COMBINED APPROACH FOR SENTIMENT ANALYSIS IN SLOVAK USING A DICTIONARY ANNOTATED BY PARTICLE SWARM OPTIMIZATION

Martin MIKULA*, Kristína MACHOVÁ**
Department of Cybernetics and Artificial Intelligence, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Letná 9, 042 00 Košice, tel. +421 (55) 602 2936, E-mail: *martin.mikula@tuke.sk, **kristina.machova@tuke.sk

ABSTRACT

Sentiment analysis in the minor languages, such as Slovak, using dictionary approach is a difficult task. It requires a lot of human effort and it is time-consuming to prepare a reliable source of information, especially good dictionary. We propose an approach which uses a biologically inspired algorithm to find optimal polarity values for sentimental words. It applies a swarm intelligence algorithms, standard Particle Swarm Optimization (PSO) and Bare-bones Particle Swarm Optimization (BBPSO), to replace a human annotator at the moment of dictionary creation. We created two dictionaries, which were annotated by the human annotator, PSO and BBPSO. These dictionaries were compared with the result that the versions annotated by PSO and BBPSO outperformed a human annotator. Then a combined approach was used to classify reviews that do not contain words from the dictionary. These reviews decrease the classification performance significantly. The combined approach implements machine learning method to build a model based on the reviews classified by the dictionary approach. The combined approach finally reduced a number of unclassified reviews from 18% and 40.2% to 0.3% and increased the macro-F1 measure from 0.694 and 0.495 to 0.865 and 0.841.

Keywords: *sentiment analysis, dictionary approach, automatic dictionary annotation, particle swarm optimization, bare-bones particle swarm optimization*

1. INTRODUCTION

The social web produces a huge amount of data every day which are very difficult to process manually. Several approaches, including machine learning approach, dictionary-based approaches and deep learning approaches were proposed to process the data automatically. The approaches which are used for sentiment analysis, aim to distinguish between positive or negative (sometimes also neutral [5]) opinions, emotions or wishes towards subjects such as products, movies or people. Machine learning approaches based on well-known machine learning algorithms such as Naïve Bayes classifier, Support Vector Machines, Maximum Entropy or k-Nearest Neighbors [7, 17, 22] are used to assign a positive or a negative polarity to the reviews. They require a labeled training dataset to learn models, that can be applied on a testing dataset. Deep learning methods use Neural networks to discover new features and new information from the current data [18, 23]. Lexicon based approaches usually use lexicons, which contain polarity words. The strength of polarity which is assigned to each word from dictionary, indicates how strong is the word correlated with positive or negative polarity.

All these methods require any source of external knowledge. Machine learning algorithms require an annotated dataset to train the sentiment classifier and to build the model. Deep learning methods require an annotated dataset too. On the other hand, dictionary-based approaches require an annotated dictionary which contains sentiment words with assigned polarity. These data provide all necessary information for sentiment classification. But this information is very unbalanced across different languages. There are languages such as English, in which a lot of previous work has been done and they provide many resources. However, the annotated sources of information in minor languages are rare and it is not a simple task to create them manually. In this paper, we focus on adapting sentiment analysis from English to Slovak. The Slovak language belongs to the group of Slavic languages (with Czech language and Polish) and it has a rich morphology. A combined approach integrates dictionary approach and machine learning method to create more flexible approach.

This paper focuses on adapting sentiment analysis from English to Slovak using automatic dictionary annotation and integration of dictionary-based approach and machine learning method. Particle Swarm Optimization (PSO) [12] is used to find optimal values of the polarity for words in the dictionary. It is a challenging task to adapt a dictionary from a major to a minor language (a language which is not commonly used) such as the Slovak language. It requires a lot of human efforts and it is also time-consuming to translate and label all words in a new dictionary. To assign correct polarity values, more annotators are needed, because the polarity values are often biased by individual preferences of annotator. The automatic translation can not be used directly. Translated words can have more than one sense and they might have a different polarity than the original word. The translated dictionary has to be annotated again. We decided to replace a human annotator by PSO in the annotation stage.

In some cases, a new dictionary might not cover all sentiment words in the target language. Hence some reviews in the target language will not be classified. To solve this problem, we implemented a combined approach which consists of the combination of dictionary approach and machine learning. It uses the dictionary to classify all reviews in the dataset. Then dataset is split into two subsets, the subset classified by dictionary approach and unclassified subset. Classified reviews are used as a training dataset for machine learning method. Then the trained model is applied to unclassified reviews. We implemented Naïve Bayes classifier to build a model and classify unlabeled reviews. It is a simple classifier based on probability theorem with good results for sentiment analysis.

