ASPECT EXTRACTION AND TWITTER SENTIMENT CLASSIFICATION BY FRAGMENT RULES

Vasilyev V. G. (vvg_2000@mail.ru),
Denisenko A. A. (denisenko_alec@mail.ru),
Solovyev D. A. (dmitry_soloviev@bk.ru)

OOO «LAN-PROJECT», Moscow, Russia

The paper deals with approaches to explicit aspect extraction from user reviews of restaurants and sentiment classification of Twitter messages of telecommunication companies based on fragment rules. This paper presents fragment rule model to sentiment classification and explicit aspect extraction. Rules may be constructed manually by experts and automatically by using machine learning procedures. We propose machine learning algorithm for sentiment classification which uses terms that are made by fragment rules and some rule based techniques to explicit aspect extraction including a method based on filtration rule generation. The article presents the results of experiments on a test set for twitter sentiment classification of telecommunication companies and explicit aspect extraction from user review of restaurant. The paper compares the proposed algorithms with baseline and the best algorithm to track. Training sets, evaluation metrics and experiments are used according to SentiRuEval. As our future work, we can point out such directions as: applying semi-supervised methods for rule generation to reduce the labor cost, using active learning methods, constructing a visualization system for rule generation, which can provide the interaction process with experts.

Key words: fragment rules, sentiment classification, aspect extraction, opinion mining

1. Introduction

Opinion mining and sentiment extraction is an actively developing sub discipline of data mining and computational linguistics. A promising approach to automatic sentiment extraction is based on extraction of specific product features — aspects and on the determination of those polarities. Usually the problem is solved in three stages. At first aspects and those polarities are extracted. Then aspects gears to categories if they are predefined. Otherwise a set of aspects is clustered and representative aspects are selected. The final stage includes category polarity classification based on polarities of individual aspects.

In this paper we present a rule-based approach which exploits fragment rule model to explicit aspect extraction from user reviews and to sentiment classification of twitter messages. The main advantage of the approach is its good interpretability.

On the one hand, there is an opportunity to use expert knowledge in the model by means of constructing rules manually. On the other side, you can build the model automatically or get the interpretable model within a procedure, which includes interaction of an expert and a system.

In paper [7] approaches to sentiment classification of movie reviews are described. These approaches based on counting the number of the proposed positive and negative words and using Naive Bayesian classifier, maximum entropy classification, support vector machine. Using support vector machine raises accuracy to 82%. Another two methods of classification gives accuracy 75–80%. In paper [1] twitter sentiment classification based on support vector machine is described. The words, phrases and part of speech are used as features. The results shown in this paper are the same as results shown in the previous paper and stressed that using part of speech does not increased accuracy.

In paper [2] two approach to sentiment classification movie review. The first approach based on the number of positive and negative terms, intensification terms, and reverses the semantic polarity of a particular term. The second approach uses a machine learning algorithm, support vector machines. Using the first approach gives accuracy about 65–70%. Using the second approach raises accuracy to 85%. Combination the two approaches not increase accuracy.

In paper [3] authors propose approach to sentiment classification with polarity shifting detection. Polarity-shifted and polarity-unshifted sentences are used as features for classification based on support vector machine. This approach allows a few to improve the quality compared to the baseline.

In addition to the vocabulary and the vector approach for sentiment classification a number of papers propose special probabilistic models, for example, tree-based sentiment classification and using relationship between words [6]. Also, a number of papers the authors clearly define the rules of assessment texts. Particularly, in paper [7] different rule for determining the scope inverse word such as "no" are formulated. Thus, in the work on sentiment classification are used as standard methods for text classification, and modified methods, which take into account polarity shifted terms, the syntactic structure of sentences, the relationship between words.

In current paper approach to twitter sentiment classification based on features extracted by using fragment rules. Thus obtained features with proper setting of rules form the space of smaller dimension and have good descriptive power, as was shown in [10].

