

Sentiment Miner: A Novel Unsupervised Framework for Aspect Detection from Customer Reviews

Ayoub Bagheri
University of Kashan
Iran
a.bagheri@ec.iut.ac.ir

Shiva Nadi
Utrecht University
Netherlands
nadi.shiva@gmail.com



ABSTRACT: *With the thriving of review sites on the Web, people have begun to express their opinions on a wide variety of products on several services. Sentiment analysis on entities of products thus becomes a rapid and effective way of assessing public opinion for business marketing. Sentiment analysis is the computational study of people's opinions, attitude, emotion or appraisal pertaining to topics, objects, products, services, organizations, individuals, events or any attributes of them. Aspect detection in sentiment analysis helps customers to effectively navigate product information by zooming in on the product features they are interested in. In this paper we introduce Sentiment Miner, a novel unsupervised framework for aspect detection problem in a sentiment analysis system for online customer reviews. Sentiment Miner tries to detect aspects using a bootstrapping algorithm based on pointwise mutual information measure. The proposed framework considers multi-word aspects as atomic terms by utilizing C-value method. Experimental results show that the Sentiment Miner improves the precision, recall and the F-score, and outperforms the state-of-the-art baseline approach.*

Keywords: Aspect Detection, Opinion Mining, Sentiment Analysis, Mutual Information

Received: 10 April 2018, Revised 23 May 2018, Accepted 29 May 2018

© 2018 DLINE. All Rights Reserved

DOI: 10.6025/jcl/2018/9/3/120-130

1. Introduction

In the recent decade, with thriving of review sites on the Web, people have begun to express their opinions on a wide variety of products on several services, thus sentiment analysis has received more and more attention between information retrieval and natural language processing researchers [1]. Up to now studies in sentiment analysis have covered a wide range of tasks, including opinion target identification or aspect detection [2]-[14], opinion word detection [2], [4], [14], opinion orientation identification [4] and sentiment classification on word, phrase, sentence and document level [1], [4], [15], [16], [17], [22].

Aspects are topics on which opinion are expressed. In the field of sentiment analysis, other names for aspect are: features, product features or opinion targets [1], [4], [5]. Aspects are important because without knowing them, the opinions expressed in a sentence or a review are of limited use. For example, in the review sentence “the volume level of the phone is not all that good”, “volume” is the aspect for which an opinion is expressed. Likewise aspect detection is critical to sentiment analysis, because its effectiveness dramatically affects the performance of opinion word detection and sentiment orientation identification. Therefore, in this study we concentrate on aspect detection for sentiment analysis.

Existing aspect detection methods can broadly be classified into two major approaches: supervised and unsupervised [1], [5]. Supervised aspect detection approaches require a set of pre-labeled training data. Although the supervised approaches can achieve reasonable effectiveness, building sufficient labeled data is often expensive and needs much human labor. Since unlabeled data are generally publicly available, it is desirable to develop a model that works with unlabeled data. Additionally due to variety and wide range of products and services being reviewed on the internet, supervised, domain-specific or language-dependent models are often not practical. Therefore the framework for the aspect detection must be robust and easily transferable between domains or languages.

In this paper, we present an unsupervised framework, Sentiment Miner, which addresses the core tasks necessary to detect aspects from review sentences in a sentiment analysis system. In the proposed model we use a novel bootstrapping algorithm. Our model requires no labeled training data or additional information. The model can easily be transform between domains or languages.

In the reminder of this paper, existing works on aspect detection in sentiment analysis will be given in Section 2. Section 3 reviews the association rule mining as a background approach in aspect detection. Section 4 describes the Sentiment Miner framework for the aspect detection. In this section, using C-value method and the iterative bootstrapping algorithm is discussed. Subsequently we describe our empirical evaluation and discuss important experimental results in Section 5. Finally we conclude with a summary and some future research directions in Section 6.

