohavi, R. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the 14th International Joint Conference on Artificial Intelligenc,* 2(12): 1137-1143.

[24] Boiy, E. and Moens, M. 2009. A machine learning approach to sentiment analysis in multilingual Web texts. *Information Retrieval (2009),* 12:526-558.