Aspect-based opinion mining has been widely researched. There are some known approaches to this task [4]: (1) frequency-based approach, (2) rule-based approach, (3) supervised learning techniques, (4) topic modelling techniques.

Frequency-based approach uses the fact that 60–70% of the aspects are explicit nouns [4]. It is argued that people writes reviews in aspect language because they also read other reviews and take the terminology. Rule-based approach uses the assumption that there is some kind of relation between aspects and polarities expressed in a text. A relation can be formalized by using rules. There is also a hybrid approach expressed in using rules for filtration of extracted noun phrases.

The problem may be considered as sequence labelling problem according to some suggested supervised machine learning methods. In particular, Hidden Markov Model and Conditional Random Fields can be used. Topic modelling techniques use the natural assumption that topics of reviews are corresponding aspects.

In this paper, a rule-based approach to aspect extraction is proposed. There are two main rule models: grammar-based and fragment-based. Grammar models include the application of context-free grammars for example Tomita parser [8]. The other model is based on using special fragments from text and represents a number of operations under these fragments. A rule in this case is a declarative description of extracted information. Our model is an example of the last approach.

Due to the fact, that recall of aspect extraction can be achieved by using various dictionaries like thesaurus and domain-specific dictionaries, an important issue is improving precision. In this case, the improvements expressed in using special filtration mechanisms for extracted aspects. Here particularly fragment rules can be used. The purpose of participation in the track was testing fragment rule-based approaches to aspect extraction and tweet classification. In addition, we attempted to use methods for automatic fragment rule generation.

The remainder of the article is as follows. In section 2 a formal description of the fragment rule language and a description of proposed approaches is given. In section 3 obtained results are analyzed; a comparison with Baseline results and the best track results is given. Section 4 presents conclusion and future work.

2. Methods

2.1. Fragment rules model

In this work for describing text features and classification rules we used a mathematical model based on defining operations on sets of text fragments [9].

Let we have the text $D = (d_1, ..., d_n)$, where the $d_i \in T$ — single element of the text, $T = \{t_1, ..., t_m\}$ — the set of all elements, n — the length of the text, m — number of different elements of the text.

Definition 1

The set $\mathbb{F} = \{ (p, q) \mid 1 \leq p \leq q \leq n \}$ will be called the set of all parts of the text length n. Fragments of the text will be called the single elements of the set $f = (f_l, f_r) \in \mathbb{F}$, that specify left f_l and right f_r border fragment (number of the first and last elements in fragment).

Definition 2

Let $f = (f_l, f_r) \in \mathbb{F}$ and $g = (g_l, g_r) \in \mathbb{F}$, then $|f| = f_r - f_l + 1$ — length of the fragment;
$g \sqsupset f$, if $g_l \leq f_l \leq f_r \leq g_r$ and $f \neq g$ — inclusion relation;
$g \ll f$, if $g_l < f_l$ or $g_l = f_l \;\&\; f_r < g_r$ — order relation.

Definition 3

The set of fragments F will be called reduced, if there is no such $f \in F$, that $g \sqsupset f$. $R(F)$ denote reduced set of fragments based on the set F, R — reduce operation.

Definition 4

The distance between the fragments $f=(f_l,f_r)\in\mathbb{F}$ and $g=(g_l,g_r)\in\mathbb{F}$ is determined as follows:

$$d(f,g)=\begin{cases} g_l-f_r, f<g,\\ f_l-g_r, g<f,\\ g_l-f_r, g=f.\end{cases}$$

Definition 5

The result of the a rule Q for the text D is the set $F_Q\subset\mathbb{F}$, containing all of the fragment relevant this rule. If $F_Q\neq\emptyset$, then call the text D relevant rule Q.

Definition 6

Basic rules is a rule $Q=t, t\in T$ whose result is $F_Q=\{f_1,\dots,f_l\}$ — reduced set of fragments, the elements that stand out in a single operation. Complex rule is a rule Q, which is obtained by performing operations on other rules $Q_1,\dots,Q_k$.