2. Related Work

Several methods have been proposed, mainly in the context of product review mining [1]-[14], [17]. The earliest attempt on aspect detection was based on the classic information extraction approach of using frequently occurring noun phrases presented by Hu and Liu [2]. Their work can be considered as the initiator work on aspect extraction from reviews. They use association rule mining (ARM) based on the Apriori algorithm to extract frequent itemsets as explicit product features, only in the form of noun phrases. Their approach works well in detecting aspects that are strongly associated with a single noun, but are less useful when aspects encompass many low-frequency terms. The proposed framework in our study uses more POS patterns to extract the candidates for aspect. Wei et al. [12] proposed a semantic-based product aspect extraction (SPE) method. Their approach begins with preprocessing task, and then employs the association rule mining to identify candidate product aspects. Afterward, on the basis of the list of positive and negative opinion words, the semantic-based refinement step identifies and then removes from the set of frequent aspects possible non-product aspects and opinion-irrelevant product aspects. The SPE approach relies primarily on frequency- and semantic-based extraction for the aspect detection, but in our study we use frequency-based and inter-connection information between the aspects and give more importance to multi-word aspects. Somprasertsri and Lalitrojwong's [13] proposed a supervised model for aspect detection by combining lexical and syntactic features with a maximum entropy technique. They extracted the learning features from an annotated corpus. Their approach uses a maximum entropy classifier for extracting aspects and includes the postprocessing step to discover the remaining aspects in the reviews by matching the list of extracted aspects against each word in the reviews.

Our proposed framework on aspect detection designed to be as unsupervised as possible, so as to make it transferable through different types of domains, as well as across languages. The motivation is to build a model to work on the characteristics of the words in reviews and interrelation information between them.

3. Association Rule Mining

In this section we introduce association rule mining, AEM, as a base-line approach for the aspect detection problem in sentiment analysis. Association rule mining is a popular approach for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using different measures of interestingness [1], [2].

Association rule mining is stated as follows [2]: Let $I = \{i_1, \dots, i_n\}$ be a set of items, and D be a set of transactions (the dataset). Each transaction consists of a subset of items in I .

An association rule is an implication of the form $X \rightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \phi$. The rule $X \rightarrow Y$ holds in D with confidence c if $c\%$ of transactions in D that support X also support Y . The rule has support s in D if $s\%$ of transactions in D contain $X \cup Y$. The task of mining association rules is to extract all association rules in D that have support and confidence greater than predefined thresholds, e.g. a user-specified minimum support and minimum confidence for thresholds. Apriori is a classic algorithm for association rule mining over transactional databases [1].

The pseudo code for the apriori algorithm is given below in Figure 1 for a transaction database T , and a support threshold of s . Usual set theoretic notation is employed, though note that T is a multiset. C_k is the candidate set for level k .

```

Apriori Algorithm Pseudocode:
function Apriori (T, s) {
    L1= {frequent items};
    for (k = 2; Lk-1 !=  $\phi$ ; k++) {
        Ck= candidates generated from Lk-1
        //that is Cartesian product  $L_{k-1} \times L_{k-1}$  and //eliminating
        any k-1 size itemset that is not
        //frequent
        for each transaction t in database do{
            increment the count of all candidates in Ck
            that are contained in t
             $L_k$  = candidates in  $C_k$  with s
        }//end for each
    }//end for
    return  $\bigcup_k L_k$ 
}

```

Figure 1. The pseudocode of apriori algorithm

As is common in association rule mining, given a set of itemsets (for instance, sets of words), the apriori algorithm attempts to find subsets which are common to at least a minimum number C of the itemsets. The apriori algorithm terminates when no further possible rules are found.

4. Sentiment Miner: the Proposed Framework

Figure 2 gives an overview of the proposed Framework, Sentiment Miner, used for detecting aspects in sentiment analysis. Below, we discuss the functions in the Sentiment Miner.

4.1 Candidate Generation

In this paper we focus on five POS (Part-Of-Speech) tags: *NN*, *JJ*, *DT*, *NNS* and *VBG*, where they are the tags for nouns, adjectives, determiners, plural nouns and verb gerunds respectively [18]. Additionally stemming is used to select one single form of a word instead of different forms [5]. Based on the observation that aspects are nouns, we extract combination of noun phrases and adjectives from review sentences. We use several POS patterns introduced in Table 1.

