Human Language Technologies – The Baltic Perspective
K. Muischnek and K. Müürisep (Eds.)

doi:10.3233/978-1-61499-912-6-112

Latvian Tweet Corpus and Investigation of Sentiment Analysis for Latvian

Mārcis PINNIS [1]
University of Economics and Culture, Lomonosova 1/5, Riga, Latvia
Tilde, Vienibas gatve 75A, Riga, Latvia

Abstract. We present the Latvian Tweet Corpus and its application in sentiment analysis by comparing four different machine learning algorithms and a lexical classification method. We show that the best results are achieved by an averaged perceptron classifier. In our experiments, the more complex neural network-based classification methods (using recurrent neural networks and word embeddings) did not yield better results.

Keywords. social networks, tweet corpus, sentiment analysis, Latvian

1. Introduction

Social networks are used by the majority of Internet users (over 71% according to Statista[2], a leading provider of market and consumer data). This large Internet user presence in social networks drives companies to keep their own social network presence and to analyse the behaviour of their customers, including the public opinion (including sentiment) expressed about the companies' products and services as well as their competitors' products and services. For this, companies require social network analytics solutions that provide sentiment analysis functionality where the users can track the change of public sentiment over longer time-spans and identify events that are responsible for improvement or degradation of public sentiment. However, companies are not the only entities that can benefit from keeping track of user/customer sentiment. Knowing what the public thinks is also important for various organisations, including political organisations.

In this work, we focus on the corpus aspects of sentiment analysis. I.e., the development of a tweet corpus for Latvian (the Latvian Tweet Corpus (LTC)) that, apart from sentiment analysis, can be used also for a variety of other tasks including (but not limited to) political discourse analysis in social networks (i.e., Twitter in this work), communication behaviour analysis (and other applications in the field of computational social sciences), troll identification (i.e., identification of users that deliberately post offensive, provocative and often false messages), social network language analysis (i.e., for the development of methods for grammar/style correction in social network messages), question answering system development, and many other tasks where a social network message corpus may be required.

[1] Corresponding Author: Mārcis Pinnis; E-mail: marcis.pinnis@tilde.lv.

[2] https://www.statista.com/statistics/260811/social-network-penetration-worldwide/

Previous work on sentiment analysis for Latvian has been mostly focussed on lexicon-based sentiment analysis methods, i.e. methods where positive and negative word lists (gazetteers) are used to determine the polarity of a text message [1,2]. Cross-lingual sentiment analysis for Latvian has been previously analysed also by Peisenieks and Skadiņš [3]. Because open source sentiment analysis tools for Latvian were not available, however, tools for English had been widely researched (as shown by the numerous shared tasks on sentiment analysis, e.g. [4,5,6,7]), they used machine translation systems to translate tweets from Latvian into English in order to enable the use of English sentiment analysis tools. This study resulted in a sentiment annotated corpus of 1177 tweets that was used also in the experiments detailed below.

In this paper, we provide details on the semi-automatic development of the sentiment-annotated tweet corpus for LTC and experiments on developing sentiment analysers for Latvian. In our experiments, we compare different neural network architectures for sentiment analysis with a lexical classification method. We experimented with simpler architectures based on an averaged perceptron implementation, and more complex architectures that utilise pre-trained word embeddings (e.g., skip-gram models) and recurrent neural networks (RNN). We show that the simpler perceptron-based implementation can outperform the newer and more complex classification methods.

2. Latvian Tweet Corpus

The Latvian Tweet Corpus is a collection of tweets that have been collected during the time-frame from August 2016 till July 2018. The corpus consists of tweets from users of four categories: 1) politicians (including members of the parliament, deputies of Riga, ministers, the president, and other politicians that are active on Twitter), political parties, or government institutions, 2) large Latvia-based companies that are actively communicating with their customers on Twitter, 3) media agencies and journalists, and 4) other users interacting with users from the aforementioned user categories. The list of users was defined manually and consists of 865 entries.