This paper is organized as follow. Section 2 briefly describe dictionary-based approaches and summarizes the related work in the field of sentiment analysis using different types of methods. Section 3 introduces PSO and its modification, which were used in our approach and section 4 details our combined approach. Section 5 analyzes achieved results and section 6 summarizes the contributions of the paper.

2. RELATED WORK

2.1. Dictionary-based approach

Dictionary-based approaches usually apply sentiments dictionaries to classify reviews into positive or negative class respectively. These dictionaries can be generated in three ways, manually, automatically and semi-automatically. Manually generated dictionaries are more accurate and usually involve only single words. They are often translated from another language or collected from the corpus. The value of polarity is copied from the original dictionary or assigned manually. The advantage is that all words in these dictionaries are related to the sentiment. However, this approach requires a lot of human effort and it is time-consuming. Manually created dictionaries separate words into positive and negative groups [14] or provide also additional lists of words such as shifters (words that can change polarity) [20], [10]. Warriner lexicon [25] or Mikula lexicon [13] provide a value of polarity to each word. Automatically generated dictionaries require less human effort. They assign values of polarity based on relations between words in existing dictionaries e.g. WordNet[1]. SentiWordNet [1] contains automatically annotated WordNet synsets according to their degrees of positivity, negativity, and neutrality. In the WordNet-Affect [19], an emotional values were added to each WordNet synset. SenticNet [2] includes commonsense knowledge, which provides background information about words. The main weakness of automatically dictionaries is that they might contain words without polarity or incorrectly assigned polarity. For this reason, the semi-automatic generation of dictionaries was introduced. It creates dictionaries automatically and checks them manually.

2.2. Evolutionary computation

There are several works which used evolutionary computation in text classification. Genetic programming was applied to find new term-weighting schemes in work [6]. The schemes were used to improve classification performance. The standard term-weighting schemes were combined with new term-weighting schemes which are more discriminative and they were created by the genetic algorithm. In work [8], Particle Swarm Optimization was applied to find the most useful features which were added as an input for the framework based on Conditional Random Field. PSO was also used to select features and combine them with Support Vector Machines to classify reviews in [3]. In our paper, PSO learns a group of numbers, which represents the values of polarity for specific words in the dictionary.

3. BASIC IDEA OF PSO

3.1. Standard PSO

Particle Swarm Optimization is an optimization algorithm which is inspired by birds flock. PSO faster converges to the final solution than genetic algorithm and each particle stores knowledge about its best solution. The possible solutions are called particles. Particles are parts of the population called swarm. Each particle keeps its best position (evaluated by the fitness function) called *pbest*. The position of the best particle chosen from the whole swarm is called *gbest*. The standard PSO consists of two steps: change velocity and update position. In the first step, each particle changes its velocity towards its *pbest* and *gbest* [9]. In the second step, the particle updates its position. A new position is calculated based on previous position and a new velocity. Each particle is represented as a vector in a D-dimensional space. The i-th particle can be represented as $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$. The velocity of the i-th particle is represented as $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$ and the best previous position of the particle is represented as $P_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$. The best particle in the swarm is represented by the g particle w is an inertia weight which balances the tends between exploration and exploitation abilities of the particles. The velocity and position are updated using the following equations 1 and 2.

$$v_{id}^{t+1} = w v_{id}^{t} + c_1 r_1^t (p_{id}^t - x_{id}^t) + c_2 r_2^t (p_{gd}^t - x_{id}^t) \quad (1)$$

where:
v_{id}^{t+1} ... is a new velocity of the i-th particle in d-th dimension in $t+1$-th iteration
w ... is an inertia weight
v_{id}^{t} ... is a velocity of the i-th particle in d-th dimension in t-th iteration
c_1 ... is a self-confidence factor
r_1 ... is a uniformly distributed random value in [0,1]
p_{id}^t ... is a *pbest* (personal best) position of the i-th particle in d-th dimension in t-th iteration
x_{id}^t ... is a current position of the i-th particle in d-th dimension in t-th iteration
c_2 ... is a swarm confidence factor
r_2 ... is a uniformly distributed random value in [0,1]
p_{gd}^t ... is a position of the *gbest* particle in d-th dimension in t-th iteration