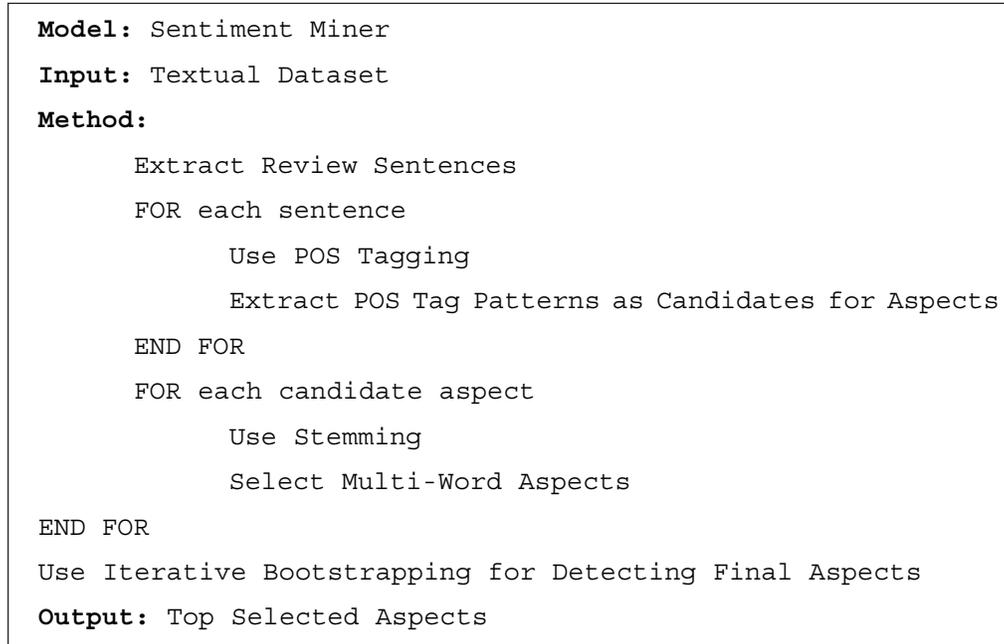


Figure 2. The proposed unsupervised framework: Sentiment Miner

Description	Patterns
Combination of nouns	Unigram to four-gram of NN and NNS
Combination of nouns and adjectives	Bigram to four-gram of JJ, NN and NNS
Combination determiners and adjectives	Bigram of DT and JJ
Combination of nouns and verb gerunds (present participle)	Bigram to trigram of DT, NN, NNS and VBG

Table 1. Heuristic combination POS patterns for candidate generation

4.2 Multi-Word Aspects

In the review sentences, some aspects that people talk about have more than one single word, “digital zoom”, “voice quality” and “battery charging system” are examples. This step is to find useful multi-word aspects from the reviews. A multi-word aspect is represented by

$$a = a_1 a_2 \dots a_n$$

where a_i represents a single-word contained in a , and n is the number of words contained in a . In this paper, we use C -value method to rank the extracted multi-word aspects and select the importance ones.

C -value method takes as input a review set and produces a list of multi-word terms ranking in the descending order of its score [19], [20]. For a multi-word term t , its C -value score can be calculated by:

- If a is not contained by any other words:

$$C\text{-value}(a_i) = \log(|a_i|) * \text{frequency}(a_i) \quad (1)$$

-Else

$$C\text{-value}(a_i) = \log(|a_i|) * \left(\text{frequency}(a_i) - \frac{1}{n} \sum_{i \in n} \text{frequency}(a_i) \right)$$

where $|a_i|$ denotes the number of words contained by a_i , $\text{frequency}(a_i)$ indicates the frequency of occurrence of in the corpus, and n denotes the number of terms in a .