Let us now determine the possible operations to build complex rules of Q from the basic rules $Q_1,\dots,Q_k$.

Definition 7

$Q=Q_1\nabla Q_2$ — binary operation OR, $F_Q\equiv R\left(F_{Q_1}\nabla F_{Q_2}\right)$,
$F_{Q_1}\nabla F_{Q_2}=\left\{f\in\mathbb{F}\,|\,\exists f_1\in F_{Q_1}, f\sqsupset f_1 or\ \exists f_2\in F_{Q_2}, f\sqsupset f_2\right\}$.

For example, the rule *good best quality* extract fragments relevant the appearance of these words in the text.

Definition 8

$Q=Q_1\Delta_{n_1}Q_2$ — binary operation AND with limit on distance between fragments, $F_Q\equiv R\left(F_{Q_1}\Delta_{n_1}F_{Q_2}\right)$, $F_{Q_1}\Delta_{n_1}F_{Q_2}=\left\{f\in\mathbb{F}\,|\,\exists f_1\in F_{Q_1} and\ \exists f_2\in F_{Q_2}, that\ f\sqsupset f_1,\ f\sqsupset f_2 and\ d(f_1,f_2)\leq n_1\right\}$.

For example, the rule *beeline &4w LTE* extract fragments, in which distance between "*beeline*" and "*LTE*" less than 4 words. This operation can be used without any limits on the distance between the words.

Definition 9

$Q=Q_1\square_{n_1,n_2}Q_2$ — binary operation of sequence with limit on distance between fragments, $F_Q\equiv R\left(F_{Q_1}\square_{n_1,n_2}F_{Q_2}\right)$, $F_{Q_1}\square_{n_1,n_2}F_{Q_2}=\{f\in\mathbb{F}\,|\,\exists f_1\in F_{Q_1} and\ \exists f_2\in F_{Q_2},\ that\ f_1<f_2,\ d(f_1,f_2)>0,\ f\sqsupset f_1, f\sqsupset f_2\ and\ n_1\leq d(f_1,f_2)\leq n_2\}$.

For example, the rule *@Company: 3w (sale discount)* extract fragments, which after the name of the company at a distance of 3 words are words of "sale" or "discount". This operation can be used without any limits on the distance between the words.

Definition 10

$Q = \bowtie (Q_1, ..., Q_k)$ — multiple operation sequences of neighbouring elements (select of neighbouring fragments), $F_Q \equiv R\left(\bowtie \left(F_{Q_1}, \ldots, F_{Q_k}\right)\right), \bowtie \left(F_{Q_1}, \ldots, F_{Q_k}\right) =$ $= \{f \in \mathbb{F} | \exists f_i \in F_{Q_i}, i \in \overline{1,k}\colon f_i < f_{i+1}, d(f_i, f_{i+1}) = 1, i \in \overline{1, k-1} and\ f \sqsupset f_i,$ $i \in \overline{1,k}\}$.

For example, the rule “*(boss head director chief) (mts beeline megafon)*” extract phrases corresponding to different telecom executives.

Definition 11

$Q = Q_1 \between Q_2$ — binary operation finding the intersection of fragments, $F_Q \equiv \left\{f \in \mathbb{F} | f \in F_{Q_1} \wedge f \in F_{Q_2}\right\}$.

For example, the rule *[Chapter $SentBegin] extract words* “*Chapter*”, that are written in the beginning of the sentence.

Definition 12

$Q = Q_1 \lhd_{n_1, n_2}$ — unary operator imposes limitations on length of the fragment, $F_Q \equiv \left\{f \in F_{Q_1} | n_1 \leq |f| \leq n_2\right\}$.

For example, the rule *(beeline & mts) #IN #INTERVAL(2w/3w)* extract fragments containing specific words in length from 2 to 3 words.

