DE GRUYTER OPEN
RIGA TECHNICAL UNIVERSITY
Applied Computer Systems
ISSN 2255-8691 (online)
ISSN 2255-8683 (print)
May 2018, vol. 23, no. 1, pp. 45–51
doi: 10.2478/acss-2018-0006
https://www.degruyter.com/view/j/acss

Sentiment Analysis in Latvian and Russian: A Survey

Rinalds Vīksna[1], Gints Jēkabsons[2*]
[1, 2]*Riga Technical University, Riga, Latvia*

***Abstract* – Social networking sites such as Facebook, Twitter and VKontakte, online stores such as eBay, Amazon and Alibaba as well as many other websites allow users to share their thoughts with their peers. Often those thoughts contain not only factual information, but also users' opinion and feelings. This subjective information may be extracted using sentiment analysis methods, which are currently a topic of active research. Most studies are carried out on the basis of texts written in English, while other languages are being less researched. The present survey focuses on research conducted on the sentiment analysis for the Latvian and Russian languages.**

***Keywords* – Machine learning, opinion mining, sentiment analysis, sentiment classification.**

I. Introduction

Nowadays, progressively more people are sharing their thoughts and experiences with their peers using social media, including social networks, blogs, forums, review sites etc. With the massive influx of the user-generated content, web content mining is attracting considerable attention due to its usefulness for identifying trends of public opinion. Sentiment analysis (also known as opinion mining) is a natural language processing task that aims at extracting such opinions and emotions found in free unstructured texts, typically into categories "positive", "neutral", and "negative" [1]. It can be viewed as a text classification problem with the attitude expressed in the text as the criterion of the classification.

Detecting sentiment of a text can be important for many applications, for instance, getting customer feedback about a brand or product [2], aggregating and summarising opinions in reviews for recommender systems [3], measuring effectiveness of political campaigns [4] and so on.

The research area concerned with the sentiment analysis of English texts has become very active [1], [5], [6], [7], giving much less attention to other languages [8], [9], [5]. The present paper examines research conducted on the sentiment analysis for the Latvian and Russian languages. The two languages were selected for this survey because they are the most used in social media in Latvia, the country which both authors are from.

The rest of the paper is organised as follows: Section II describes general approaches to sentiment analysis and lists the sub-tasks they involve. Section III summarises how various authors acquire annotated corpora and what data is publicly available as a result. Section IV discusses the text preprocessing and feature selection sub-task and how different authors tackle it. Section V explores sentiment classification methods and gives an overview of their achieved results. Section VI concludes the paper.

II. Approaches to Sentiment Analysis

Classifying sentiment of a text is generally done using either a machine learning approach or a lexicon-based approach [1], [5], [6], [7]. Machine learning approach uses a training dataset to train a classification model to differentiate among classes, i.e., sentiment categories. Success of machine learning approach mainly depends on the availability of features most useful for classification.

Lexicon-based approach uses a lexicon of tokens (words, phrases, emoticons, emoji and others) with known sentiment, and matches tokens from the lexicon with tokens of a text to classify its sentiment. Success of lexicon-based methods mainly depends on the quality of lexicon used.

Sentiment analysis is a complex problem; several sub-tasks are required to perform it on given texts (see Fig.1): data acquisition, data preprocessing, feature selection, classifier creation using the acquired data and evaluation of classification performance of the created classifier. The sub-tasks are further explored in the subsequent sections.

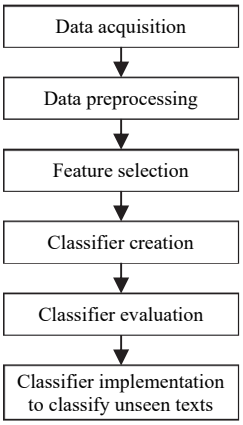

Fig.1. The sub-tasks required to perform the sentiment analysis on given texts.

III. Data Acquisition

A typical sentiment analysis system needs an annotated text corpus on which the system is trained and/or evaluated. Lexicon-based systems require a lexicon containing words labelled with sentiment class.

*Corresponding author's e-mail: gints.jekabsons@rtu.lv

In the English language, resources with annotated texts are widely available, e.g., [10], [11], [12]; however, in Russian and especially in Latvian only a few such resources are available. The present section explores various methods authors of the articles under survey have used to acquire and annotate such datasets.

Garkaje et al. [13] used corpus consisting of over 11 million user comments from the most popular Latvian news portals (Apollo, Tvnet and Delfi). 3000 of collected documents were manually classified into 3 categories: aggressive, non-aggressive and neutral. Each document was labelled by 2 annotators, and if the opinions did not match, the third annotator made final vote.