Figure 3 shows the pseudocode for the C-value method given a word for detecting multi-word aspects in Sentiment Miner framework.

```
Function CValueForAWord(String word)
{
    double cValue;
    String s[];
    int n;
    int index;
    int wordFrequency;
    int PTword = 1; // number of extracted candidate terms that
    contain the word
    int fb = 0; // frequency of occurrence of longer candidate term
    b in the corpus
    s = word.split(" ");
    n = s.length;
    index = candidateFeatureList.indexOf(word);
    wordFrequency=candidateFeatureReviewSentenceFrequency.
    get(index);
    for(int i = 0; i < candidateFeatureList.size(); i++)
    {
        if(i == index)
            continue;
        if(candidateFeatureList.get(i).contains(word))
        {
            PTword++;
            fb+= candidateFeatureReviewSentenceFrequency.get(i);
        }
    }
    cValue= (double) n * (double)(wordFrequency - (1.0 /
    (double)PTword) * (double)fb);

    return cValue;
}
```

Figure 3. The pseudocode for the C-value method

4.3 Pointwise Mutual Information (PMI)

PMI is a measure of association used in information theory and statistics [1], [5], [15]. The PMI value of two random variables is a quantity that measures the mutual dependence of the two random variables. In other words, for the field of text mining PMI compares the probability of observing two words together with the probabilities of observing two words independently. So it can be used to estimate whether the two words have a genuine association or just be observed by chance [5]. Researchers have applied this measurement to many natural language processing and data mining problems such as feature selection and text classification.

PMI can be computed by:

$$PMI(a, b) = \log \frac{P(a, b)}{P(a)P(b)} \quad (2)$$

Where $P(a, b)$ is the probability of co-occurrence of word a and word b together, and $P(a)$ and $P(b)$ are the probability of occurrence of word a and word b in the review documents respectively. Usually, word probabilities $P(a)$, $P(b)$ and joint probabilities $P(a, b)$ can be estimated by counting the number of observations of a , b and the co-occurrence of a and b in a corpus normalizing by the size of the corpus.

Intuitively PMI measures if the co-occurrence of a and b is more likely than their independent occurrences. The quality of the PMI algorithm largely depends on the size of training data. If there is no co-occurrence of a and b words in the corpus, the accuracy of PMI becomes an issue.

4.4 Iterative Bootstrapping Algorithm for Detecting Aspects

Iterative bootstrapping algorithm focuses on to learn final list of aspects from a small number of unsupervised seed set information. Bootstrapping can be viewed as an iterative clustering technique which in each iteration, the most interesting and valuable candidate is chosen to adjust the current seed set. This technique continues until satisfying a stopping criterion like a predefined number of outputs. The important part in an iterative bootstrapping algorithm is how to measure the value score of each candidate in each iteration. The proposed iterative bootstrapping algorithm for detecting aspects is shown in Figure 2. In this algorithm we use A-score metric to measure the value score of each candidate in each iteration.

```
Algorithm: Iterative Bootstrapping for Detecting Aspects
Input: Aspects, Candidate Aspects
Method:
    FOR each candidate aspect
        Calculate PMI
        Add the Aspect with Maximum PMI to the Aspects
    END FOR
    Copy Aspects to Final Aspects
Output: Final Aspects
```

Figure 4. The proposed iterative bootstrapping algorithm for detecting aspects

From Figure 4, the task of the proposed iterative bootstrapping algorithm is to enlarge the initial set into a final list of aspects. In each iteration, the current version of the aspect set and the list of candidate aspects are used to find the value score of PMI metric for each candidate, resulting one more aspect for the seed set. Finally, the augmented aspect set is the final aspect list and the output of the algorithm.

5. Experiment Results

In this section we discuss the experimental results for the proposed framework and presented algorithms.

5.1 Data Collection

We employed datasets of customer reviews for five products with feature annotations for our evaluation purpose [2]. This dataset focus on different domain of electronic products: Apex AD2600 Progressive-scan DVD player, Canon G3, Creative Labs Nomad Jukebox Zen Xtra 40 GB, Nikon Coolpix 4300, and Nokia 6610, and have been widely used by researchers for opinion mining. Table 2 shows number of reviews, number of review sentences and the number of manually tagged product aspects for each product in this dataset. Since these five datasets are small for aspect detection in review mining, we crawled many other product reviews from *Amazon.com* and *cnet.com*. The details of each dataset are given in Table III. Newly extracted product reviews are from the same domain as Table 2, but the difference is that they are not from the same specific product but from similar series of the product [5]. For example ‘Canon’ in Table 3 shows reviews for ‘Canon g3’, ‘Canon g6’, ‘Canon g7’ and ‘Canon g10’.