In order to collect the LTC, we defined queries for every user (1095 in total). The queries were used to find tweets that mention the monitored users. For this purpose we used the Standard search API[3] of Twitter. Additionally, tweets from user timelines were collected to ensure that we would collect also all tweets created by the monitored users. The Twitter API was executed once every six seconds (due to restrictions enforced by Twitter), therefore, productive queries and users were dynamically prioritised.

Each tweet in the corpus (see Figure 1 for an example) is stored by preserving the most important meta-data for communication behaviour analysis and sentiment analysis, i.e., the message itself, information about the user who wrote the tweet, and information about the connection of the tweet to other tweets (i.e., whether the tweet was a reply to or a re-tweet of a different tweet). Additionally, automatic sentiment scores are added to every tweet enabling corpus-based sentiment analysis. E.g., Figure 2 depicts communication between politicians during the last year (with a minimum of 5 sent messages) with averaged sentiment scores represented in the colours of graph edges.

[3] https://developer.twitter.com/en/docs/tweets/search/api-reference/get-search-tweets.html

```
{
    "message": "Sirsnīgi sveicu Igauniju neatkarības
       pasludināšanas 100.gadadienā un novēlu turpināt
       iesākto ceļu, sasniedzot aizvien jaunas un augst
       ākas virsotnes! Lai miera, brīvības un labklājī
       bas pilna ir Igaunijas nākotne!#Estonia100 https
       ://t.co/Hz06yYgYOM",
    "id": 2741825,
    "tweetId": 967291062904553472.0,
    "createdAt": "2018-02-24T08:52:15",
    "language": "lv",
    "inReplyToStatusId": null,
    "inReplyToUserId": null,
    "inReplyToScreenName": null,
    "userId": 113691546.0,
    "userName": "Raimonds Vējonis",
    "userScreenName": "Vejonis",
    "countryCode": null,
    "placeName": null,
    "placeFullName": null,
    "placeType": null,
    "retweetedId": null,
    "sentiment": 0.578
  },
```

Figure 1. An example of a tweet entry in the Latvian Tweet Corpus

The corpus consists of a total amount of 3,867,444 tweets, out of which 2,882,670 are tweets in Latvian. The total number of unique tweets (excluding re-tweets and tweets with identical messages) is 1,191,730.

3. Sentiment Analysis Experiments

3.1. Data

In order to develop a sentiment analyser using supervised learning methods, we required a sentiment-annotated tweet corpus. For this purpose, we:

1. Annotated a sub-set of the LPTC using three sentiment categories (negative - -1, neutral - 0, and positive - 1). The annotation was carried out during the initial phase of the corpus collection (from August 2016 till November 2016), therefore it is rather dis-balanced with respect to the language and topics covered in the remaining corpus. However, the result of the manual annotation was a set of 6,778 annotated tweets. This dataset is the "Gold" dataset.
2. Performed automatic annotation of tweets containing emotionally (mostly) unambiguous emoticons (for examples, refer to Figure 3. This step produced a

