FOUNDATIONS OF COMPUTING AND DECISION SCIENCES Vol. 38 (2013) No. 2

DOI: 10.2478/fcds-2013-0002

ISSN 0867-6356 e-ISSN 2300-3405

PERFORMANCE OF K-NEAREST NEIGHBORS ALGORITHM IN OPINION CLASSIFICATION¹

Krzysztof JEDRZEJEWSKI², Maurycy ZAMORSKI³

Abstract. This paper presents another approach for determining document's semantic orientation process. It includes a brief introduction describing the area of application of opinion mining, and some definitions useful in the field. The most commonly used methods are mentioned and some alternative ones are described. Experiment results are presented which show that kNN algorithm gives similar results to proportional algorithm.

Keywords: opinion mining, text mining, semantic orientation

1. Introduction

As global Internet network grows rapidly, it is commonly used by a vast number of people to exchange information. By information we mean almost anything, from newspaper articles, to video streaming. One of quite new phenomena is an advent of social network websites, discussion boards (forums), price and product comparators and much more, where users can share their opinions in certain areas. Many such pages implement mechanisms of valuation, where one can, apart from writing a comment, choose whether this comment is positive or negative – in the simplest case. The problem appears when we deal with the text only, without any additional information on the character of the statement, e.g. on discussion boards, or in raw comments to some newspaper article. In this situation the only solution is to process the text, preserving the semantics of the expression in such way, that it can be understood by a computer algorithm. After that, we can evaluate, with a certain probability, whether the processed phrase has a positive or negative value and, therefore, classify it to a positive or negative class of an examined data collection.

¹ This article is an extended version of the previous one entitled "Determining document's semantic orientation using kNN algorithm".

² Poznań University of Technology, Institute of Computing Science, Piotrowo 2, 60-965 Poznań, Poland, e-mail: Krzysztof.Jedrzejewski@cs.put.poznan.pl

³ Poznań University of Technology, Institute of Computing Science, Piotrowo 2, 60-965 Poznań, Poland, e-mail: Maurycy.Zamorski@cs.put.poznan.pl

Such knowledge is desirable not only for pure scientific reasons, but has a great value in marketing. Having the data of opinion changing over some period of time we can evaluate, for example, a lifecycle of a certain product, and, thus, predict income and profitability of an investment. Also, such data can be used to upgrade certain weaknesses of the product, which results in better quality and, therefore, larger profit.

To prove that sentiment analysis can provide a reliable source of information a survey has been made among the group of 2000 American citizens [5]. It shows that nowadays a large percentage of people are taking into consideration opinions of others in the process of searching for the best product to buy. The results indicate that:

- 81% of Internet users have done an online research on a product at least once;
- 20% of Internet users do online research on a product on a daily basis;
- 32% of Internet users have provided a rating on a product;
- 30% of Internet users have posted a comment or a review regarding a product;
- Consumers are willing to pay 20% 99% more for the high rated product (5 per 5 stars).

The latter information seems to be the most significant from a marketing point of view. This shows that opinions can shape not only the amount of product sold, but also the price.

Other areas where this kind of data is useful are e.g. political sciences. We can gain much information from opinions on certain topics and treat them as a part of social consultation process and even more. By continuous monitoring of public opinion on certain topics we can implement complex social systems with feedback loop, concerning e.g. law regulations or such. In the survey of 2500 American adult citizens concerning 2006 American elections [10] about a quarter of the respondents said that they were looking online to get perspectives from within or outside their community. Also, 28% said that most sites they view shared their point of view, but another 29% said that most of sites did not share their point of view. Only 8% of the surveyed posted their own comment on the topic. It proves that marketing sciences are not the only place where sentiment analysis can be applied in practice.

2. Related work

Determining document's semantic orientation is one of the tasks performed in the area of opinion mining. Opinion mining is a new scientific domain, strongly related to data mining and to natural language processing or machine learning. Opinion mining methods can be approached in several different ways.

