Deep Learning vs. Classic Models on a New Uzbek Sentiment Analysis Dataset

Elmurod Kuriyozov[*], **Sanatbek Matlatipov**[†], **Miguel A. Alonso**[*], **Carlos Gómez-Rodríguez**[*]

[*]Universidade da Coruña, CITIC
Grupo LYS, Departamento de Computación. Facultade de Informática, Campus de Elviña, A Coruña 15071, Spain
{e.kuriyozov, miguel.alonso, carlos.gomez}@udc.es

[†]National University of Uzbekistan
University Str. 4, 100174, Tashkent, Uzbekistan
mr.sanatbek@gmail.com

Abstract

Making natural language processing technologies available for low-resource languages is an important goal to improve the access to technology in their communities of speakers. To our knowledge, there are no well-established linguistic resources for the development of sentiment analysis applications for the Uzbek language. In this paper, we fill that gap by providing its first annotated corpora for polarity classification. Our methodology considers collecting a medium-size manually annotated dataset and a larger-size dataset automatically translated from existing resources. Then, we use these datasets to train what, to our knowledge, are the first sentiment analysis models on the Uzbek language, using both traditional machine learning techniques and recent deep learning models. Both sets of techniques achieve similar accuracy (the best model on the manually annotated test set is a convolutional neural network with 88.89% accuracy, and on the translated set, a logistic regression with 89.56% accuracy); with the accuracy of the deep learning models being limited by the quality of available pre-trained word embeddings.

1. Introduction

The advancement of technologies in the field of Natural Language Processing (NLP) over the past few years has led to achieve very high accuracy results, allowing the creation of useful applications that play an important role in many areas now. In particular, the adoption of deep learning models has boosted accuracy figures across a wide range of NLP tasks. As a part of this trend, sentiment classification, a prominent example of the applications of NLP, has seen substantial gains in performance by using deep learning approaches compared to its predecessor approaches (Barnes et al., 2017). However, low-resource languages still lack access to those performance improvements. Neural network models, which have gained wide popularity in recent years, are generally considered as the best supervised sentiment classification technique for resource-rich languages so far (Socher et al., 2013; Barnes et al., 2017; Zhang et al., 2018), but they require significant amounts of annotated training data to work well.

Meanwhile, the fact that a language can be considered a low-resource language does not necessarily mean that it is spoken by a small community. For instance, the language we focus on in this paper is Uzbek, which is spoken by more than 33 million native speakers in Uzbekistan as well as elsewhere in Central Asia and a part of China.[1]

It is also important to point out that NLP tools in general, and sentiment analysis tools in particular, benefit from taking into account the particularities of the language under consideration (Jang and Shin, 2010; Vilares et al., 2015). Uzbek is a Turkic language that is the first official and only declared national language of Uzbekistan. The language of Uzbeks (in native language: *O'zbek tili* or *O'zbekcha*) is a null-subject, agglutinative language and has many dialects, varying widely from region to region, which introduces more difficult problems to tackle.[2]

The main contributions of this paper are:

1. The creation of the first annotated dataset for sentiment analysis in Uzbek language, obtained from reviews of the top 100 Google Play Store applications used in Uzbekistan. This manually annotated dataset contains 2500 positive and 1800 negative reviews. Furthermore, we have also built a larger dataset by automatically translating (using Google Translate API) an existing English dataset[3] of application reviews. The translated dataset has ≈10K positive and ≈10K negative app reviews, after manually eliminating the major machine translation errors by either correcting or removing them completely.

2. The definition of the baselines for sentiment analyses in Uzbek by considering both traditional machine learning methods as well as recent deep learning techniques fed with fastText pre-trained word embeddings.[4] Although all the tested models are relatively accurate and differences between models are small, the neural network models tested do not manage to substantially outperform traditional models. We believe that the quality of currently available pre-trained

This work has received funding from the ANSWER-ASAP project (TIN2017-85160-C2-1-R) from MINECO, and from Xunta de Galicia (ED431B 2017/01 and Oportunius program).

[1] https://en.wikipedia.org/wiki/Uzbek_language

[2] Little information about Uzbek lanlzguages is available in English. A good starting point for readers who are interested could be: http://aboutworldlanguages.com/uzbek

[3] https://github.com/amitt001/Android-App-Reviews-Dataset

[4] https://fasttext.cc

word embeddings for Uzbek is not enough to let deep learning models perform at their full potential.