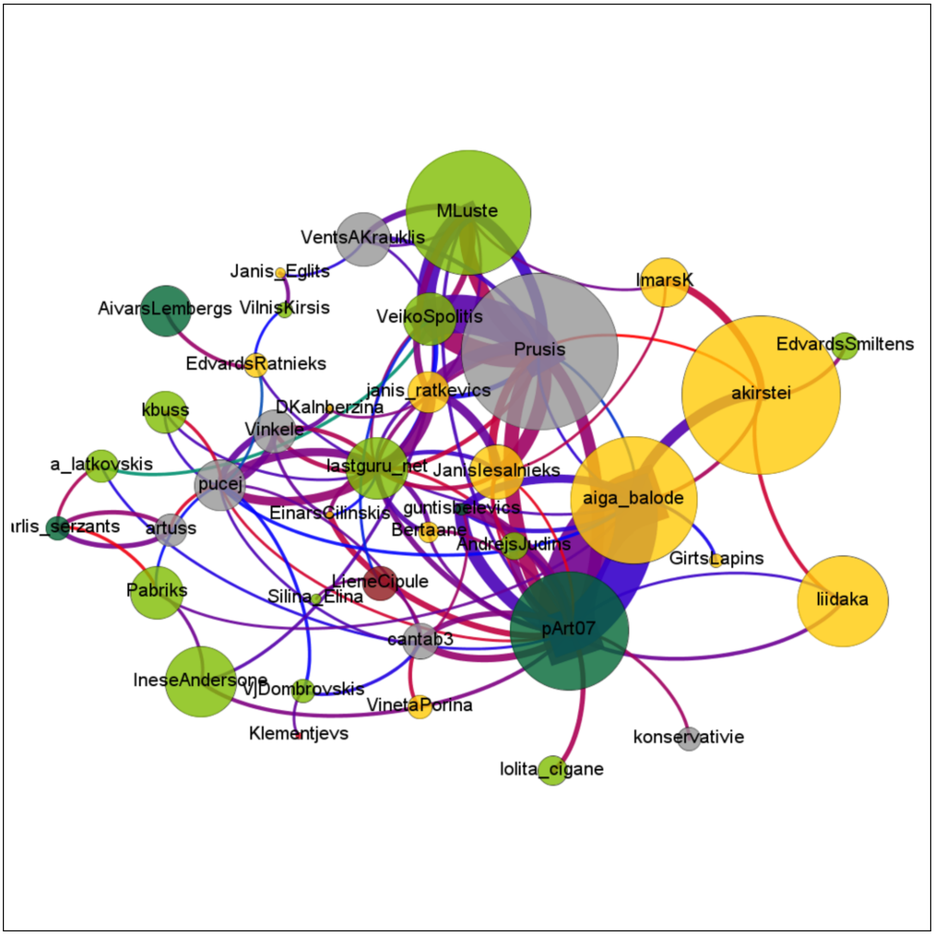

Figure 2. Example of directed (clockwise) communication between politicians (the colours of the arrows represent the average sentiment from negative (red) to positive (green)

dataset of 23,685 annotated tweets. Note that this is a potentially noisy dataset! This dataset is the "Auto (with ☺)" dataset. In order to test whether such automatic annotation does not negatively impact machine learning-based methods (by making the models learn to classify based on emoticon presence only), we created two other datasets based on this dataset - the "Auto (no ☺)" dataset has emoticons removed and the "Auto (both)" dataset combines both automatically acquired datasets.

3. Reused the corpus of 1,178 annotated tweets created by Peisenieks and Skadiņš [3]. This dataset is the "Peisenieks" dataset.
4. Used a machine translation system [8] to translate English sentiment-annotated datasets into Latvian. The datasets were: 1) the SemEval shared task data from 2013[4] and 2014[5], 2) data from a *kaggle* competition on movie review sentiment

[4] https://www.cs.york.ac.uk/semeval-2013/task2.html

[5] http://alt.qcri.org/semeval2014/task9/

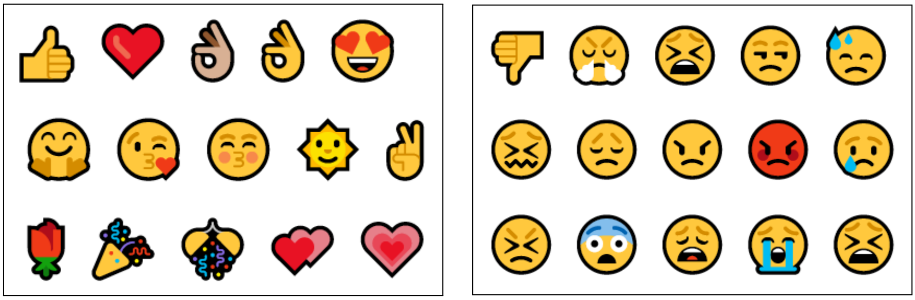

Figure 3. Examples of (a non-exhaustive) list of positive (left) and negative (right) emoticons used for automatic tweet annotation

analysis[6], and 3) data from a movie review data set[7]. A total of 45,531 annotated sentences were acquired using this method. Note that important information may be lost when translating from one language into another language using a machine translation system, therefore this dataset is also expected to be noisy. This dataset is the "English" dataset.