Peisenieks [14] collected 1.2 million tweets from Twitter. To select mostly tweets in the Latvian language, rough contour of Latvia was used as a Twitter query parameter. Peisenieks and Skadiņš then proceeded in [15] to create annotated corpus of tweets written in the Latvian language using a custom crowd-sourcing website for rating tweets into 3 classes. Final corpus consists of 383 positive, 627 neutral and 167 negative tweets, and it is available on Github [16]. The authors used Fleiss' kappa to measure agreement between annotators, and found reliability of agreement to be 0.284, which can be considered a fair agreement.

Nicmanis [17] reused corpus created by Peisenieks and Skadiņš [15] and also collected his own corpus from Twitter by using tweets from a single user and his followers. Collected tweets were rated by human annotators resulting in a corpus of 3131 annotated tweets (1085 positive, 1712 neutral and 334 negative). The corpus is publicly available on Github [18].

Gediņš [19] acquired 1 million tweets from Soon, Ltd. For model training purposes, tweets containing emoticons were extracted and labelled as positive or negative according to the emoticons. As a result, 13000 negative tweets, 130000 positive tweets and 750000 neutral tweets were collected. For training purposes, the number of used tweets for each sentiment was equalised: 13000 negative, 13000 positive and 13000 neutral tweets were used. Tweets which did not contain emoticons were considered as belonging to "mostly" neutral class as they could contain sentiment even if no emoticons were used. Additionally, tweets were rated by human annotator. By comparing human-annotated tweets with labels found using emoticons, 69.3 % agreement was achieved using 3 classes. However, 85.4 % tweets rated by human as positive and 94.8 % tweets rated by human as negative were also rated with the same label using emoticons. 1000 tweets classified as being neutral by absence of emoticons were checked by human, and only 65.3 % of tweets were found to be neutral therefore, reliability of "neutral" corpus was considered poor.

Špats and Birzniece [20] used lexicon of "Positive and Negative Sentiment Words" list from Pumpurs [21]. They reused labelled tweet corpus from [15], and also collected their own Latvian tweet corpus. They collected 90000 tweets and labelled them using a list of emoticons and emoji for positive and negative sentiment. Tweets which did not contain any sentiment according to their created lexicon (available in [22]) were considered neutral. This labelling process resulted in a corpus of 5556 noisy-labelled tweets.

Two Twitter corpora were created by Loukachevitch and Rubtsova [23] in Russian for an open evaluation task. The corpora were created for the telecom domain (5000 tweets for training and 3845 tweets for evaluation) and banking domain (5000 tweets for training and 4549 tweets for evaluation). Training corpus tweets were labelled by at least two human assessors. Their task was to label each tweet by estimating the tweet reputation-oriented attitude to the labelled entity. Tweets in the evaluation corpus were labelled by at least three human assessors. The authors noticed that some users did not want to seem too rude so they added positive emoticons to clearly negative or ironic messages, and that was one of the reasons why simple methods of tweet classification based on emoticons did not work well.

Loukachevitch and Chetviorkin [24] gathered user reviews for an open evaluation task on three topics: movies, books and digital cameras. Posts about films and books were collected from internet portal imhonet.ru and rated using a ten-point scale. Reviews about digital cameras were collected from Yandex-market site. Collected texts were labelled by experts using 2 (positive, negative), 3 (positive, negative, satisfactory) and 5 (excellent, good, satisfactory, bad, awful) classes, and Fleiss' kappa was calculated to assess agreement between experts.

Sakenovich [25] collected corpus consisting of 30000 news articles in the Russian language, which consisted of 11286 neutral, 10958 positive and 7756 negative human-labelled texts.

Tutubalina and Nikolenko [26] used a restaurant review corpus from SentiRuEval-2015 dealing with aspect-based sentiment classification [27]. For training they used 17132 unlabelled reviews, for evaluation they used SentiRuEval's training and testing data with 201 and 203 reviews, respectively.

Shalunts and Backfried [28] used a training dataset consisting of 32 news articles in English, 32 in German and 48 in Russian. Testing dataset consists of 50 articles in each language. Each article was rated by 3 experts into 4 classes – positive, negative, neutral and mixed. A sentiment lexicon was acquired by human expert annotating a list of words and extending it to cover the required topics.

Galinsky et al. [29] collected 98134 reviews (63088 positive and 35046 negative) from torg.mail.ru and restoclub.ru websites as a training corpus, as well as 37882 reviews (26807 positive and 11075 negative) from TripAdvisor web site as a test corpus. The training corpus was then augmented to obtain a larger corpus by duplicating the reviews and replacing their words with synonyms using synonym dictionaries. Only words which are reflexive synonyms (word A is a synonym to word B, and B is a synonym to A) are used. Using this method, the training corpus was extended to 195372 reviews. Another corpus augmentation method used was adding new adjectives to nouns. First, the corpus is analysed to estimate the typical adjectives that occur next to a specified noun, then, according to the computed probabilities, reviews are augmented by adding new adjectives to nouns that do not yet have an associated adjective next to them in the text. Using this method, the

training corpus was extended to 196268 reviews. The authors demonstrated that the augmentation using synonyms can considerably increase sentiment classification performance while the augmentation using adjectives actually decreases the performance.