Dataset	Number of reviews	Number of review sentences	Number of manual aspects
Canon G3	45	597	100
Nikon Coolpix 4300	34	346	74
Nokia 6610	41	546	109
Creative Labs Nomad Jukebox Zen Xtra	95	1716	180
Apex AD2600	99	740	110

Table 2. Summary of Customer Review Datasets

Dataset	Number of reviews	Number of review sentences
Canon	484	5875
Nikon	503	4335
Nokia	471	4069
Creative	525	6206
Apex	497	4124

Table 3. Detailed information of the five review datasets

Since the product features in the customer review datasets in Table 2 have already been annotated by human annotators, these annotated product features form a gold standard for the corresponding domain and we use them as reference values for each dataset.

5.2 Evaluation Measures

We use precision, recall, and F-score to measure the effectiveness of an aspect extraction technique [21].

The precision, recall and F-score are calculated based on Table 4 as:

$$Precision = TP / (TP + FP) \tag{3}$$

$$Recall = TP / (TP + FN) \tag{4}$$

$$F - score = 2 * Precision * Recall / (Precision + Recall) \tag{5}$$

	Not Predicted as Aspect	Predicted as Aspect
Wrong Aspects	TN	FP
Correct Aspects	FN	TP

Table 4. Contingency table

5.3 Comparative Study

In our evaluation, after preprocessing and extracting the candidates, we score each multi-word aspect with the C-value method and select those with the score higher than the average, and then we merge single-word and multi-word aspects in a list.

Finding an appropriate number of good initial aspects for bootstrapping algorithm is an important step. In our experiments we used PMI metric to extract automatically the seed set. We have experimented with different numbers of aspects (i.e., 5, 10, 15 and 20) for iterative bootstrapping, and found that the best number of the initial aspects is about 10 to 15. Therefore seeds were automatically chosen for iterative bootstrapping algorithm, and the stopping criterion is defined when about 70 to 120 aspects have been learned. Table 5 shows the experimental results of Sentiment Miner at the main steps described in Section 4, selecting multi-word aspects using C-value and iterative bootstrapping with PMI steps.

Dataset	Using C-value	Bootstrapping with PMI
Precision		
Canon	26.88	63.85
Nikon	27.48	53.57
Nokia	26.31	60.49
Creative	16.56	72.88
Apex	17.80	56.92
Recall		
Canon	74.02	68.83
Nikon	70.58	58.82
Nokia	79.71	71.01
Creative	77.46	60.56
Apex	61.90	58.73

Table 5. Recall and precision at the main steps of the Sentiment Miner

Table 5 gives all the precision and recall results at the main steps of the Sentiment Miner. In this table, column 1 lists each product. Each column gives the precision and recall for each product. Column 2 uses extracted single-word aspects and selected multi-word aspects based on C-value approach for each product. The results indicate that extracted aspects contain a lot of errors. Using this step alone gives poor results in precision. Column 3 shows the corresponding results after employing Iterative bootstrapping algorithm with PMI metric. We can see that the precision is improved significantly by this step but the recall drops.

We evaluate the effectiveness of the proposed framework compared with the benchmarked results by the proposed ARM-based approach by Hu and Liu. The ARM technique exploits a list of positive and negative adjectives to recognize opinion words semantically and subsequently extract product aspects expressed in customer reviews.

Table 6 shows the experimental results of Sentiment Miner in comparison with the ARM technique. The ARM technique employ a minimum support threshold set at 1% in the frequent aspect identification step for finding aspects according to the apriori association rule mining.

Product	ARM		Sentiment Miner	
	Precision	Recall	Precision	Recall
Canon	51.10	63.00	63.85	68.83
Nikon	51.00	67.60	53.57	58.82
Nokia	49.50	57.80	60.49	71.01
Creative	37.00	56.10	72.88	60.56
Apex	51.00	60.00	56.92	58.73

Table 6. Experiment results of comparative study

From Table 6, the maximum precision and recall of the existing ARM technique are 51.10% and 67.60% respectively, whereas the maximum for precision and recall of the Sentiment Miner framework are 72.88% and 71.01% respectively. For all five customer datasets precision of Sentiment Miner is higher than ARM, while the recall for the product datasets of Canon, Nokia and Creative is higher for Sentiment Miner than the ARM. In Nikon and Apex datasets the recall of ARM get higher than the Sentiment Miner. Therefore the effectiveness of Sentiment Miner is better than that of the ARM technique, recording improvements in almost all cases in precision and recall.