5. Annotated a time-balanced evaluation dataset of 1000 tweets from the LPTC. This corpus will be used to evaluate the sentiment analysis methods described in Section 3.2.

In addition to the sentiment-annotated datasets, we used the Latvian stop-word list from the ACCURAT Toolkit [9] and a revised version of the positive and negative word lists that were originally created by Pumpurs [10].

3.2. Experiments

Sentiment analysis experiments were performed using four machine learning algorithms: 1) an averaged perceptron classifier implemented by the author to utilise also positive and negative word features, 2) FastText, a text classification method that efficiently trains and uses skip-gram [11] word embeddings for text classification, 3) StarSpace, a recent text classification method introduced by Wu et al. [12] that, similarly to FastText is based on word embedding methods, and 4) an RNN based method implemented by Chen [13] that uses uni-directional and bi-directional long short-term memory units (LSTM and BiLSTM) for sentiment classification.

In the experiments, we analyse how to pre-process data (whether to use stop-words or not), whether upsampling allows achieving higher quality, whether word embeddings (for the second and third algorithms) are important, whether positive/negative word lists (for the first algorithm) are important, and finally, which of the methods allows for achieving the highest results.

3.3. Results

The results of the perceptron classifier (in terms of classification accuracy) are given in Table 1. The results show that the best results can be achieved when using positive/neg-

[6] https://www.kaggle.com/c/sentiment-analysis-on-movie-reviews/data

[7] http://ai.stanford.edu/ amaas/data/sentiment/

Table 1. Results of the averaged perceptron classifier (accuracy scores)

	No upsampling				**Upsampling**			
	Pos/neg list		**No pos/neg list**		**Pos/neg list**		**No pos/neg list**	
	Stopw.	**No stopw.**	**Stopw.**	**No stopw.**	**Stopw.**	**No stopw.**	**Stopw.**	**No stopw.**
Perceptron classifier without stemming								
Gold	0.651	0.650	0.611	0.610	0.649	0.653	0.612	0.594
Gold+Peisenieks	0.657	0.660	0.608	0.591	0.656	0.657	0.605	0.597
Gold+Auto (with ☺)	0.621	0.615	0.563	0.536	0.625	0.614	0.561	0.550
Gold+Auto (no ☺)	0.486	0.471	0.475	0.457	0.479	0.472	0.475	0.462
Gold+Auto (both)	0.481	0.471	0.471	0.468	0.485	0.477	0.471	0.449
Gold+English	0.614	0.591	0.584	0.565	0.606	0.589	0.602	0.574
Perceptron classifier with stemming								
Gold	0.661	0.662	0.609	0.621	0.657	0.652	0.612	0.615
Gold+Peisenieks	**0.676**	0.662	0.622	0.621	0.675	0.655	0.613	0.608
Gold+Auto (with ☺)	0.624	0.628	0.595	0.587	0.634	0.627	0.583	0.585
Gold+Auto (no ☺)	0.512	0.480	0.493	0.486	0.492	0.500	0.482	0.490
Gold+Auto (both)	0.487	0.493	0.481	0.460	0.490	0.500	0.482	0.498
Gold+English	0.613	0.596	0.592	0.580	0.610	0.593	0.603	0.593

ative word list-based features. There is an average accuracy increase (compared to scenarios without these features) of relative 5.5%. In terms of pre-processing, stemming of words allows to achieve higher results (relative 2.6% higher compared to the scenarios without stemming), however, removal of stop-words did not (in general) yield better results - the accuracy after stop-word removal dropped by average 1.4%. Training data upsampling for less frequent categories had almost no effect on the results (there is an average quality improvement of just 0.1%).