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \quad (2)$$

where:
x_{id}^{t+1} ... is a new position of the i-th particle in d-th dimension in $t+1$-th iteration
x_{id}^{t} ... is a current position of the i-th particle in d-th dimension in t-th iteration
v_{id}^{t+1} ... is a new velocity of the i-th particle in d-th dimension in $t+1$-th iteration

[1] http://wordnet.princeton.edu/

$d = 1,2,\ldots,D$, and in our system D represents the number of dimensions which corresponds to the number of words in the dictionary, $i = 1,2,\ldots,N$, and N is the number of particles in the swarm, $t = 1,2,\ldots$, denotes the iteration number. Two numbers r_1, r_2 are uniformly distributed random values in [0,1] which avoid the falling down in local optima. c_1 and c_2 respectively are important parameters, known as the self-confidence factor and the swarm confidence factor respectively. They define a type of trajectory the particle travels, so they control the searching behavior of the particle [4]. The stopping criteria of the algorithm often depends on the type of problem. In practice, PSO runs until a fixed number of iterations is done or an error bound is reached.

3.2. Bare-bones PSO (BBPSO)

The standard PSO uses *pbest* and *gbest* to update the position of the particle. The impact of these values was studied in [11]. In this work, *pbest* and *gbest* were set as constants and the trajectories of the particles were investigated. They were plotted and the obtained histogram had a shape of the tidy bell curve with a center between *pbest* and *gbest*. From the results, it was suggested that trajectory can be determined by the difference between *pbest* and *gbest*. So these positions can determine the particle's movement. Based on the results a new PSO method called Bare-bones PSO (BBPSO) was derived. This model of PSO is based on Gaussian distribution $\mathcal{N}(\mu,\sigma)$ with the mean μ and standard deviation σ as shown in Eq. 3.

$$x_{id}^{t+1} = \begin{cases} \mathcal{N}(\mu,\sigma), & rand() < 0.5 \\ p_{id}^{t}, & \text{otherwise} \end{cases} \tag{3}$$

where:
x_{id}^{t+1} ... is a new position of the *i*-th particle in *d*-th dimension in *t+1*-th iteration
p_{id}^{t} ... is a *pbest* (personal best) position of the *i*-th particle in *d*-th dimension in *t*-th iteration

μ is the center of *pbest* and *gbest*, and σ is the absolute difference between *pbest* and *gbest*. The $rand()$ function is used to speed up convergence by retaining the previous best position *pbest*.

4. PROPOSED METHOD

We created two dictionaries to analyze sentiment using dictionaries. The first dictionary (big dictionary) was translated from English. It was manually extended and contains domain depended words (the meaning of the word depends on the domain). For this reason, we decided to create a new dictionary (small dictionary). It is expected that this dictionary is domain independent because it was extracted from six English dictionaries and only domain independent words are included in all dictionaries. Dictionaries were analyzed and only overlapping words from all of them were picked up. The dictionary size is smaller than the size of the big dictionary which is also important. The number of particles in our PSO implementation depends on the size of the dictionary and it influences the time needed to find an optimal solution.

For each dictionary, three versions were generated. The first version was annotated manually by a human annotator, the second version was annotated by PSO, and the third using BBSPO. Then, all versions were used for sentiment analysis in the Slovak language.

To adapt to the target language, the combined approach was used. It combines dictionary-based approach and machine learning method. In our work, we decided to apply Naïve Bayes classifier, which is a simple probabilistic classifier with good results in sentiment analysis.

4.1. Big dictionary

The big dictionary was derived from an English dictionary. The original dictionary [10] consists of 6789 words, including 13 negations. We translated only positive and negative words. Synonyms and antonyms from the Slovak thesaurus were found for each original word. The thesaurus was also used to determine intensifiers and negations. The big dictionary consists of 598 positive words, 772 negative words, 41 intensifiers and 19 negations. The first version of this dictionary was annotated manually. The range of polarity from -3 (the most negative word) to +3 (the most positive word) was chosen as a polarity values.

For each word in the dictionary, the English form was searched by a double translation. "Double translation" means that each word was translated into English and then, it was translated back to Slovak. In case, that the word had the same meaning before and after translation, the English form of the word was used.