- a) Supervised learning methods [3][4] methods using training sets as a starting point for comparison and evaluation of performance. In [3] the overall process is split into four main stages: information retrieval from a review site, feature selection (tokenizing, initial stats, thresholding, language processing, substitution, collocation), using scores for classification and performance check. Paper [4] provides a method for predicting adjectives' semantic orientation, also those, which alone cannot be classified to a positive or negative class.
- b) Unsupervised learning methods [6][12][13] methods without training set, mainly used for grouping objects and cluster analysis. [6] describes a feature-based opinion summarization of customer reviews. This task is performed in two steps: (i)

identifying the features of a product that the consumers commented on and ranking them in accordance to the frequencies of their appearance in reviews; (ii) identifying how many customers reviewed each feature positively or negatively. Paper [12], however, introduces an algorithm for unsupervised learning of semantic orientation from extremely large corpora by issuing queries to a Web search engine and using pointwise mutual information to analyze the results. In [13] there is a proposition of SO-PMI algorithm variation for Japanese weblog opinion. SO-PMI is an unsupervised algorithm that works good for English language documents, but this example shows that those algorithms can work good or even better for other languages.

c) information retrieval methods [9][16] – methods of retrieving information from documents par excellence. In [9] authors are trying to compensate the subjectivity of reviews by grouping them by product's features. Proposed method consists of four stages: identification of product features, identification of opinions regarding these features, determining the polarity of opinions and ranking opinions by their linguistic strength. Paper [16] describes opinion retrieval algorithm which finds topic relevant documents from a base set, classifies them and assigns ranks based on their relevance to the query.

In this paper we strongly relate to scoring algorithm described in [7], where proportional method was used in process of determining document's semantic orientation. That method is being based on another approach described in [3], which we refer to as the score method.

There are also other works concerning opinion mining [3][15] describing conceptions to deal with documents modeled as sets of words or vectors, and many more [8].

3. Main definitions

Before we get to kNN method description there is a need to explain some definitions crucial in understanding the entire concept:

- a) document whole examined set of words, e.g. a statement or a collection of statements on a certain topic;
- b) term a single word, being a component of a document; term may be also after lemmatization or stemming – processes of identifying core of a word, without any grammatical changes;
- c) n-gram a part of a text containing n characters, including white spaces;
- d) token used interchangeably, means a word or an n-gram;
- e) training set set of raw, unchanged text documents with classes assigned arbitrarily, in most cases by a human;
- f) positive / negative class describes whether a document contains a positive or negative opinion on a certain topic.

4. Basic concepts

Document's semantic orientation is often calculated by aggregation of the level of relationship between the words found in this document and a positive or negative class.

$$\gamma(d) = \begin{cases} C_P, \ eval(d) > 0\\ C_N, \ eval(d) < 0 \end{cases}$$
(1)

where

$$eval(d) = \frac{\sum_{t_i \in d} score(t_i)}{|d|}$$
(2)

or

$$eval(d) = \sum_{t_i \in d} score(t_i)$$
 (3)

where t_i is the i-th term of the document d, |d| is the number of terms in a document d, C_P and C_N are positive and negative classes, *score()* is a function assigning positive or negative values to terms, depending on their relation with the suitable class. Such assignment is based partly on intuition. We assume, that in positively marked documents there is the majority of words which alone may be considered as positive, e.g. "good", "well", "remarkable" and also words which are more often present in documents representing certain class, but alone are neutral. However such assumption is only good for simple sentences. More elaborate ones may contain greater number of negative words than positive, e.g. "I would lie if I wrote that this product is bad", and still be considered as positive one. Unfortunately those kinds of algorithms are more prone to errors in such kinds of statements. One of the methods utilizing the concept of supervised learning [3] has a score function in the following form:

$$score(t) = \frac{p(t|C_P) - p(t|C_N)}{p(t|C_P) + p(t|C_N)}$$

$$\tag{4}$$

where $p(t|C_P)$ and $p(t|C_N)$ are conditional probabilities of the occurrence of the term t in positive and negative class, respectively. These probabilities can be approximated by term occurrence frequencies in the training set.

The most positive terms are those which occur significantly more frequent in positive collection than in negative one. Likewise, the most negative terms are the most frequent ones in negative collections.

The disadvantage of the solution described above is that it assigns extreme values of 1 or -1 when a term occurs only in one class. Such situation takes place mostly in case of terms that are very rare in a language. To minimize the impact of this problem different approach based on pointwise algorithm was introduced in [7]. It is based on the ratio of

term occurrence frequency in documents assigned to a positive and negative classes. The score function is as follows:

$$score(t) = \begin{cases} p_t - 1 & \text{, iff } p_i \ge 1\\ -\left(\frac{1}{p_t} - 1\right), \text{ iff } p_i < 1 \end{cases}$$
(5)

where

$$p_t = \frac{p(t|C_P) + \varepsilon}{p(t|C_N) + \varepsilon} \tag{6}$$

where, p_t is the raw semantic orientation of the term t, $p(t|C_P)$ and $p(t|C_N)$ are conditional probabilities of occurrences of the term t in documents from positive and negative classes, respectively, and ε is a small positive control value for terms that appear in only one class, that prevents from division by zero and causes rare terms to have assigned value closer to 0 than those that occur frequently. Minimal or maximal is not limited. The most negative or positive values are assigned to tokens occurring often in one set and not occurring in the other one.