To be able to construct rules include negation and conditional statements (when the presence of the expression is checked, but it is not included in the final fragment) are special variants of binary rules $\nabla, \Delta, \square, \bowtie, \between, \lhd,\ \Delta_{n_1}, \square_{n_1 n_2}$, in which one of the operands is considered negative or conditional. For example, $\square^{\nrightarrow}_{n_1,n_2}$ is operator finding the sequence in which the second operand is taken from the negation; $\square^{\nleftarrow}_{n_1,n_2}$ is operator finding the sequence in which the first operand is taken from the negation; $\square^{\leftarrow}_{n_1,n_2}$ — is operator finding the sequence in which the first operand is conditional. The rule $\square^{\nrightarrow}_{n_1,n_2}$ defined as $Q = Q_1 \square^{\nrightarrow}_{n_1,n_2} Q_2\ \ F_Q \equiv \left\{f \in F_{Q_1} | \ \exists! f_2 \in F_{Q_2}\colon f < f_2, 0 < n_1 \leq d(f, f_2) \leq n_2\right\}$.

For example, the rule *no ^:3 (good best quality)* extract the word “*good*”, “*best*” and “*quality*” before which there is no word “no” at distance of three words.

#define command sets the named expression. In the pre-treatment rules text expression is substituted into the rule text. These expressions are used to avoid repeating elements in complex rules. *#set* command s used to set the saved variables. Unlike #define command at the first reference to the variable is made save search results and on subsequent calls text processing is not performed. To use named expressions or saved variables in the rule is necessary to use operators @ and @@.

For example, *#define Good (good best quality)* sets the named expression *Good,* which should be handled @*Good.*

2.2. Sentiment classification

For sentiment classification we used a hybrid approach which is based on combining rule-based feature extraction and classifier training by machine learning methods. Classifier induction includes training set pre-processing, feature extraction by using predefined set of fragment rules, training classifier by using selected machine learning methods.

Texts in the training set are pre-processed by using the following procedures:

1. Graphematical analysis (tokenization, sentence boundary detection, phonetic coding, word descriptors extraction).
2. Linguistic analysis (lemmatization, part of speech tagging, word sense disambiguation, collocation extraction, syntactic features extraction).
3. Low level indexes construction (inverted index of source word forms, inverted index of lemma word forms, inverted index of word descriptors).

The general scheme of the learning algorithm has the following form.

1. Building vector representation of texts by using the set of fragment rules.
2. Dimension reduction and feature weights calculation.
3. Training and evaluation of the classifier on the training set.

At the first step the predefined set of 100 special fragment rules are used for features extraction.

Example of fragment rule:
@@COND^:5((@@NEG^:5\s(@@INTENS^:5\s($Adj $Verb $Noun $Adv))) &5\s? @@OBJECT),

where @@COND—condition words (“if”), @@NEG—negative words, @@INTENS—intensive words (“very”, “far”, ”purely”), @@OBJECT—object (“mts”, “megafon”, “beeline”).

At the second step we used common methods for dimension reduction and feature weights calculation.

At the third step two classifiers are trained, one classifier for the positive class and one for the negative class. For classifier training we used our robust realization of the following standard machine learning methods:

1. Bayesian classifier based on multivariate Gaussian distribution (gmm),
2. K-nearest neighbours classifier (knn),
3. Von Mises-Fisher classifier (vmfs),
4. Roccio classifier (roccio),
5. Support vector machines classifier (svm).