Bobichev et al. [30] used 10194 news articles collected from censor.net.ua. Of those, 2018 texts were chosen for manual annotation. Annotation was done by students into positive, negative and neutral texts. A training corpus was created from 331 texts that were annotated by more than three annotators with the same sentiment category. The low agreement among annotators can be explained by the fact that usually news articles are written either objectively or so that they look objective.

As one can see, to study the sentiment analysis in Latvian, most authors of the articles under survey created their corpora from Twitter [14], [15], [17], [19], [20]. Papers on the sentiment analysis in Russian are more diverse: apart from Twitter [23], authors use news articles [25], [28], [30] and reviews [24], [26], [29]. In the Latvian language, human annotated corpora are small, even if combined together, so researchers are either using noisy-labelling or lexicon-based methods, which do not require a large annotated corpus.

IV. Data Preprocessing and Feature Selection

Sentiment analysis is a classification problem. Using a classifier to classify an object requires it to be represented as a vector of features. The authors of all examined articles used some preprocessing before transforming documents into feature vectors suitable for use by a classification method as texts written online by users have heterogeneous structure and writing styles and may contain transliteration, different languages as well as mistakes. Moreover, for example, tweets frequently contain hashtags marked by "#", usernames marked by "@", retweets marked by "RT", and links, all of which usually do not carry information about sentiment expressed.

Preprocessing is mostly used for cleaning and normalization of text. In the preparation of data, the first stage of preprocessing might involve complete removal of some training samples:

- Discard duplicate and automatically generated texts – this can be especially important for tweets because of the constant stream of automatic tweets and retweets that appear in large numbers and have approximately the same content. This step is used in [14], [20];
- Discard texts not written in the target language – this can be necessary for a source that potentially contains texts in diverse languages (e.g., a typical social network). The reason is that for a sentiment classification in one language a text in another language is, in fact, just noise as most of the words cannot be recognised or are recognised incorrectly. This step is used in [13], [15].

After the first stage of data preparation as well as at the stage when a sentiment analysis system is ready to classify new texts, generally, text preprocessing may include the following activities:

- Remove HTML tags – if a text contains any HTML tags, e.g.," " or "
", they should be removed. This is used in [25];
- Remove numbers or replace them with a special tag – numbers found in texts are usually unique and, therefore, are not useful as features. This is used in [25];
- Remove punctuation marks – a very typical part of cleaning a text;
- Transform all letters to either lowercase or uppercase letters – also a very typical part of cleaning a text;
- Remove stop words – stop words are words which carry little meaning and therefore are considered safe to remove. This is used in [13], [25], [26];
- Stemming/Lemmatization – used to reduce inflectional and derivationally related forms of a word to a common base form – either stem or lemma. This has additional importance for languages such as Latvian or Russian, as these languages use inflections, suffixes and prefixes which multiply the number of unique words (and, therefore, potential features) making the problem of sparseness and noise in textual data (see, e.g., [1]) even worse. This step is used in [13], [25], [28];
- Remove transliteration – the Latvian alphabet contains a number of special letters such as "ā", "č", "ū". Instead of these, a user may write double letters: "aa", "ch", "uu". This step replaces such combinations back to the correct letter [13];
- Remove usernames (starting with "@") – tweets may contain mentioning of usernames, which usually are unique and are not useful as features [19], [20];
- Remove/replace hashtags (starting with "#") – these are usually not useful as features and are removed [20] or replaced with words, if, e.g., recognised as a word by a dictionary [17];
- Remove web links or replace them with a special tag – links found in texts are usually unique and, therefore, are not useful as features [19], [20], [25];
- Tokenization of sentences into primitives – this step converts a sentence into a vector of words or symbols further used as features in the feature vector.

Result of preprocessing is a clean normalized sequence of tokens, which can be mapped to feature vectors. To do the mapping, the authors of the articles under survey used either "Bag of words" approach (done by most authors), word embeddings [17], [25] or character-level embeddings [30]. Additional features of text are considered in [24], such as a number of exclamation or question marks or presence of obscene language in a text. Gediņš [19] uses a special feature, which replaces rarely used positive or negative words. Some authors also include emoticons or emojis in their feature vectors – either directly [28] or as a separate feature counting them [19], [17]. Shalunts and Backfried [28] used exclamation marks, repeated letters and capitalization as separate sentiment "booster" features that amplify detected sentiment. Gulbinskis [31] extracted specific phrases, which carried information about sentiment in a given text. To obtain such phrases, custom phrase rules were developed and then part of speech analyser (SemTi-

Kamols morphological analysis tool [32]) was used to extract phrases matching given rules.