It is notable that we observe a more substantial improvement in precision than in recall with our proposed framework and techniques. Observing from Table 6, our framework makes significant improvements over ARM in all the datasets in precision. However, the proposed Sentiment Miner framework achieves an averagely higher recall than the ARM technique. One reason is that for the iterative bootstrapping algorithm we limit number of output aspects between 70 and 120 aspects, therefore the precision for the output will be better than the recall, another reason for low recall is that of our framework only works in detecting explicit aspects from review sentences.

Figure 5 shows the F-score measures of ARM and the proposed Sentiment Miner framework using different product datasets. In the four of datasets, Canon, Nokia, Creative and Apex, the Sentiment Miner achieves the highest F-score, while in the Nikon

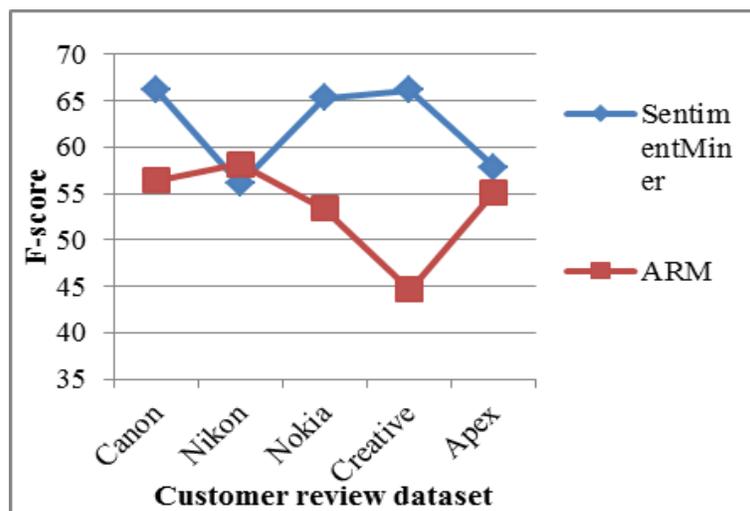


Figure 5. F-scores of ARM and the proposed Sentiment Miner framework

dataset ARM has 2.06% improvement over the Sentiment Miner. This indicates our unsupervised model is effective in extracting correct aspects. We can thus draw the conclusion that the proposed Sentiment Miner is superior to the ARM baseline, and can be used in practical settings. The best F-score of the proposed Sentiment Miner is for Canon dataset and is about 66.24% and the worst case is for Nikon dataset and is 56.07%.

This comparative evaluation suggests that the Sentiment Miner, which involves frequency-based and inter-connection information between the aspects and gives more importance to multi-word aspects, attains better effectiveness for product aspect extraction. The existing ARM technique depends on the frequencies of nouns or noun phrases for the aspect extraction. For Example, our proposed framework is effective in detecting aspects such as “digital zoom” or “battery charging system”, which the ARM technique is failed on extraction of these non-noun phrases. Additionally, we can tune the parameters in the Sentiment Miner to extracts aspects with less or more words, for example aspect “canon power shot g3” can be finding by the framework. Finally, the results show using a completely unsupervised approach for aspect detection in sentiment analysis could achieve promising performances.

6. Conclusion

This paper proposed the task of aspect detection in sentiment analysis in an unsupervised setting. The paper proposed Sentiment Miner to use the PMI algorithm with a new Iterative bootstrapping algorithm to solve the problem. Sentiment Miner detects both single and multi-word aspects from customer reviews. Empirical evaluations using 5 real-life data sets show that the proposed method is superior to the ARM baseline. In our future work, we will focus on further improving the precision. Also we plan to employ clustering methods in conjunction with the model to extract implicit and explicit aspects together to summarize output based on the opinions that have been expressed on them.