4.2. Small dictionary

The second dictionary (small dictionary) was derived from the six different English dictionaries used in works [2, 10, 15, 16, 21, 24]. The comparison of these dictionaries can be seen in Table 1. The English dictionaries were analyzed and only overlapping words were picked up to a new dictionary. To translate these words to Slovak, the English translations from the big dictionary were used. The overlapping words were found and their Slovak forms were added to the dictionary. A new lexicon contains 220 words, including 85 positive words and 135 negative words. Intensifiers and negation were not added because they were not included in all original dictionaries. The first version of the dictionary was annotated manually with a range of polarity from -3 to 3.

Table 1 The comparison of dictionaries used for creation of the small dictionary.

Dictionary	Pos.	Neg.	Intens.	Opos.
Hu and Liu dic. [10]	4783	2006	-	13
Taboada lexicon [21]	2535	4039	219	-
SenticNet 4.0 [2]	27405	22595	-	-
AFINN [16]	878	1598	-	-
Sentiment140 [15]	38312	24156	-	-
SentiStrength [24]	399	524	28	17

4.3. Annotation by PSO

Another two versions of the dictionaries were annotated by PSO and BBPSO, respectively. PSO is an efficient, robust and simple optimization algorithm which has been successfully applied to optimizing various functions.

During evolutionary process, each particle represents one sub-version of dictionary and can be encoded as a vector $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ where $x_{ij} \in \{-3, 3\}$, $i = 1, 2, \ldots, N$ where N is the number of particles and $j = 1, 2, \ldots, D$ where D denotes the number of words in a dictionary. The particle size depends on the size of the dictionary. From the big dictionary, only positive and negative words were used, thus the particle size is 1370 polarity values. The particle that represents the small dictionary has size of 220 polarity values. The designed approach is shown in Figure 1.

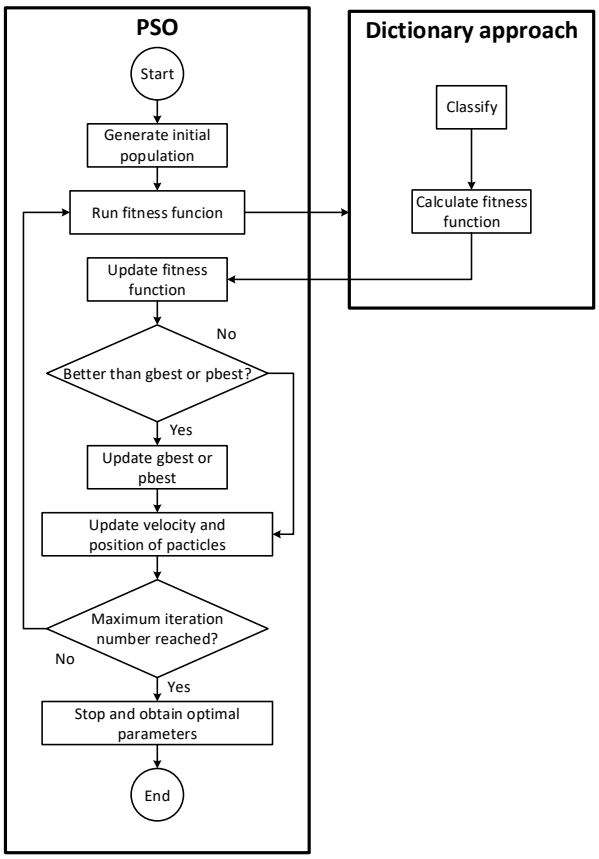

Fig. 1 Overview of the system.

4.3.1. The standard PSO method

The main idea of using PSO is to find an optimal value of polarity for each word. The position of the particle represents one potential solution, which can be represented as a vector with D-dimensions corresponding to the size of the dictionary. The initial population is generated randomly and then PSO algorithm is applied. The algorithm evaluates each particle based on the fitness function, sets up *pbest* for each particle and searches for the *gbest*. In the next iteration, a velocity of each particle is calculated based on its *pbest* and *gbest*, and the position of the particle is updated. The particle is evaluated again and *pbest* and *gbest* are updated. This process runs until a fixed number of iterations is done.