Bobichev et al. [30] experimented with two methods for selection of the most informative set of features: 1) "Correlation-based Feature Subset Selection" which evaluates features by considering the individual predictive ability of each feature together with redundancy between features; 2) "Information Gain Evaluation" which considers a feature according to information gain with respect to the class. Both methods performed similarly and both were better than if the whole bag of words was used without any feature selection.

As one can see, features which some researchers consider irrelevant, others include as part of their feature vector, and, as a result, the list of the used features varies greatly among researchers. This leads to a conclusion that text preprocessing and selection of features are a topical issue that requires further research.

V. Classification Methods

The section explores various classification methods the authors of the articles under survey use to classify a text according to its sentiment and summarises their achieved classification performance.

The articles report the classification performance in the form of the following measures: Precision, Recall, F1 and Accuracy. They are calculated as follows:

$$Precision = \frac{tp}{tp + fp},$$

$$Recall = \frac{tp}{tp + fn},$$

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall},$$

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn},$$

where *tp* is the number of true positives, *tn* is the number of true negatives, *fp* is the number of false positives and *fn* is the number of false negatives. Here, "positive" and "negative" refer to the predicted class label of the classifier, while the "true" and "false" refer to whether the predicted label was correct or incorrect compared to the expected label. In general, a greater score is better with any of the measures; however, scores obtained using one measure are not comparable with scores obtained using another measure.

Table I gives a summary of articles used in this survey – classification methods used, language and domain considered, classes used, as well as achieved results. As can be seen, all the authors used a machine learning approach except the authors of [20], [26], [28] who (also) used a lexicon-based approach.

Garkaje et al. [13] used Naïve Bayes classifier. It was observed that inter-annotator agreement was 78 %, which was established as the upper bound of classifier accuracy. The best classification results were achieved when the text was normalized and transliteration removed – overall accuracy of 72.2 % with F1 measure of 33.1% for the aggressive class and F1 measure of 82.4 % for the non-aggressive class.

Peisenieks and Skadiņš [15] translated Latvian tweets into English using Google Translate (https://translate.google.lv), Bing Translator (https://www.bing.com/translator) and Tilde Translator (https://translate.tilde.com) and estimated sentiments of the tweets using three sentiment analysis tools available online: AlchemyAPI (https://www.alchemyapi.com), Textalytics (https://www.meaningcloud.com) and Semantria (https://www.lexalytics.com). All nine combinations (three translation tools with three sentiment analysis tools) were evaluated and compared. Peisenieks found that such an approach achieved overall accuracy in the range of 45.6–76.0 %, with best combination being Bing Translator together with AlchemyAPI achieving the highest overall accuracy of 76.0 %. However, all combinations performed poorly while classifying neutral tweets – overall accuracy was 21.3–35.5 %; and again the best performance was demonstrated by the combination of Bing Translator and AlchemyAPI.

The tweet corpus gathered by Nicmanis [17] was too small – the results achieved by training on the corpus were too poor so the author decided not to publish them.

Gulbinskis [31] used Pointwise Mutual Information and Information Retrieval (PMI-IR) algorithm [33] to analyse sentiment of posts and comments in various online resources. This algorithm does not require training data; instead it uses number of results returned from a web search engine in response to query containing a given phrase together with some sentiment carrying a word from lexicon available at [34] from server manatee.ailab.lv to infer sentiment polarity. Accuracy for 2 classes "positive" and "negative" was 75 %.

Gediņš [19] trained Naïve Bayes classifier using word unigrams, bigrams, unigrams with bigrams and unigrams with morphological information. The accuracy achieved was 70.1 %, 66.1 %, 70.8 % and 71.4 %, respectively. Worsening of results when using bigrams was explained by the increased number of less expressive features. It was found that 13 000 Latvian tweets contained 75 454 unique words, while English corpora of about the same size contained 4227 to 9045 unique words. Therefore, the main work was focused on selecting features, which were most common and expressive. Using a tool developed by Paikens [35], stemming was performed and model was retrained using roots of words as features and accuracy of 69 % was achieved. Best results were achieved by dropping words, which occurred in the corpus less than predefined number of times. Optimum was determined to be 7 occurrences, reducing the number of features (words) from 75 454 to 5260 while achieving accuracy of 75.8 %.