3. The definition of the steps for translating an available dataset automatically to a low-resource language, analysing the quality loss in the case of English-Uzbek translation.

All the resources, including the datasets, the list of top 100 apps whose reviews were collected, the source code used to collect the reviews and the one for baseline classifiers, are publicly available at the project's repository.[5]

The remainder of this paper is organized as follows: after this Introduction, Sect. 2. describes related work that has been done so far. It is followed by a description of the methodology in Sect. 3. and continues with Sect. 4. which focuses on Experiments and Results. The final Sect. 5. concludes the paper and highlights the future work.

2. Related Work

We only know of one existing sentiment analysis resource for the Uzbek language: a multilingual collection of sentiment lexicons presented in (Chen and Skiena, 2014) that includes Uzbek, but the Uzbek lexicon is very small and is not evaluated on an actual sentiment analysis system or dataset. To our knowledge, there are no existing annotated corpora on which it could be evaluated.

Other languages of the Turkic family such as Turkish and Kazakh have made considerable progress in the field. For example, a system for unsupervised sentiment analysis on Turkish texts is presented in (Vural et al., 2012), based on a customization of SentiStrength (Thelwall and Paltoglou) by translating its polarity lexicon to Turkish, obtaining a 76% accuracy in classifying Turkish movie reviews as positive or negative.

Sentiment analysis of Turkish political news in online media was studied in (Kaya et al., 2012) using four different classifiers (Naive Bayes, Maximum Entropy, SVM, and character-based n-gram language models) with a variety of text features (frequency of polar word unigrams, bigrams, root words, adjectives and effective polar words) concluding that the Maximum Entropy and the n-gram models are more effective when compared to SVM and Naive Bayes, reporting an accuracy of 76% for binary classification.

A sentiment analysis system for Turkish that gets a 79.06% accuracy in binary sentiment classification of movie reviews is described in (Dehkharghani et al., 2017), but it needs several linguistic resources and tools, such as a dependency parser and a WordNet annotated with sentiment information, which are not available for Uzbek.

(Yergesh et al., 2017) presented a rule-based sentiment analysis system for Kazakh working on a dictionary, morphological rules and an ontological model, achieving 83% binary classification accuracy for simple sentences.

A modern Deep Learning approach for solving Kazakh and Russian-language Sentiment Analysis tasks was investigated in (Sakenovich and Zharmagambetov, 2016). Particularly, Long Short-Term Memory (LSTM) was used to handle long-distance dependencies, and word embeddings (word2vec, GloVe) were used as the main feature.

3. Methodology

3.1. Data Collection

When it comes to choosing an available source to collect data for low-resource languages, the usual approach for resource-rich languages, such as Twitter data (Zimbra et al., 2018) or movie reviews (Chakraborty et al., 2018), may not qualify and end up being very scarce or not sufficient to work with. So one has to find out what is the most widespread web service from which a large amount of open data can be collected for a specific low-resource language. In the case of Uzbek, most of its speakers use mobile devices for accessing the Internet, and Android retains a share of more than 85% of the mobile Operating Systems market (as of February 2019)[6]. This is the reason why the reviews of Google Play Store Applications have been chosen as the data source for our research.

We selected the list of top 100 applications used in Uzbekistan, retrieving for each review its text and its associated star rating (from 1 to 5 stars). In order to promote future research on the Uzbek language, the project repository that has been created to share the sources of this paper contains a file with the list of URLs for those apps and the Python script for crawling the Play Store reviews. Due to Google's anti-spam and anti-DDOS policies, there are certain limitations on harvesting data, such as that only the most relevant 40 reviews can be obtained in a single request and up to 4500 in several requests (the corresponding source code has also been included).

3.2. Pre-processing

We observed that the collection of texts (together with the associated star ratings) downloaded by the above procedure was noisy, so we performed a correction process. The comments containing only emojis, names or any other irrelevant content, such as username mentions, URLs or specific app names were removed. Those written in languages different from Uzbek (mostly in Russian and some in English) were manually translated. There is another small inconvenience, specific to dealing with Uzbek texts: although currently the official and most-used alphabet for the language is the Latin one, some people still tend to write in the Cyrillic alphabet, which was the official alphabet decades ago and is still used in practice (Dietrich, 2018). Those Cyrillic comments were collected and transformed to the Latin one using an available online tool.[7] A small example is shown in Figure 1.