To the method of calculation described by formula (4) we refer to as the score method, and to the one described by (5) we refer to as the proportional method.

5. Our approach

The method proposed in this paper for determining document's semantic orientation is the use of *k*-nearest neighbor (*k*NN) algorithm, that assigns classified object to the class most common in the collection of *k* examples from training set \mathbb{E} , that are most similar:

$$\gamma(d) = \begin{cases} C_P, \ iff \ |SIM_k(d) \cap C_P| > |SIM_k(d) \cap C_N| \\ C_N, \ iff \ |SIM_k(d) \cap C_P| < |SIM_k(d) \cap C_N| \end{cases}$$
(7)

where γ is semantic orientation of a document d, C_P and C_N are positive and negative classes, respectively, $SIM_k(d)$ is the nearest neighborhood of document d, i.e. training set's subset, which cardinality is equal to k, and

$$\overset{\forall}{e \in SIM_k(d)} \overset{\forall}{e' \in \mathbb{E} \setminus SIM_k(d)} sim(d, e) \ge sim(d, e')$$
(8)

where \mathbb{E} is a training set and sim(d,e) denotes a function measuring similarity between documents d and e.

In addition to the classic version of kNN algorithm, we propose in this paper two concepts inspired by this approach.

The first algorithm, Amplified Neighbors' Similarities Sum (NSS), takes into account the level of similarity between classified document and examples in its nearest neighborhood in \mathbb{E} and the disproportion between number of examples representing positive and negative class in the training set.

$$\gamma(d) = \begin{cases} C_P, & iff eval(d) > 0\\ C_N, & iff eval(d) < 0 \end{cases}$$
(9)

where C_P and C_N are positive and negative classes, respectively, and *eval* (d) is a function evaluating document, expressed by the formula:

$$eval(d) = \sum_{e \in SIM_k(d)} (sim(d, e) \times pol(e) \times disp(e))$$
(10)

where sim(d,e) denotes a function measuring similarity between documents d and e, and $SIM_k(d)$ is the nearest neighborhood of document d, and

$$pol(e) = \begin{cases} 1, iff \ \mathbb{C}(e) = C_P \\ -1, iff \ \mathbb{C}(e) = C_N \end{cases}$$
(11)

$$disp(e) = \begin{cases} 1, iff \ \mathbb{C}(e) = C_P \\ \frac{|C_P|}{|C_N|}, iff \ \mathbb{C}(e) = C_N \end{cases}$$
(12)

where $\mathbb{C}(e)$ denotes class represented by document e – positive or negative, and $|C_P|$ and $|C_N|$ denotes numbers of examples representing positive and negative class in the training set. The aim of disp(e) function is to compensate disproportion of cardinalities of positive and negative learning sets.

Our second approach, Amplified Similarities Sum (ASS), is similar to the one above. The only difference is that we treat the whole training set as the neighborhood, i.e. $k = \infty$. Therefore, evaluating function becomes:

$$eval(d) = \sum_{e \in \mathbb{E}} (sim(d, e) \times pol(e) \times disp(e))$$
(13)

where sim(d,e) denotes a function measuring similarity between documents d and e, and E denotes the training set.

To calculate similarity between documents we use well-known TF-IDF scheme [11] with cosine similarity. In that model documents are represented as vectors. Similarity is calculated as a cosine of the angle between representations of two documents.

$$TF - IDF_{t,e,\mathbb{E}} = TF_{t,e} \times IDF_{t,\mathbb{E}}$$
(14)

where

$$IDF_{t,\mathbb{E}} = \log\left(\frac{|\mathbb{E}|}{|\{e:e \in \mathbb{E} \land t \in e\}|}\right)$$
(15)

 $TF_{t,e}$ is a frequency of the term *t* in the example *e* from the training set \mathbb{E} , and $IDF_{t,\mathbb{E}}$ is a measure of rarity of the term across the whole training set.