Špats and Birzniece [20] compared Naïve Bayes method with a lexicon-based approach. Naïve Bayes achieved 62 % accuracy when trained with a human-labelled tweet corpus and 55 % when trained with a noisy-labelled tweet corpus. The lexicon-based approach achieved 73 %.

Loukachevich and Rubtsova [23] report on results of various classification methods (Support Vector Machine (SVM), maximum entropy classifier, rule-based classifier) on the three-class (positive / neutral / negative) problem. The best results in terms of F1 measure were 48.8 % for the telecom domain (against a baseline of 18.2 %) using SVM, and 36.0 % for the banking domain (against a baseline of 12.7 %) using maximum entropy classifier on word n-grams, symbol n-grams and topic modelling results. Independent expert labelling obtained F1 of 70.3 % for telecom tweets, which could be considered a maximum possible performance value in this task. Low F1 values were explained by the difficulty of a task and the limited size of a training corpus.

Loukachevich and Rubtsova [24] report that the best results in their open evaluation task were achieved using SVM and maximum entropy classifiers. The results in the three domains were the following: accuracy with two classes: 83.1–96.1 %, accuracy with three classes: 69.4–75.2 %, accuracy with five classes: 40.7–51.3 %; F1 with two classes: 66.9–71.5 %, F1 with three classes: 48.0–56.0 %, F1 with five classes: 33.6–40.2 %.

Sakenovich [25] used Long Short-Term Memory (LSTM) recurrent neural network. LSTM considers sequential dependencies of words in a text. Furthermore, it overcomes the common exploding or vanishing gradient problems of recurrent neural networks. The best performing model was a stacked two layer LSTM, which achieved average precision of 84.5 %, recall of 86.4 % and accuracy of 86.3 %.

Tutubalina and Nikolenko [26] developed a technique to extract and expand a lexicon for a given aspect. For an aspect, the following features are extracted: lowercase character n-grams, lexicon-based unigrams, context unigrams and bigrams, aspect based bigrams; as well as lexicon-based features: max and min sentiment score, total and averaged sums of the word sentiment scores. They compared a manually constructed general purpose lexicon (baseline classifier) with a maximum entropy classifier built using the generated lexicon. Models using the generated aspect based lexicon showed on average 1–2 % improvement over a model using a manually built lexicon. They achieved precision of 74.8 %, recall of 66.3 % and F1 of 69.1 % while a baseline model achieved precision of 73.8 %, recall of 65.7 % and F1 of 67.6 %.

TABLE I

SUMMARY OF ARTICLES USED IN THE SURVEY

Classification methods used	Language	Data source and domain	Classes	Result	Notes	Reference
Naïve Bayes classifier	Latvian	General news from news portals	Aggressive, Non-aggressive	Accuracy 72.2 %, aggressive F1 32.9 %, non-aggressive F1 82.4 %	Inter-annotator agreement was 78 %	Garkaje et al. [13]
Machine translation and existing sentiment analysis tools for English	Latvian translated to English	Twitter (multi-domain)	Positive, (Neutral), Negative	Accuracy 76 % without neutral class and 35.5 % with neutral class	Best result was achieved using Bing Translator + AlchemyAPI	Peisenieks and Skadiņš [15]
PMI-IR algorithm	Latvian	Blogs, news, Twitter (multi-domain)	Positive, Negative	Accuracy 75 %		Gulbinskis [31]
Naïve Bayes classifier	Latvian	Twitter (multi-domain)	Positive, Negative	Accuracy 75.8 %		Gediņš [19]
Lexicon-based and Naïve Bayes	Latvian	Twitter (multi-domain)	Positive, Neutral, Negative	Accuracy 73 % for a lexicon-based method, 62 % for Naïve Bayes	Accuracy 55 % for Naïve Bayes with noisy-labelled data	Špats and Birzniece [20]
SVM, maximum entropy classifier, rule-based classifier	Russian	Twitter (telecom and banking domain)	Positive, Neutral, Negative	F1 48.8 % for the telecom domain; F1 36.0 % for the banking domain	Expert labelling F1 was 70.3 %. Baseline: 18 % for telecom, 13 % for banking	Loukachevich and Rubtsova [23]
SVM, maximum entropy classifier	Russian	Reviews of movies, books, digital cameras	2, 3 and 5 sentiment classes	Accuracy: 2 classes 83–96%, 3 classes 69–75 %, 5 classes 41–51 %	F1: 2 classes 67–72 %, 3 classes 48–56 %, 5 classes 34–40 %	Loukachevich and Chetviorkin [24]
LSTM recurrent neural network	Russian	General news from news portals	Positive, Neutral, Negative	Precision 84.5 %, recall 86.4 %, accuracy 86.3 %		Sakenovich [25]
Lexicon-based approach with maximum entropy classifier	Russian	Restaurant reviews	Positive, Neutral, Negative	Precision 74.8 %, recall 66.3 %, F1 69.1 %	Baseline: precision 73.8 %, recall 65.7 %, F1 67.6 %	Tutubalina and Nikolenko [26]
Lexicon-based (SentiSAIL)	Russian	General news from news portals	Positive, Neutral, Negative	Accuracy 90 %	Inter-annotator agreement was 92.7 %	Shalunts and Backfried [28]
Convolutional Neural Network	Russian	Reviews of restaurants and products	Positive, Neutral, Negative	Accuracy 87.0 % on the 1st test set and 71.6 % on the 2nd test set		Galinsky et al. [29]
SVM and three types of Naïve Bayes	Russian	General news from a news portal	Positive, Neutral, Negative	Best results by Naïve Bayes: F1 80.2–84.5 %, depending on a domain	Other methods: F1 53.3–81.4 % depending on a domain	Bobichev et al. [30]