For experiments with standard PSO, the following parameters were used:

- inertia weight = 0.729844
- number of particles = 15000
- number of iterations = 100
- c_1 = 1.49618
- c_2 = 1.49618
- max velocity = 2

4.3.2. The BBPSO method

The main idea of using BBPSO is also to find an optimal value of polarity for each word. In contrast to the standard PSO, Bare-Bones PSO uses *pbest* and *gbest* to calculate mean and standard deviation for Gaussian distribution. It also randomly initiates the first population, then evaluates each particle based on the fitness function, sets up *pbest* for each particle and searches for the *gbest*. To update the position of the particle, BBPSO uses Gaussian distribution $\mathcal{N}(\mu_{id}, \sigma_{id})$ with the mean μ_{id} and standard deviation σ_{id}. μ_{id} and σ_{id} are calculated using the following Eq. 4 and Eq. 5:

$$\mu_{id} = \frac{(p_{gd} + p_{id})}{2} \tag{4}$$

$$\sigma_{id} = |p_{gd} - p_{id}| \tag{5}$$

where:
p^t_{gd} ... is a position of the *gbest* particle in d-th dimension
p^t_{id} ... is a *pbest* (personal best) position of the i-th particle in d-th dimension

$d = 1, 2, \ldots, D$ and D represents the number of words in the dictionary, $i = 1, 2, \ldots, N$, and N is the number of particles in the swarm.

The particle is evaluated again and *pbest* and *gbest* are updated. This process runs until a fixed number of iterations is done.

4.3.3. Fitness function

The fitness function is based on simple dictionary approach. It combines polarity values generated by PSO/BBPSO with words in the dictionary to create a temporary dictionary. The temporary dictionary classifies reviews in the dataset. Datasets are stored in preprocessed form. The input review is split into sentences and words. Each word is compared with the words in the dictionary

and if the word is found in the dictionary, the polarity value of the sentence is updated. If the word is positive, the polarity of the sentence increases and if the word is negative, the sentence polarity decreases. This process can be described by Eq. 6.

$$P_s = \sum_{i=1}^{w_s} pw_i \tag{6}$$

where:
P_s ... is the sentence's polarity
pw_i ... is the polarity of i-th word
w_s ... is a number of words in the sentence s

The polarity of the review is summed from polarities of all sentences. The class is assigned based on the polarity of the review. Precision and recall are calculated based on the comparison of the system assigned classes with the gold standard labels. A precision is calculated by Eq. 7 and it is a ratio of correctly evaluated positive reviews (tp) to all reviews marked by the algorithm as positive ($tp + fp$). A recall is calculated by Eq. 8 and it is a ratio of correctly evaluated positive reviews (tp) to all reviews labeled as positive ($tp + fn$). The same method of calculation was used to calculate the precision and recall for negative reviews. They are applied to calculate the F1 measure which is a harmonic mean between precision and recall. It is be calculated by Eq. 9.

Table 2 Contingency table for precision and recall

	Positive set	Negative set
Positive by algorithm	true positive (tp)	false positive (fp)
Negative by algorithm	false negative (fn)	true negative (tn)

$$Precision = \frac{tp}{(tp + fp)} \tag{7}$$

$$Recall = \frac{tp}{(tp + fn)} \tag{8}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{9}$$

The final values of the fitness function are derived from the macro-F1 measure which evaluates the performance on the unbalanced dataset. Macro-F1 is an average of F1 measures calculated for each class (Eq. 10). Thus, Marco-F1 shows the effectiveness in each class, independently of the size of the class.

$$Macro - F1 = \frac{F1(+) + F1(-)}{2} \tag{10}$$

where:
$F1(+)$ ... is the F1 measure for positive reviews
$F1(-)$ ... is the F1 measure for negative reviews

4.4. Combined approach

In order to adapt to a new language and classify comments which do not contain words from the current dictionary, the combined approach is applied. It combines dictionary based approach and a machine learning method. Dictionary-based approach classifies all reviews in the dataset. Then the dataset is split into two subsets, reviews classified by dictionary approach and unclassified reviews. The classified reviews are divided into the positive and the negative group and they are sorted from the most positive/negative to the less positive/negative. The positive group of reviews is compared with the negative one to find which group contains fewer comments. All reviews from the smaller group and the same number of reviews from the bigger group are selected to form a new balanced dataset. This dataset is used as a training dataset from the machine learning method. The distribution is very important because, if we had more comments with one polarity, it could influence the results.