Since IDF measure prefers rarely occurring terms, it may not be best suited for Opinion Mining algorithms, e.g. in corpus containing opinions of the users concerning phones they are using; brand names like *Samsung*, *Nokia* or *HTC* would occur less often, than opinion specific words like *good* or *ugly*. So, brand name would have higher IDF value, thus it would become more important. Therefore, we propose replacement of IDF value with the absolute value of term evaluation consistent with score or proportional method [7].

$$TE_{t,C_P,C_N} = \left| score(t)_{C_P,C_N} \right| \tag{16}$$

6. Experiments

6.1. Test set

The main objective of experiments was to test the accuracy of the classification algorithm proposed in Section 5. We used collections of opinions harvested from the site *Znany lekarz*, which gathers opinions about physicians. Each opinion is linked to a grade on a scale from 1 to 6. We have assumed that opinions associated with grades 1 and 2 are negative, and opinions with grades 5 and 6 indicate a positive feedback. The dataset contains 2380 negative opinions and 11 764 positive opinions.

Additionally, we have created second data set, containing equal number of positive and negative documents, by removing the 9384 randomly chosen positive opinions from the collection described above. The resulting set contained 2380 positive and 2380 negative opinions.

Further in this paper we refer to the first dataset as *lekarz* and to the second one as *lekarz_eq*.

6.2. Performance measures

To evaluate the effectiveness of the classification we have used two measures. First one is a well-known classification accuracy (A) described by equation:

$$A = \frac{t_p + t_n}{t_p + t_p + t_n + f_n} \tag{17}$$

where t_p and f_p are true positives and false positives (i.e., numbers of positive examples from the test set classified correctly and incorrectly), and t_n and f_n are true negatives and false negatives (i.e., numbers of negative examples from the test set classified correctly and incorrectly). In addition, we have used binary classification quality (Q) [7], which is similar to the F1 measure, and takes into account precision and recall achieved in both classes. This measure is expressed by equations:

$$Q = \begin{cases} \frac{4}{\frac{1}{rec_{P}} + \frac{1}{rec_{N}} + \frac{1}{prec_{P}} + \frac{1}{prec_{N}}}, & \text{iff } 0 \notin \{rec_{P}, rec_{N}, prec_{P}, prec_{N}\} \\ 0, & \text{iff } 0 \in \{rec_{P}, rec_{N}, prec_{P}, prec_{N}\} \end{cases}$$
(188)

where

$$prec_P = \frac{t_p}{t_p + f_p} \quad prec_N = \frac{t_n}{t_n + f_n} \tag{199, 20}$$

$$rec_{p} = \frac{t_{p}}{t_{p}+f_{n}} \quad rec_{N} = \frac{t_{n}}{t_{n}+f_{p}}$$
(21, 22)

where t_p , f_p , t_n , f_n are the same values as in the definition of classification accuracy.

The quality measure is used because it is not susceptible to imbalance in a size of classes. For example, if the positive class has 90 elements and the negative has 10 elements, than the algorithm assigning every object to the positive class has 90% accuracy. Such behavior may be confusing and counterintuitive.

6.3. Experiment setup

We have performed the 10-fold cross-validation experiments with document representations based on terms and n-grams. In our experiments on classification using term representation we have performed tests using lemmatization, with *morfologik-stemming* [14], stemming, with *Stempel* [2][12], and without using any text pre-processing method. We also have tested the impact of removal of stop-words and rarely occurring words. We have derived the stop-word list from Polish *Wikipedia* [1]. As rarely occurring words we treated those that appeared in fewer documents than β :

$$\beta = \left\lfloor \frac{|\mathcal{C}_*|}{|\mathcal{C}_{\#}|} \right\rfloor + 2 \tag{23}$$

where C* denotes the majority class and C# denotes the minority class in the training set. In experiments on n-gram representation we have tested impact of replacement of IDF value, with a value assigned to a token by proportional or score method, on classification performance. Value of 2 is added to lower chance of treating rarely occurring terms associated equally with positive and negative class as a token strongly connected to the one of these classes. When cardinalities of both sets are equal or almost equal, this leads to $\beta = 3$. If β would be equal 2 than with probability of 50% term would have assigned estimation equal 0, and with probabilities of 50% absolute value would be equal 2 (score(t) = 2 or -2). We have also run experiments in which we removed tokens occurring in fewer documents than β threshold. All experiments were performed on texts converted to lower case. As size of neighborhood *k* we have assumed 11. Since average length of a word in Polish is approximately 6 and we aimed our algorithm to operate on word fragments, we have assumed 5 as length of n-grams.