Shalunts and Backfried [28] developed a lexicon-based sentiment classification method called SentiSAIL. The method implements three algorithms to calculate sentiment scores: Maximization – positive and negative scores are taken as the scores of most positive and most negative terms; Averaging – positive and negative scores are calculated as average of all its positive and negative scores; Aggregation – positive and negative scores are obtained by aggregating scores of all positive and negative words, bounded by 5 for positive score values and − 5 for negative score values. The score for the entire text is calculated as average of scores for all sentences in a document. Final sentiment class is assigned using thresholding, i.e., if a value passes a certain threshold it is assigned a respective class. SentiSAIL achieved an average test set accuracy of 90 %.

Galinsky et al. [29] used character-level embeddings with a convolutional neural network. With the unaugmented training corpus, they achieved accuracy of 84.6 % for their first test set and 71.6 % for their second test set. With the training corpus augmented with synonyms (see Section III above), they achieved accuracy of 87.0 % for their first test set and 70.2 % for their second test set.

Bobichev et al. [30] experimented with several machine learning algorithms – SVM, Bernoulli Naïve Bayes, Multinomial Naïve Bayes, Discriminative Multinomial Naïve Bayes. The best results were achieved using the Bernoulli Naïve Bayes algorithm – F1 measure of 84.5 % in the economic domain, 80.2 % in the social domain and 81.7 % in sports domain. Other methods achieved F1 measure of 53.3–81.4 % depending on a domain.

For sentiment analysis in Latvian, the lack of large annotated corpus does not allow for successful training of neural networks or other complex classification models; therefore, authors use methods that do not require large corpora. As one can see, they either use Naïve Bayes or methods that do not require any training corpora. All methods achieved similar results – accuracy in the range of 72–76 %.

In contrast, for sentiment analysis in Russian, a wider range of methods have been used: lexicon-based, SVM, convolutional and LSTM neural networks, rule-based methods, maximum entropy, as well as various types of Naïve Bayes. This list matches the list of the most popular methods for sentiment analysis in English [1].

Since for the Russian language large annotated corpora are available, neural networks can be used with good success – for sentiment classification into three classes two of the best results were achieved by Sakenovitch [25] (using LSTM recurrent neural network trained on 30000 labelled texts) and Galinsky [29] (using Convolutional Neural Network trained on 195372 labelled texts). It should be noted that, while Shalunts and Backfried [28] reported that their lexicon-based method achieved an impressive accuracy of 90 %, this might be at least partly because their corpus had much narrower domain than corpora of other authors.

While most of the best results for English have been achieved using SVMs [1], whether the method is able to achieve similar success for Latvian or Russian is still an open question. It should be noted that currently there are no studies published that would experiment with SVMs for the sentiment analysis in Latvian.

VI. Conclusion

The paper has presented an overview of research conducted on the sentiment analysis for texts in the Latvian and Russian languages. Approaches and methods for text acquisition, text preprocessing and sentiment classification used by various authors were compiled and summarised. After analysing these articles, the following main conclusions can be drawn. First, enhancements of sentiment classification methods as well as text preprocessing and feature selection are still an open field for research. Second, the Latvian language still lacks sufficiently large high quality annotated text corpora to train complex classification models with success. Here, a possible area for future research may be preparation of the necessary large annotated corpora (perhaps partly as an integration of the existing corpora discovered during the present study). Third, the classification results from the various authors cannot be easily compared as they are acquired using different corpora and incompatible performance criteria. Here, a possible area for future research is performing consistent comparison of the methods using the same corpora and the same performance measures for the languages at hand. Fourth, many less popular classification methods that have already been tried for English, are currently completely neglected for Latvian and Russian, e.g., Random Forest, Boosting, Fuzzy Logic, Radial Basis Function Neural Networks and others [1] (in addition to methods such as SVMs, rule-based methods and neural networks that have not been tried for Latvian yet but have already been tried for Russian). Experimenting with these methods is yet another possible area for future research.