References

- [1] Liu, B., Zhang, L. (2012). A survey of opinion mining and sentiment analysis, *Mining Text Data*, Springer, p. 415-463.
- [2] Hu, M., Liu, B. (2004). Mining and summarizing customer reviews, *In: International Conference on Knowledge Discovery and Data Mining, ACM SIGKDD*, p. 168-177.
- [3] Liu, B., Hu, M., Cheng, J. (2005). Opinion observer: analyzing and comparing opinions on the web, *In: International Conference of World Wide Web*, p. 342-351.
- [4] Qiu, G., Liu, B., Bu, J., Chen, C. (2011). Opinion word expansion and target extraction through double propagation, *Computational Linguistics* 37 (1) 9-27.
- [5] Bagheri, A., Saraee, M., de Jong, F. (2013). Care more about customers: unsupervised domain-independent aspect detection for sentiment analysis of customer reviews, *Knowledge-Based Systems*, 52, p. 201-213.
- [6] Brody, S., Elhadad, N. (2010). An unsupervised aspect-sentiment model for online reviews, *In: Annual Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics*, p. 804-812.
- [7] Bagheri, A., Saraee, M., de Jong, F. (2013). An Unsupervised Aspect Detection Model for Sentiment Analysis of Reviews, *In: Natural Language Processing and Information Systems, Springer Berlin Heidelberg*, p. 140-151.
- [8] Titov, I., McDonald, R. (2008). A joint model of text and aspect ratings for sentiment summarization, *In: Annual Meeting on Association for Computational Linguistics and the Human Language Technology Conference*, p. 308–316.
- [9] Yu, J., Zha, Z. J., Wang, M., Chua, T. S. (2011). Aspect ranking: Identifying important product aspects from online consumer reviews, *In: 49th Annual Meeting of the Association for Computational Linguistics*, p. 1496–1505.
- [10] Zhu, J., H., Wang, M., Zhu, B.K., Tsou, M. (2011). Ma, Aspect-based opinion polling from customer reviews, *IEEE Transactions on Affective Computing*, 2 (1) 37-49.
- [11] Popescu A. M., Etzioni, O. (2005). Extracting product features and opinions from reviews, *In: Proceedings of Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics*, p. 339-346.
- [12] Wei, C. P., Chen, Y. M., Yang, C. S., Yang, C. C. (2010). Understanding what concerns consumers: a semantic approach to

product feature extraction from consumer reviews, *Information Systems and E-Business Management*, 8 (2) 149-167.

- [13] Somprasertsri, G., Lalitrojwong, P. (2008). Automatic product feature extraction from online product reviews using maximum entropy with lexical and syntactic features, *In: Proceedings of IEEE International Conference on Information Reuse and Integration*, p. 250-255.
- [14] Thet, T. T., Na, J. C., Khoo, C. S (2010). Aspect-based sentiment analysis of movie reviews on discussion boards, *Journal of Information Science*, 36 (6) 823-848.
- [15] Saraee, M., Bagheri, A. (2013). Feature Selection Methods in Persian Sentiment Analysis, *In: Natural Language Processing and Information Systems*, Springer Berlin Heidelberg, p. 303-308.
- [16] Raut, S. S., Mune, A. R., Student, M. E. (2018). Review Paper on Sentiments Analyzer by Using a Supervised Joint Topic Modeling Approach. *International Journal of Engineering Science*, 15976.
- [17] Schouten, K., Frasincar, F. (2016). Survey on aspect-level sentiment analysis, *IEEE Transactions on Knowledge and Data Engineering* 28 (3) 813-830.
- [18] Toutanova K., Klein, D., Manning, C. D., Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network, *In: Proceedings of HLT-NAACL*, p. 252-259.
- [19] Yoshida, M., Nakagawa, H. (2005). Automatic term extraction based on perplexity of compound words, *Natural Language Processing*, 269-279.
- [20] Nakagawa, H., T. Mori, T. (2003). Automatic term recognition based on statistics of compound nouns and their components, *Terminology*, 9 (2) 201-219.
- [21] Yang, Y. (1999). An evaluation of statistical approaches to text categorization, *Information retrieval*, 1 (1) 69-90.
- [22] Wang, W., Wang, H., Song, Y (2017). Ranking product aspects through sentiment analysis of online reviews, *Journal of Experimental & Theoretical Artificial Intelligence*. 29 (2) 227-246.