7. Results

In this Section we present the results obtained by running all combinations of the test described in Section 6. The results show that replacement of IDF value with values assigned by score or proportional method tends to increase the quality and accuracy of classification. In most cases the removal of rare tokens leads to slight improvement of classification efficiency.

In both, Table 1 and Table 2, Q and A are considered better when they have lower value.

7.1. Experiments on term representation

Table 1 shows that performance of classification is worsened when stemming or lemmatization is performed. We see the cause of this behavior in properties of the Polish language, which has a rich grammar [7] Various inflected forms of words may have different probability of occurrence in each class, so changing it to a basic form reduces the amount of information. Removal of stop words has minimal and ambiguous influence on classification performance. Removal of rarely occurring terms slightly decreases values of quality and accuracy of classification. Also, proportional method provided with better results rather than pointwise method. This suggests that the effect of noise caused by the inclusion of terms for which we have little information is smaller than the effect of reduction of the information caused by their removal.

Table 1. Sums of ranks from descending rankings of algorithm configurations (Data set X [ASS, kNN, NSS] X preprocessing method X term scoring strategy X removal (or not) of stopwords) based on terms, calculated independently for each set based on values of quality Q and accuracy A. Value in each row is the sum of all ranking positions assigned to algorithm configurations. β denotes removal of tokens occurring in less documents than β .

Configuration element value	Q	А
ASS	3456	3699
kNN	2481	1861
NSS	2253	2630
No pre-processing	2234	2317
Stemming	2588	2580
Lemmatization	3368	3293
IDF	2430	2361
Proportional method	1253	1215
Prop. method + β	1422	1495
Score method	1508	1489
Score method + β	1577	1630
w/o stop-words	4658	4523
with stop-words	3532	3667



Figure 1. Average values of classification quality, achieved during the tests using a document representation based on terms. P - proportional method, S - score method, PF, SF - proportional and score method respectively with removal of terms occurring in fewer documents than β .



Figure 2. Average values of classification accuracy, achieved during the tests using a document representation based on terms.

7.2. Experiments on n-gram representation

As in the case of representation based on terms, the removal of rarely occurring terms slightly decreases the values of measures of classification performance.

Table 2. Sums of ranks from descending rankings of algorithm configurations (Data set X [ASS, kNN, NSS] X term scoring strategy) based on n-grams, calculated independently for each set based on values of quality Q and accuracy A. Value in each row is the sum of all ranking positions assigned to algorithm configurations. β denotes removal of tokens occurring in less documents than β .

Configuration element value	Q	А
ASS	101	109
kNN	71	57
NSS	68	74
IDF	71	69
Proportional method	42	39
Prop. method + β	48	45
Score method	39	43
Score method + β	40	44



Figure 3. Average values of classification quality, achieved during the tests using a document representation based on n-grams.



Figure 4. Average values of classification accuracy, achieved during the tests using a document representation based on n-grams.

Experiments performed using both, term and n-gram, document representations show better classification performance when IDF value was replaced by values assigned to a token by the score or proportional methods. NSS gives better results than classic *k*NN according to the quality of classification measure and worse according to the accuracy measure. This difference is due to the susceptibility of classical *k*NN methods for disproportion between the sizes of the positive and negative classes in the training set.



7.3. Impact of the scoring strategy

Figure 5. Distribution of values assigned to the tokens by the score method. Dataset *lekarz*. 5 character long n-grams. No tokens were removed.



Figure 7. Distribution of values assigned to the tokens by the proportional method. Dataset *lekarz*. 5 character long n-grams. No tokens were removed.



Figure 6. Distribution of values assigned to the tokens by the score method. Dataset *lekarz*. 5 character long n-grams. Tokens occurring in less than β documents were removed.



Figure 8. Distribution of values assigned to the tokens by the proportional method. Dataset *lekarz*. 5 character long n-grams. Tokens occurring in less than β documents were removed.

We find similarity of the results obtained for score and proportional method interesting. In particular due to the occurrence of predicted defects in the distribution of values assigned by the score method (Figures 5, 6, 7 and 8). It appears the logic of kNN algorithms compensates the problem mentioned above.