References

[1] K. Ravi and V. Ravi, "A survey on opinion mining and sentiment analysis: Tasks, approaches and applications," *Knowledge-Based Systems*, vol. 89, pp. 14–46, 2015. https://doi.org/10.1016/j.knosys.2015.06.015

[2] Thomson Reuters, "Thomson Reuters Adds Unique Twitter and News Sentiment Analysis to Thomson Reuters Eikon" [Online]. Available: https://www.thomsonreuters.com/en/press-releases/2014/thomson-reuters-adds-unique-twitter-and-news-sentiment-analysis-to-thomson-reuters-eikon.html. [Accessed: Mar.8, 2018].

[3] L. Chen, G. Chen, and F. Wang, "Recommender Systems Based on User Reviews : The State of the Art," *Systematic Reviews 2015*, vol. 4, no. 5, 2015.https://doi.org/10.1007/s11257-015-9155-5

[4] A. Ceron, "Enlightening the voters: The effectiveness of alternative electoral strategies in the 2013 Italian election monitored through (sentiment) analysis of Twitter posts," *European Consortium for Political Research*, pp. 1–25, 2013.

[5] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Engineering Journal*, vol. 5, pp. 1093–1113, 2014. https://doi.org/10.1016/j.asej.2014.04.011

[6] S. Vohra and J. Teraiya, "Applications and Challenges for Sentiment Analysis : A Survey," *International Journal of Engineering Research and Technology,* vol. 2, no. 2, pp. 1–6, 2013.

[7] V. A. Kharde and S. S. Sonawane, "Sentiment Analysis of Twitter Data: A Survey of Techniques," *International Journal of Computer Applications*, vol. 139, no. 11, pp. 5–15, 2016. https://doi.org/10.5120/ijca2016908625

[8] K. Dashtipour, S. Poria, A. Hussain, E. Cambria, A. Y. A. Hawalah, A. Gelbukh, and Q. Zhou, "Multilingual Sentiment Analysis: State of the Art and Independent Comparison of Techniques," *Cognitive Computation*, vol. 8, pp. 757–771, 2016. https://doi.org/10.1007/s12559-016-9415-7

[9] M. Korayem, K. Aljadda, and D. Crandall, "Sentiment/subjectivity analysis survey for languages other than English," *Social network analysis and mining*, vol. 6, no. 1, pp. 1–28, 2016. https://doi.org/10.1007/s13278-016-0381-6

[10] "Datasets - Linked Data Models for Emotion and Sentiment Analysis Community Group." [Online]. Available: https://www.w3.org/community/sentiment/wiki/Datasets. [Accessed: Mar. 8, 2018].

[11] A. Esuli and F. Sebastiani, "SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining," *Proceedings of 5th Conference Language Resources and Evaluation*, pp. 417–422, 2006.

[12] "MPQA Resources." [Online]. Available: http://mpqa.cs.pitt.edu/. [Accessed: Mar. 8, 2018].

[13] G. Garkaje, E. Zilgalve, and R. Dargis, "Normalization and Automatized Sentiment Analysis of Contemporary Online Latvian Language," *Frontiers in Artificial Intelligence and Applications*, vol. 268, pp. 83–86, 2014. https://doi.org/10.3233/978-1-61499-442-8-83

[14] J. Peisenieks, "Mašīntulkošanas iespējas Twitter sīkziņu sentimenta analīzē," B. S. thesis, University of Latvia, Latvia, 2014.

[15] J. Peisenieks and R. Skadiņš, "Uses of Machine Translation in the Sentiment Analysis of Tweets," *Frontiers in Artificial Intelligence and Applications*, vol. 268, pp. 126–131, 2014.

[16] J. Peisenieks, "Latvian tweet sentiment corpus." [Online]. Available: https://github.com/FnTm/latvian-tweet-sentiment-corpus. [Accessed: Mar. 10, 2018].

[17] D. Nicmanis, "Sabiedrības attieksmes modelēšana, izmantojot sentimenta analīzi," B.S. thesis, University of Latvia, Latvia, 2017.

[18] D. Nicmanis, "LV twitter sentiment corpus." [Online]. Available: https://github.com/nicemanis/LV-twitter-sentiment-corpus. [Accessed: 10-Mar-2018].

[19] K. Gediņš, "Automātiskā teksta emocionālās noskaņas noteikšana latviešu valodā," B. S. thesis, University of Latvia, Latvia, 2013.