Trained positive and negative classifiers are used for building the final decision rule of the following form:

$$d'(u) = \begin{cases} 1, d_{pos}(u) > d_{neg}(u) \mid d_{pos}(u) = d_{neg}(u) = 1, w_{pos}(u) > w_{neg}(u) \\ -1, d_{pos}(u) < d_{neg}(u) \mid d_{pos}(u) = d_{neg}(u) = 1, w_{pos}(u) < w_{neg}(u) \\ 0, d_{pos}(u) = d_{neg}(u) = 0 \end{cases}$$

where $d'(u) \in \{-1, 0, 1\}$ is the final decision rule, $d_{pos}(u) \in \{0, 1\}$ and $d_{neg}(u) \in \{0, 1\}$ is the decision rules for positive and negative class, $w_{pos}(u) \in [0, 1]$ and $w_{neg}(u) \in [0, 1]$

and degree of compliance positive or negative class (for probabilistic classifiers it is the probability assignment to the corresponding class, for svm it is the distance to corresponding hyperplane etc.), u — the set of features in the text.

2.3. Rule-based explicit aspect extraction

There are two types of aspects defined in aspect-based opinion mining: explicit and implicit. Explicit aspects are concepts that explicitly mentioned in a sentence. Implicit aspects are expressed indirectly. This section proposes a number of approaches to explicit aspect extraction based on fragment rules. Preliminary let $A = \{a_1, ..., a_n\}$ be a set off unique aspects extracted by experts and represented in the training set. Training set has been provide by SentiRuEval organizers [5].

Multiple operation OR

Basically for the purpose of explicit aspect extraction this kind of fragment rule can be used:

$$Q = Q_{\nabla}(a_1, a_2, ..., a_n),\ a_i \in A.$$

Here Q_{∇} — is a rule, where operation OR acts as a connector between unique aspects. In fact, an appropriate set of fragments is extracted for each aspect. The result of the operation is a reduced united set of fragments.

Multiple operation OR with maximizing reduction

In the concerned case, the following situation may arise. Instead of a whole aspect, structural parts can be extracted. For example, there are three extracted aspects HOT, DISH, HOT DISH. A standard reduction method will delete the biggest fragment HOT DISH, and we'll have two aspects instead of one. In this regard, it was decided to modify the reduction method and to exclude fragments which are included in other fragments. Also it should be noted that neighbouring fragments may be one aspect. Therefore overlapping fragments and neighbouring fragments should be combined. As a result, fragments of the maximum length are extracted.

Rule-based filtration

Also it seems appropriate to use rule-based filtration for aspect extraction. The extraction algorithm constructed as follows. At first using aspects selected by experts fragments from an aspect to the nearest adjective are extracted. Then, the most common rules based on the extracted fragments (templates) are formed. Here in the feature space is defined previously. The generated rules are applied to filter the set of extracted candidate-aspects by counting support and removal of candidates with support below a threshold. As already mentioned, recall may be achieved by using appropriate dictionaries. In this case, the filtration process is necessary to improve precision. Definition of the context of some aspects allows to separate situations where the term is not an aspect.

Let (a_i) be a rule, a result is a set of fragments from the aspect a_i to the nearest adjective. The aspect extraction algorithm for each aspect selected by experts generates a set of aspect contexts $Q(a_i)$ by applying rule $Q(a_i)$ to the training set L.

Then the rule generation algorithm builds templates of these contexts. In each review candidate-aspects are extracted and filtered by using these templates. Finally, we have a set of extracted explicit aspects.

Algorithm2. Explicit aspect extraction with filtration

Input. A_L — set of aspects selected by experts
I — hierarchy of features,
L — train set,
R — test set

Output. A_T — extracted explicit aspects.

Step 1. For all $a_i \in A_L$
$GenerateRules(I, Q_L(a_i))$

Step 2. For all $r \in R$
For all $a_i \in A$
$A_T \leftarrow A_T \cup FilterAspects(Q_r(a_i))$

There are a number of classical algorithms for searching frequent item sets which used for generating rules such as *Apriori*, *FP-growth*, *Eclat*. One important difference between these algorithms is a method of data representation. Basically there are two approaches—horizontal and vertical representation. In the vertical representation it's necessary to have lists of fragments that match elements of a rule. In the horizontal representation each fragment corresponds to a set of rule elements. Vertical representation is more practical in case of the fragment model. In this context, it is possible to apply one of the known algorithms — Eclat [11]. Especially because support of rules is determined by the intersection of sets of fragments.