The training dataset is used to calculate a probability that the word w from the sentence s is connected with class c (positive or negative). Thus if the word is found in the review, the probability P, that this word w is from class c is calculated by the simple probability method described in formula 11. Classes assigned by the dictionary approach are used to build a model.

$$P(w_c) = \frac{w_c}{w_d} \tag{11}$$

where:
$P(w_c)$ ... the probability that the word is from class c
w_c ... the number of occurrences of word w in class c
w_d ... the number of occurrences of word w in the whole dataset

In case that the word is not assigned to the specific class and the probability would be zero, a method which returns a very low number instead of zero is implemented.

After the model is built, it is used to classify the unclassified reviews from the original dataset. The polarity of the review consists of the probabilities of each word connected to the positive and negative class. The probability is computed by 12.

$$P_{sc} = \frac{\sum_{i=1}^{w_s} P(w_{ic})}{w_s} \tag{12}$$

where:
P_{sc} ... the probability that sentence is from class c
$\sum P(w_{ic})$ ... the summed probabilities of all words from the sentence s which are from class c
w_s ... the number of words in sentence s

The reviews are added to the positive class if the final positive probability is higher than the negative probability and vice-versa.

5. EXPERIMENTS AND RESULTS

5.1. Dictionary annotation

Our approach was tested on two datasets. A Slovak dataset contains 5242 reviews from different websites. It consists of 2573 positive and 2669 negative comments. Neutral comments were removed. The reviews refer to different domains such as electronics reviews, books reviews, movie reviews and politics. The dataset includes 155 522 words. To compare our approach with other works, a movie dataset [17], which contains 1000 positive and 1000 negative reviews collected from rottentomatos.com is used. The dataset is preprocessed and it is translated to Slovak using Google translator[2]. All datasets are labeled as positive or negative manually.

Each dataset was randomly split on ratio 90:10, which means that it is looking for an optimal solution on 90% of the dataset and the optimal solution is validated on the 10% of unseen comments. The same subsets are applied in all experiments, including the manually labeled dictionary. The manually labeled dictionary was evaluated on the same 10% subset.

The performance of all versions of dictionaries was compared and the results can be seen in Table 3.

Table 3 The comparison of F1-measures achieved by three different versions of two dictionaries.

dictionary	Slovak dataset	movie dataset
big dictionary labeled manually	0.767	0.629
big dictionary labeled by PSO	0.698	0.694
big dictionary labeled by BBPSO	**0.775**	**0.743**
small dictionary labeled manually	0.501	0.679
small dictionary labeled by PSO	0.509	0.727
small dictionary labeled by BBPSO	**0.528**	**0.738**

It can be seen, that standard PSO is able to find better values of polarity than human. Thus it outperforms human annotator in three cases and there is only one experiment in which human annotator achieved better results than standard PSO. The standard PSO labeled dictionary achieved better results in three experiments: the classification of the Slovak dataset using a small dictionary, the classification of movie reviews using big and also a small dictionary. Then BBPSO is applied and it achieved even better results than standard PSO in every experiment. It significantly outperforms both previous types of annotation, annotation by a human and annotation by the standard PSO.

Both dictionaries labeled by PSO and BBPSO respectively, outperformed the human labeled dictionary, which is interesting. There are several reasons for this. Human labeling can be biased which means it is based on intuition. It might not work well in some cases. It is not trained in any way and there is not any known data applied during human labeling. Our PSO based approach, which is similar to machine learning methods works well in learning optimal polarity values for words in the dictionary which is contradictive to the normal understanding in this field.

[2]https://translate.google.sk/

The movie dataset is used to compare our approach with a method described in work [21]. They achieved performance between 68.05% (a simple sentiment analysis using just words from the dictionary) and 76.37% (a sentiment analysis based on additional features). Our small dictionary labeled by BSO achieved performance 73.67% with the dictionary size 220 words which represent only 3.2% of the size of the dictionary (6793 words) proposed in the work by Taboada [21].

5.2. Combined approach

The combined approach was tested on the Slovak dataset. In this case, the dataset was split into two parts. The reviews labeled by dictionary approach created a training dataset and unclassified reviews created a testing set. The training dataset was balanced to equal number positive and negative reviews and it was used to build a probability model. The model was applied to testing dataset and the results can be seen in Table 4 and Table 5.

Table 4 The comparison of different approaches for opinion analysis using the big dictionary.