The results of the FastText and StarSpace experiments are given in Table 2. It is evident that the skip-gram word embeddings from FastText allow improving results (by an average of 7.1% over the scenarios without embeddings) and the FastText classifier achieves higher quality than the StarSpace classifier. We experimented also with upsampling, but it had no effect on the results[8]. The removal of stop-words showed to degrade classification accuracy for the StarSpace classifier by 24.8%, however, it had only a small negative effect of 2% for the FastText classifier. The best result was achieved using the FastText classifier with skip-gram embeddings and stop-word removal on the *Gold* dataset.

The results of the LSTM and BiLSTM classifiers are provided in Table 3. It is evident that the results are much lower than previous (both perceptron and FastText model results) results (by even up to absolute 10 accuracy points) and there is minimal difference between the classification accuracy of LSTM and BiLSTM models.

The implementation by Chen [13] did not support word embeddings, which for the FastText classifier allowed to improve the classification accuracy. Therefore, we investigated also the attention-based LSTM implementation by Baziotis et al. [14], which achieved the best results at the 2017 SemEval shared task on sentiment analysis and supports word embeddings. We trained two models using the *Gold+Peisenieks* and

[8] The results were, therefore, not included in the paper.

Table 2. Results of the FastText and StarSpace classifiers (accuracy scores)

	Without Embeddings		**With skip-gram Embeddings**	
	Stopw.	**No stopw.**	**Stopw.**	**No stopw.**
FastText skip-gram model-based classifier				
Gold	0.605	0.596	0.651	**0.654**
Gold+Peisenieks	0.597	0.593	0.626	0.617
Gold+Auto (with ☺)	0.572	0.513	0.582	0.546
Gold+Auto (no ☺)	0.445	0.457	0.457	0.465
Gold+Auto (both)	0.400	0.408	0.469	0.462
Gold+English	0.571	0.537	0.611	0.587
StarSpace models				
Gold	0.571	0.383		
Gold+Peisenieks	**0.581**	0.340		
Gold+Auto (with ☺)	0.511	0.436		
Gold+Auto (no☺)	0.460	0.390		
Gold+Auto (both)	0.472	0.370		
Gold+English	0.550	0.374		

Gold+Auto (with ☺) datasets. Unfortunately, the results showed that the authors' implementation achieves accuracies of only 42.6% and 41.0% respectively.

In addition to the machine learning-based classification methods, we performed also two experiments with a lexical classification method that assigns a classification score based on whether there are more positive keywords than negative keywords present in a message. If there are equal numbers of positive and negative keywords present, the class of "neutral" is assigned. The lexical classifier achieves an accuracy of 52.9% without stemming and 45.4% with stemming.

The best results were achieved by the perceptron classifier when trained on both the manually annotated dataset from the LPTC and the dataset created by Peisenieks and Skadiņš. This may indicate that the noise introduced by the automatic processing in the other datasets is too high to train better models.

4. Conclusion

The paper described the Latvian Tweet Corpus and its application in sentiment analysis. However, the corpus can be useful for many more different tasks, such as communication behaviour analysis, question-answering, and many more application areas.

The sentiment analysis experiments showed that the best overall results were achieved by the perceptron classifier achieving an accuracy of 67.6% on the evaluation dataset. At the same time, the more complex neural network-based classification methods performed worse, however, better than the lexical classification-based method.

The datasets used in these experiments as well as further information on the experiments can be found online at https://github.com/pmarcis/latvian-tweet-corpus.