Rules of the form $Q_1 \square_{1,1} Q_2 \square_{1,1} \ldots \square_{1,1} Q_n$ are used for filtration. Searching of rules is based on a feature hierarchy. As elements of the hierarchy you may have parts of speech descriptors, single words, etc. Sequentially from the descriptor $Any (any word) a rule is expanding and specifying. A selection criterion is a degree of specificity of rules and a minimal support threshold. The specificity of the rules increases depending on a number of elements and their place in the hierarchy. The more elements and the lower the place of elements in the hierarchy then specificity is higher. In this case, the rules are eliminated with support below a threshold. As a result, every aspect is associated with set of rules. In such a way, filtration is done when there are only those candidate-aspects which match at least one rule.

3. Evaluation

3.1. Twitter sentiment classification

Used for teaching training set consisting of 3,846 tweets of telecommunications companies. Each company which was mentioned on Twitter rated on a scale {–1, 0, 1}.

Test set consists of 5,322 tweets about telecommunications companies. The objective of the testing was to include every mention of the company to one of three classes: positive, negative or neutral. Indicators macro *F*-measure and micro *F*-measure used to assess the quality. Test results are shown in Table 1. The table shows the best method, Baseline and 5 runs:

- 9_1 Bayesian classifier based on a mixture of multivariate normal distributions (*gmm*),
- 9_2 classifier k-nearest neighbours (*knn*),
- 9_3 Bayesian classifier based on the distribution of von Mises-Fisher (*vmfs*),
- 9_4 centroid classifier Roccio (*roccio*),
- 9_5 classifier based on support vector machines (svm).

Baseline refers all tweets to the most frequent class, in this case a negative. Used for teaching training set consisting of 3,846 tweets of telecommunications companies. Each company which was mentioned on Twitter rated on a scale {–1, 0, 1}.

Indicators macro *F*-measure and micro *F*-measure used to assess the quality [5].

Table 1. Evaluation of the quality of sentiment classification tweets

Algorithm	Macro *F*-measure	Micro *F*-measure
9_1 (gmm)	0,3158	0,3331
9_2 (knn)	0,2328	0,2626
9_3 (vmfs)	0,3305	0,3371
9_4 (roccio)	0,3310	0,3501
9_5 (svm)	0,3527	0,3765
Baseline	0,1823	0,3370
2_B	0,4829	0,5362

Evaluating the quality of classification are at Baseline micro *F*-measure and substantially higher macro *F*-measure. This can be explained feature Baseline and calculation rule micro and macro *F*-measure. Macro *F*-measure — is the average amount of standard *F*-measure that calculated separately for the three classes. Baseline algorithm has zero *F*-measure for two classes (positive and neutral), but *F*-measure negative class has a value of about 55%. By averaging the three classes *F*-measure is found to be 18%.Our algorithm solves these problems. The algorithm based on support vector machines shown best quality. The algorithm based on k-nearest neighbours showed the worst result. As we can see our result are comparable with result of other participants.

3.2. Explicit aspect extraction

Performance evaluation was made against the training set (gold standard), provided by organizers. The set consists of 202 annotated reviews in Russian. We used standard measures: precision, recall and F-measure. In official results the method based on multiple operation OR with maximizing reduction has identifier — 11.1.

Table 2. Evaluation results for explicit aspect extraction

Method	Strong demands			Weak demands		
	P	R	F1	P	R	F1
OR	49%	71%	58%	59%	72%	65%
Multiple operation OR with maximizing reduction [11.1]	51%	73%	60%	61%	74%	66%
Rule-basedfiltration	60%	64%	62%	66%	69%	67%
Baseline	55%	69%	61%	65%	70%	67%
[2.1] The best result/strong	72%	57%	63%	81%	62%	69%
[4.1] The best result/weak	55%	69%	61%	69%	79%	73%

In general, participants in the official track had comparable results. It turns out that the approach based on transferring aspects from the train set to the test set with normalization shows the same results as approaches used sophisticated models for training.