Approach	F1(+)	F1(-)	Macro F1
human dict. without CA	0.740	0.645	0.694
human dict. with CA	0.852	0.826	0.839
PSO dict. without CA	0.717	0.608	0.663
PSO dict. with CA	0.847	0.821	0.834
BBPSO dict. without CA	0.743	0.667	0.705
BBPSO dict. with CA	**0.869**	**0.860**	**0.865**

Table 5 The comparison of different approaches for opinion analysis using the small dictionary.

Approach	F1(+)	F1(-)	Macro F1
human dict. without CA	0.590	0.400	0.495
human dict. with CA	0.836	0.813	0.825
PSO dict. without CA	0.589	0.407	0.498
PSO dict. with CA	0.840	0.828	0.834
BBPSO dict. without CA	0.595	0.421	0.508
BBPSO dict. with CA	**0.847**	**0.834**	**0.841**

These results show that the unclassified comments degrade the performance. The macro F1 measure with unclassified reviews was around 0.694 using the big dictionary and 0.495 using the small dictionary. The original dictionary approach using the big dictionary was not able to classify 18%. The combined approach using the big dictionary classified 99.8%. The original dictionary approach using the small dictionary unclassified 40.2% and the combined approach using the small dictionary classified 99.7%. The results show that the combined approaches achieved better results in every experiment. It can be seen that dictionaries annotated by PSO and BBPSO achieved better results, because they provided slightly better results using dictionary approach.

6. CONCLUSION

Sentiment analysis in the minor languages using dictionary approach is not simple and requires a lot of human effort to prepare a good source of information, especially a dictionary. In this paper, an automated method for dictionary annotation is proposed. It uses optimization algorithms, such as Particle swarm optimization (PSO) and Bare-bones particle swarm optimization (BBPSO), to find optimal values of polarity for words in the dictionary. Then the combined approach is applied to adapt the dictionary and classify unclassified reviews from the dataset. Two dictionaries were created, to prove a better performance of the dictionary which was annotated by PSO or BBPSO respectively. The dictionaries annotated by the optimization algorithms outperformed the dictionary annotated by a human and provided better knowledge for the combined approach. In next stage, the combined approach built a probability model which was able to classify the reviews that did not contain words from the dictionary and reduced the number of unclassified reviews from 18% and 40.2%, respectively to 0.3%.

ACKNOWLEDGEMENT

The work presented in this paper was supported by the Slovak Grant Agency of the Ministry of Education and Academy of Science of the Slovak Republic under VEGA grant No. 1/0493/16 and by the Slovak Research and Development Agency under the contract No. APVV-16-0213 and the contract No. APVV-17-0267.

REFERENCES

[1] BACCIANELLA, S. – ESULI, A. – SEBASTIANI, F.: SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. Proceedings of LREC 2010.

[2] BAJPAI, R. – CAMBRIAa, E. – PORIA, S. – SCHULLER, B. W.: SenticNet 4: A Semantic Resource for Sentiment Analysis Based on Conceptual Primitives. COLING 2016.

[3] BASARI, S. H. – HUSSIN, B. – ANANTAA, I. G. P. – ZENIARJA, J.: *Opinion Mining of Movie Review using Hybrid Method of Support Vector Machine and Particle Swarm Optimization*, Procedia Engineering **53**, No. (2013) 453–462

[4] Van den BERGH, F. – ENGELBRECHT, A. P.: A study of particle swarm optimization particle trajectories. Information sciences, 2006.

[5] CHATURVEDI, I. – RAGUSA, E. – GASTALDO, P. – ZUNINO, R. – CAMBRIA, E.: *Bayesian network based extreme learning machine for subjectivity detection*, Journal of the Franklin Institute , No. (2017) 1–18 http://www.sciencedirect.com/science/article/pii/S0016003217303009

[6] ESCALANTE, H. J. – GARCÍA-LIMÓN, M. A. – MORALES-REYES, A. – GRAFF, M. – MONTES-Y-GÓMEZ, M. – MORALES, E. F. – MARTÍNEZ-CARRANZA, J.: *Term-weighting Learning via Genetic Programming for Text Classification*, Knowledge-Based Systems **83**, No. C (2015) 176–189 http://dx.doi.org/10.1016/j.knosys.2015.03.025

[7] GO, A. – BHAYANI, A. – HUNAG, L.: Twitter Sentiment Classification using Distant Supervision, Stanford University, 2013 http://cs.stanford.edu/people/alecmgo/papers/TwitterDistantSupervision09\.pdf,2013

[8] GUPTA, D. K. – Reddy, S. K. – Shweta, EKBAL, A.: PSO-ASent: Feature Selection Using Particle Swarm Optimization for Aspect Based Sentiment Analysis, Germany, Springer International Publishing, 2015.