Table 3. Results of the LSTM and BiLSTM classifiers (accuracy scores)

	Without Upsampling		**With Upsampling**	
	Stopw.	**No stopw.**	**Stopw.**	**No stopw.**
LSTM classifier				
Gold	0.521	0.550	0.542	**0.556**
Gold+Peisenieks	0.459	0.495	0.518	0.548
Gold+Auto (with ☺)	0.497	0.530	0.504	0.452
Gold+Auto (no ☺)	0.436	0.422	0.411	0.445
Gold+Auto (both)	0.448	0.418	0.425	0.435
Gold+English	0.448	0.424	0.470	0.428
BiLSTM classifier				
Gold	**0.575**	0.531	0.486	0.529
Gold+Peisenieks	0.493	0.504	0.538	0.534
Gold+Auto (with ☺)	0.465	0.46	0.468	0.459
Gold+Auto (no ☺)	0.459	0.455	0.438	0.446
Gold+Auto (both)	0.448	0.417	0.434	0.415
Gold+English	0.451	0.445	0.447	0.415

References

[1] G. Špats and I. Birzniece, Opinion Mining in Latvian Text Using Semantic Polarity Analysis and Machine Learning Approach, *Complex Systems Informatics and Modeling Quarterly* (2016), 51–59.

[2] G. Garkāje, E. Zilgalve and R. Darģis, Normalization and Automatized Sentiment Analysis of Contemporary Online Latvian, in: *Human Language Technologies-The Baltic Perspective: Proceedings of the Sixth International Conference Baltic HLT 2014*, Vol. 268, IOS Press, 2014, p. 83.

[3] J. Peisenieks and R. Skadiņš, Uses of Machine Translation in the Sentiment Analysis of Tweets, in: *Proceedings of the Sixth International Conference Baltic HLT 2014*, 2014.

[4] S. Rosenthal, N. Farra and P. Nakov, SemEval-2017 task 4: Sentiment analysis in Twitter, in: *Proceedings of SemEval-2017*, 2017, pp. 502–518.

[5] P. Nakov, A. Ritter, S. Rosenthal, F. Sebastiani and V. Stoyanov, SemEval-2016 task 4: Sentiment analysis in Twitter, in: *Proceedings of SemEval-2016*, 2016, pp. 1–18.

[6] S. Rosenthal, P. Nakov, S. Kiritchenko, S. Mohammad, A. Ritter and V. Stoyanov, Semeval-2015 task 10: Sentiment analysis in twitter, in: *Proceedings of SemEval 2015*, 2015, pp. 451–463.

[7] J.P. Pestian, P. Matykiewicz, M. Linn-Gust, B. South, O. Uzuner, J. Wiebe, K.B. Cohen, J. Hurdle and C. Brew, Sentiment analysis of suicide notes: A shared task, *Biomedical informatics insights* **5** (2012).

[8] M. Pinnis, R. Krišlauks, T. Miks, D. Deksne and V. Šics, Tilde's Machine Translation Systems for WMT 2017, in: *Proceedings of WMT 2017: Shared Task Papers*, Copenhagen, Denmark, 2017, pp. 374–381.

[9] M. Pinnis, R. Ion, D. tefnescu, F. Su, I. Skadia, A. Vasijevs and B. Babych, ACCURAT Toolkit for Multi-Level Alignment and Information Extraction from Comparable Corpora, in: *Proceedings of the ACL 2012 System Demonstrations*, Association for Computational Linguistics, 2012, pp. 91–96.

[10] A. Pumpurs, Lexicon of Positive and Negative Sentiment Words in Latvian, 2014. https://github.com/pumpurs/SentimentWordsLV.

[11] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado and J. Dean, Distributed Representations of Words and Phrases and their Compositionality, in: *Advances in Neural Information Processing Systems 26*, C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani and K.Q. Weinberger, eds, Curran Associates, Inc., 2013, pp. 3111–3119.

[12] L. Wu, A. Fisch, S. Chopra, K. Adams, A. Bordes and J. Weston, StarSpace: Embed All The Things!, *arXiv preprint arXiv:1709.03856* (2017).

[13] X. Chen, Sentiment Analysis implemented using Gluon and MXNet, 2018. https://github.com/chen0040/mxnet-sentiment-analysis.

[14] C. Baziotis, N. Pelekis and C. Doulkeridis, DataStories at SemEval-2017 Task 4: Deep LSTM with Attention for Message-level and Topic-based Sentiment Analysis, in: *Proceedings of SemEval-2017*, Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 747–754.