The results show that the modification of multiple OR operation generally contributes to the performance. It can be argued that maximizing reduction showed an advantage compared to minimizing reduction when there are only those fragments that contain no other. This reduction is applied in solving text classification tasks and offers advantages in terms of speed of execution of classification rules. In the future, different types of reduction can take the form of individual operations instead of using in default.

Application of rules in filtration also has a positive effect on the result, but there are a number of issues that require further study. Along with increasing precision recall decreases. To solve this problem it is advisable to consider other criteria of rule selection to find suitable experimental values of boundary parameters for rule specificity and support of candidate-aspects to achieve a minimum reduction of recall.

4. Conclusions and Future work

The paper deals with approaches to explicit aspect extraction and sentiment classification. The algorithm based on support vector machines shown best quality. The algorithm based on k-nearest neighbours showed the worst result. The results are at the level of the average results presented in sentiment analysis track. The algorithm based on SVM using as features normalized lemma and syntactic links shown the best results on the track. In the efforts to extract the aspects we can say that the simplest approach shows comparable with the rest of the results. The use of filtering rules to improve the accuracy while reducing completeness. In this regard, it is necessary to separately evaluate the effect of boundary parameters on the result.

As our future work, we can point out such directions as: applying semi-supervised methods for rule generation to reduce the labor cost, using active learning methods, constructing a visualization system for rule generation, which can provide the interaction process with experts. Also expanding of the fragment rule model can give new expressive possibilities.

References

1. *Go A., Huang L., Bhayani R.* (2009), Twitter sentiment classification using distant supervision, CS224N Project Report, Stanford.
2. *Kennedy A., Inkpen D.* (2006), Sentiment Classification of Movie Reviews Using Contextual Valence Shifters, Computational Intelligence, vol. 22(2), pp. 110–125.
3. *Li S., Lee S. Y. M., Chen Y., Huang C.-R., Zhou G.* (2010), Sentiment Classification and Polarity Shifting, Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), Beijing, China, pp. 635–643.
4. *Liu B.* (2012), Sentiment Analysis and Opinion Mining, Morgan and Claypool Publishers.
5. *Loukachevitch N., Blinov P., Kotel'nikov P., Rubtsova Ju., Ivanov V., Tutubalina E.* (2015), SentiRuEval: Testing Object-Oriented Sentiment Analysis Systems in Russian.
6. *Nakagawa T., Inui K., Kurohashi S.* (2010), Dependency tree-based sentiment classification using CRFs with hidden variables, The 2010 Annual Conference of the North American Chapter of the ACL, Los Angeles, USA, pp. 786–794.
7. *Pang B., Lee L., Vaithyanathan S.* (2002), Thumbs up? Sentiment Classification using Machine Learning Techniques, Proceedings of EMNLP, Philadephia, Pennsylvania, USA, pp. 79–86.
8. *Tomita parser: https://tech.yandex.ru/tomita/*
9. *Vasilyev V. G.* (2011), Fragment extraction and text classification by logical rules [Klassifikatsija i vydelenie fragmentov v tekstah na osnove logicheskih pravil] Digital libraries: Advanced Methods and Technologies, Digital Collections RCDL'2011, Voronezh, pp. 133–139.
10. *Vasilyev V. G., Davidov S. U.* (2013), Sentiment classification by combined approach. Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference «Dialog 2013", available at: www.dialog-21.ru/digests/dialog2013/materials/pdf/VasilyevVG.pdf
11. *Zaki M. J., Parthasarathy S., Ogihara M., Li W.* (1997), New Algorithms for Fast Discovery of Association Rules, Proceedings of the Third International Conference on Knowledge Discovery in Databases and Data Mining, Newport Beach, California, USA, pp. 283–286.