[9] KACPRZYK, J. – PEDRYCZ, J.: Springer Handbook of Computational Intelligence, 2015.

[10] HU, M. – LIU, B.: Mining and Summarizing Customer Reviews. Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 04), New York, ACM, 2004.

[11] KENNEDY, J.: Bare bones particle swarms. Proceedings of the IEEE Swarm Intelligence Symposium (SIS), 2003.

[12] KENNEDY, J. – EBERHART R. C.: Particle swarm optimization. Proceedings of the IEEE International Conference on Neural Networks, IEEE press, 1995.

[13] MIKULA, M. – MACHOVÁ, K.: Classification of opinion in conversational content. Proceedings of the IEEE 13th International Symposium on Applied Machine Intelligence and Informatics, Slovakia, 2015.

[14] MOHAMMAD, S. M. – TURNEY, P. D.: Crowdsourcing a wordemotion association lexicon. Computational Intelligence, 2013.

[15] MOHAMMAD, S. M. – KIRITCHENKO S. – ZHU, X.: NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets. Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013), Georgia, 2013.

[16] NIELSEN, F.: A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. CoRR, 2011.

[17] PANG, B. – LEE, L. – VAITHYANATHAN, S.: Thumbs Up?: Sentiment Classification Using Machine Learning Techniques. Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing (EMNLP 02), Stroudsburg, Association for Computational Linguistics, 2002.

[18] dos SANTOS, C. N. – GATTIT, M.: Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts. Proceedings of COLING 2014 the 25th International Conference on Computational Linguistics, 2014.

[19] STRAPPARAVA, C. – VALITUTTI, A.: WordNet-Affect: an Affective Extension of WordNet. Proceedings of the 4th International Conference on Language Resources and Evaluation. European Language Resources Association (ELRA), 2004.

[20] STONE, P. J. – DUNPHY, D. C. – SMITH, M. S. – OGILVIE, D. M.: The General Inquirer: A Computer Approach to Content Analysis, The MIT Press, 1966.

[21] TABOADA, M. – BROOKE, J. – TOFILOSKI, M. – VOLL, K. – STEDE, M.: *Lexicon-based Methods for Sentiment Analysis. Computational Linguistics*, Computational Linguistics **38**, No. 2 (2011) 267–307

[22] TAN, S. – ZHANG, J.: *An Empirical Study of Sentiment Analysis for Chinese Documents*, Expert Systems with Applications: An International Journal **34**, No. 4 (2008) 2622–2629.

[23] TANG, D. – WEI, F. – QIN, B. – LIU, T. – ZHOU, M.: Coooolll: A Deep Learning System for Twitter Sentiment Classification. Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), 2014.

[24] THELWALL, M. – BUCKLEY, K. – PALTOGLOU, G. – CAI, D. – KAPPAS, A.: *Sentiment Strength Detection in Short Informal Text*, Journal of the American Society for Information Science and Technology **61**, No. 12 (2010) 2544–2558 http://dx.doi.org/10.1002/asi.21416

[25] WARRINER, A. B. – KUPERMAN, V. – BRYSBAERT, M.: Norms of valence, arousal, and dominance for 13,915 English lemmas. Behavior Research Methods, 2013.

Received December 19, 2017, accepted March, 21, 2018

BIOGRAPHIES

Martin Mikula obtained his degree (MSc.) in 2014 at the Department of Cybernetics and Artificial Intelligence at the Technical University in Košice. Currently, he is a postgraduate student. He studies business informatics at the Faculty of Electrical Engineering and Informatics at Technical University in Košice. His research interests are in big data, including knowledge discovery, data mining, text mining and sentiment analysis in conversational content.

Kristína Machová graduated (MSc.) in 1985 at the Department of Technical Cybernetics at the Technical University in Košice. She defended her PhD thesis in the field of machine learning in 1996. She workes as an associate professor at the Department of Cybernetics and Artificial Intelligence of the Faculty of Electrical Engineering and Informatics at the Technical University of Košice. Her scientific research focus is on mining within conversational content, the dictionary approach and the machine learning approach to the sentiment analysis with the focus on the opinion and emotion classification for an application in robotics, on authority identification, text document processing using classification and clustering machine learning methods.