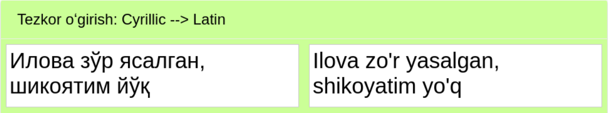
3.3. Annotation

This paper is intended to present only a binary classified dataset, so the main task was to label the reviews as positive or negative. A neutral class was not considered for the sake of simplicity since this is, to our knowledge, the

[5] https://github.com/elmurod1202/uzbek-sentiment-analysis

[6] http://gs.statcounter.com/os-market-share/mobile/uzbekistan

[7] Online Cyrillic to Latin transformation tool: https://savodxon.uz



Tezkor o'girish: Cyrillic --> Latin

Илова зўр ясалган, шикоятим йўқ	Ilova zo'r yasalgan, shikoyatim yo'q

Figure 1: Savodxon.uz: Online Cyrillic to Latin alphabet transformation tool, specifically for Uzbek language. (The example text in English: ”The app is nicely created, I have no complaints”)

first sentiment analysis dataset for our chosen language, so we preferred to start from the simplest setting. The annotation process was done by two native Uzbek speakers manually labeling the reviews, giving them a score of either 0 or 1, meaning that the review is either negative or positive, respectively. A third score was obtained from the dataset's rating column as follows:

- Reviews with 4- and 5-star ratings were labeled as positive (1);
- Reviews with 1- and 2-star ratings were labeled as negative (0);
- The majority of reviews with 3-star rating also turned out to have negative opinion so we labeled them as negative (0) as well, but both annotators removed the objective reviews.

Finally, the review was given a polarity according to the majority label. This process resulted into 2500 reviews annotated as positive and 1800 as negative.

3.4. Translation

In order to further extend the resources to support sentiment analysis, another larger dataset was obtained through machine translation. An available English dataset of positive and negative reviews of Android apps, containing 10000 reviews of each class, was automatically translated using MTRANSLATE[8]: an unofficial Google Translate API from English to Uzbek. The next step was to determine whether the translation was accurate enough to work with. Thus, we manually went through the translation results quickly and examined a random subset of the reviews, large enough to make a reasonable decision on overall accuracy. Although the translation was not clear enough to use for daily purposes, the meaning of the sentences was approximately preserved, and in particular, the sentiment polarity was kept (except for very few exceptional cases). An example of the translation can be seen in Figure 2.

As a result, we have obtained two datasets with the sizes shown in Table 1. While the translated dataset is quite balanced, the manually annotated dataset has about 3:4 ratio of negative to positive reviews. Each of the datasets has been split into a training and a test set following a 90:10 ratio, for the experiments in the next section.

4. Experiments & Results

To create the baseline models for Uzbek sentiment analysis, we chose various classifiers from different families, including different methods of Logistic Regression (LR), Support Vector Machines (SVM), and recent Deep Learning methods, such as Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN).

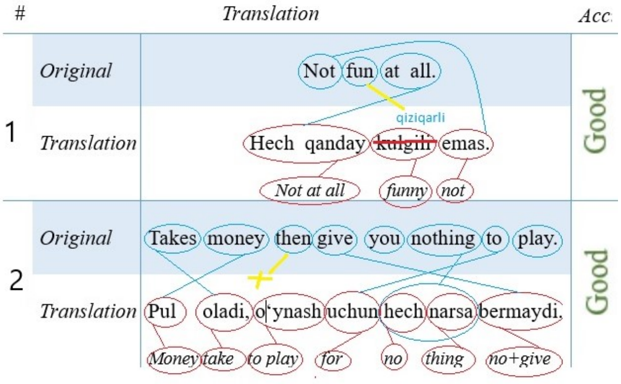


Figure 2: An example of the translation process on two random negative reviews. As can be observed, the polarity of the comments is preserved.