The removal of tokens occurring in less than β documents makes distribution of values assigned closer to the normal distribution (Figures 6 and 8), but in most cases worsens the efficiency of classification (Figures 1, 2, 3 and 4). We believe that the reason for this behavior is the significant reduction of the volume of information the classification algorithm processes.

8. Conclusions

To estimate the general performance of kNN-based algorithms there is a need to compare it to results achieved by methods described in [7]. Results are shown in Table 3. Best Q and Best A are the best individual results for combination of other algorithm configuration elements giving best results, not shown in table. Higher values are considered better.

Data collection	Representation type	Algorithm family	Best Q	Best A
lekarz	term	Scoring	0,8195	0,8917
lekarz_eq	term	Scoring	0,8892	0,8893
lekarz	n-gram	Scoring	0,8817	0,9334
lekarz_eq	n-gram	Scoring	0,9124	0,9124
lekarz	term	kNN	0,8389	0,9198
lekarz_eq	term	kNN	0,8905	0,8905
lekarz	n-gram	kNN	0,8418	0,9231
lekarz_eq	n-gram	kNN	0,8837	0,8838

 Table 3.
 Best individual results in experiments in Section 7 and [7]

Table 3 indicates that the use of both kinds of algorithms -kNN and scoring [7] give similar results. However, kNN-based algorithms consume much more resources (CPU processing time and memory).

The highest values of quality and accuracy measures are almost the same in data collections with equal number of elements belonging to positive and negative class. This proves that quality measure is reliable and may be considered as primary measure for estimation of algorithm's performance. If such assumption is taken into account, we can observe that scoring-based algorithms that operate using n-gram representation give slightly better results in all cases.

References

[1] Stop listy. http://pl.wikipedia.org/wiki/Stop_listy. *Wikipedia, wolna encyklopedia,* Accessed 12 February 2010.

- [2] Białecki A, Stempel Algorithmic Stemmer for Polish Language, http://www.getopt.org/stempel/, Accessed 12 February 2010.
- [3] Dave K, Lawrence S, Pennock DM, Mining the peanut gallery: opinion extraction and semantic classification of product reviews, in: *Proceedings of the 12th international conference on World Wide Web*, New York, USA, 2003, 519–528.
- [4] Hatzivassiloglou V, McKeown KR, Predicting the semantic orientation of adjectives, *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and the 8th Conference of the European*, New Brunswick, Canada, 1997, 174-181.
- [5] Horrigan J, Online shopping, Pew Internet & American Life Project Report, 2008.
- [6] Hu M, Liu B, Mining opinion features in customer reviews, in: *Proceedings of the* 19th national conference on Artificial intelligence, 2004, 755-760.
- [7] Jędrzejewski K, Morzy M, Opinion Mining and Social Networks: a Promising Match, in: *First Workshop on Social Network Analysis in Applications SNAA 2011*. Kaohsiung, Taiwan, 2011.
- [8] Pang B, Lee L, Opinion Mining and Sentiment Analysis, Now Publishers inc, 2008.
- [9] Popescu AM, Entzioni O, Extracting Product Features and Opinions from Reviews, in: Kao A, Poteet SR (Eds). *Natural Language Processing and Text Mining*, Springer, London, 2007, 9-28.
- [10] Rainie L, Horrigan J, Election 2006 online, *Pew Internet & American Life Project Report*, 2007.
- [11] Salton G, Wong A, Yang CS, A vector space model for automatic indexing. Technical Report, New York, USA, 1974.
- [12] Turney PD, Littman ML, Unsupervised learning of semantic orientation from a Hundred-Billion-word corpus, 2002.
- [13] Wang G, Araki K, Modifying SO-PMI for Japanese Weblog Opinion Mining by Using a Balancing Factor and Detecting Neutral Expressions, in: *Proceedings of NAACL HLT 2007*, Companion Volume 2007, New York, USA, 2007, 189-192.
- [14] Weiss D, Miłkowski M, Morfologik-stemming, http://morfologik.blogspot.com/, Accessed 12 February 2010
- [15] Xu RF, Wong KF, Xia YQ, Coarse-Fine Opinion Mining WIA, in: NTCIR-7 MOAT Task. In. Proceedings of NTCIR-7 Workshop, Japan, 2008.
- [16] Zhang W, Yu C, Meng W, Opinion retrieval from blogs, in: *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, 2007, 831-840.

Presented at the 16th East-European Conference on Advances in Databases and Information Systems September, 17-20, 2012, Poznań, Poland