[20] G. Špats and I. Birzniece, "Opinion Mining in Latvian Text Using Semantic Polarity Analysis and Machine Learning Approach," Complex Systems Informatics and Modeling Quarterly, no. 7, pp. 51–59, 2016. https://doi.org/10.7250/csimq.2016-7.03

[21] "Latvian positive and negative sentiment words." [Online]. Available: https://github.com/pumpurs/SentimentWordsLV. [Accessed: Mar. 10, 2018].

[22] G. Špats, "Resources for opinion mining for written content classification in Latvian text." [Online]. Available: https://github.com/gatis/om/tree/master/lexicon. [Accessed: Mar. 10, 2018].

[23] N. Loukachevitch and Y. Rubtsova, "Entity-Oriented Sentiment Analysis of Tweets: Results and Problems Natalia," *Proceedings of the 18th International Conference on Text, Speech, and Dialogue, Lecture Notes in Computer Science*, vol. 9302, pp. 551–559, 2015. https://doi.org/10.1007/978-3-319-24033-6_62

[24] N. V. Loukachevitch and I. I. Chetviorkin, "Open evaluation of sentiment-analysis systems based on the material of the Russian language," *Scientific and Technical Information Processing*, vol. 41, no. 6, pp. 370–376, 2014. https://doi.org/10.3103/S0147688214060057

[25] N. S. Sakenovich and A. S. Zharmagambetov, "On One Approach of Solving Sentiment Analysis Task for Kazakh and Russian Languages Using Deep Learning," *Computational Collective Intelligence, Lecture Notes in Computer Science*, vol. 10449, pp. 537–545, 2017. https://doi.org/10.1007/978-3-319-45246-3_51

[26] E. Tutubalina and S. Nikolenko, "Constructing Aspect-Based Sentiment Lexicons with Topic Modeling," *Proceedings of 5th Conference on Analysis of Images, Social Networks, and Text*, vol. 10716, pp. 208–220, 2017. https://doi.org/10.1007/978-3-319-52920-2_20

[27] N. V. Loukachevitch, P. D. Blinov, E. V. Kotelnikov, Y. V. Rubtsova, V. V. Ivanov, and E. V. Tutubalina, "SentiRuEval: Testing Object-oriented sentiment analysis systems in Russian," *Computational Linguistics and Intellectual Technologies,* vol. 2, no. 14, pp. 3–15, 2015.

[28] G. Shalunts and G. Backfried, "SentiSAIL: Sentiment Analysis in English, German and Russian," *Proceedings of the 11th International Conference on Machine Learning and Data Mining in Pattern Recognition*, vol. 9166, 2015. https://doi.org/10.1007/978-3-319-21024-7_6

[29] R. Galinsky, A. Alekseev, and S. Nikolenko, "Improving Neural Network Models for Natural Language Processing in Russian with Synonyms," *Proceedings of AINL FRUCT 2016 Conference*, vol. 3, pp. 45–51, 2016.

[30] V. Bobichev, O. Kanishcheva, and O. Cherednichenko, "Sentiment analysis in the Ukrainian and Russian news," *Proceedings of 2017 IEEE 1st Electrical and Computer Engineering UKRCON 2017*, pp. 1050–1055, 2017. https://doi.org/10.1109/UKRCON.2017.8100410

[31] I. Gulbinskis, "Digitālo tekstu sentimenta analīze," B. S. thesis, University of Latvia, Latvia, 2010.

[32] "SemTi-Kamols." [Online]. Available: http://www.semti-kamols.lv/?sadala=220. [Accessed: Mar. 10, 2018].

[33] P. D. Turney, "Thumbs up or thumbs down? Semantic Orientation applied to Unsupervised Classification of Reviews," *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, no. July, pp. 417–424, 2002.

[34] "Latviešu valodas tekstu korpuss." [Online]. Available: http://www.korpuss.lv/. [Accessed: Mar. 8, 2018].

[35] P. Paikens, "Latvian morphology module." [Online]. Available: https://github.com/PeterisP/morphology. [Accessed: Mar. 8, 2018].

Rinalds Vīksna, *B. sc. ing.,* is a Master student at the Department of Software Engineering, Institute of Applied Computer Systems, Riga Technical University, Latvia. He obtained his Bachelor degree at Riga Technical University in 2016. His current research interests include sentiment analysis.
E-mail: rinaldsviksna@gmail.com

Gints Jēkabsons, *Dr. sc. ing.,* is an Assistant Professor and Researcher at the Institute of Applied Computer Systems, Riga Technical University, Latvia. He obtained his Doctoral degree at Riga Technical University in 2009. His current research interests include machine learning and natural language processing.
E-mail: gints.jekabsons@rtu.lv
ORCID iD: https://orcid.org/0000-0002-9575-2488