Datasets	Positive	Negative	Total
Manual Dataset	2500	1800	4300
Translated Dataset	9632	8853	18485

Table 1: Number of reviews per dataset and polarity class

We implemented LR and SVM models by means of the Scikit-Learn (Pedregosa et al., 2011) machine learning library in Python with default configuration parameters. For the LR models, we implemented a variant based on word n-grams (unigrams and bigrams), and one with character n-grams (with n ranging from 1 to 4). We also tested a model combining said word and character n-gram features.

In the case of Deep Learning models, we used Keras (Chollet et al., 2015) on top of TensorFlow (Abadi et al., 2015). We use as input the FastText pre-trained word embeddings of size 300 (Grave et al., 2018) for Uzbek language, that were created from Wiki pages and CommonCrawl, [9] which, to our knowledge, are the only available pre-trained word embeddings for Uzbek language so far. The source code for all the chosen baseline models is available on the project's GitHub repository.

For the CNN model, we used a multi-channel CNN with 256 filters and three parallel channels with kernel sizes of 2,3 and 5, and dropout of 0.3. The output of the hidden layer is the concatenation of the max pooling of the three channels. For RNN, we use a bidirectional network of 100 GRUs. The output of the hidden layer is the concatenation of the average and max pooling of the hidden states. For the combination of deep learning models, we stacked the CNN on top of the GRU. In the three cases, the final output is obtained through a sigmoid activation function applied on the previous layer. In all cases, Adam optimization algorithm, an extension of stochastic gradient descent, was chosen for training, with standard parameters: learning rate $\alpha = 0.0001$ and exponential decay rates $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Binary cross-entropy was used

[8] https://github.com/mouuff/mtranslate

[9] http://commoncrawl.org

Methods used	**ManualTT**	**TransTT**	**TTMT**
Support-vector Machines based on linear kernel model	0.8002	0.8588	0.7756
Logistic Regression model based on word ngrams	0.8547	0.8810	0.7720
Recurrent + Convolutional neural network	0.8653	0.8864	0.7850
Recurrent Neural Network with fastText pre-trained word embeddings	0.8782	0.8832	0.7996
Logistic Regression model based on word and character ngram	0.8846	**0.8956**	**0.8145**
Recurrent Neural Network without pre-trained embeddings	0.8868	0.8832	0.8052
Logistic Regression model based on character ngrams	0.8868	0.8945	0.8021
Convolutional Neural Network (Multichannel)	**0.8888**	0.8832	0.8120

Table 2: Accuracy results with different training and test sets. **ManualTT** - Manually annotated Training and Test sets. **TransTT** - Translated Training and Test sets. **TTMT** - Translated dataset for Training, Annotated dataset for Test set.

as loss function.

As our performance metric, we use classification accuracy. This is the most intuitive performance measure for a binary classifier, and it is merely a ratio of correctly predicted observations to total observations (Powers, 2011).

$$\text{accuracy} = \frac{\sum \text{true positive} + \sum \text{true negative}}{\sum \text{total population}}$$

Since we have worked on relatively small dataset, other metrics, such as the runtime complexity and memory allocations were not taken into account.

Table 2 shows the accuracy obtained in three different configurations: a first one working on the manually annotated dataset (ManualTT), a second one on the translated dataset (TransTT) and a third one in which training was performed on translated dataset while testing was performed on the manually annotated dataset.

The LR based on word n-grams obtained a binary classification accuracy of 88.1% on the translated dataset, while the one based on character n-grams, with its better handling of misspelled words, improved it to 89.45%. To take advantage of both methods, we combined the two and got 89.56% accuracy, the best performance for the translated dataset obtained in this paper. The deep learning models have shown accuracies ranging from 86.53% (using RNN+CNN) to 88.88% (using Multichannel CNN) on our manually annotated dataset, the latter being the best result on this dataset, while the RNN+CNN combination performed well on the translated dataset with 88.64% average accuracy, slightly better than others (88.32% for single RNN and CNN models).

Table 3 shows per-class metrics of our best result on the translated dataset, obtained from the LR model based on word and character n-grams trained on that same dataset.

Classes	**Precision**	**Recall**	**F1-score**
Negative	0.89	0.91	0.90
Positive	0.90	0.88	0.89

Table 3: Performance metrics of the best result on the translated dataset.

Although the results obtained have been good in general terms, those obtained for deep learning models have not clearly surpassed the results obtained by other classifiers. This is mainly due to some of the complexities of Uzbek language. Indeed, Uzbek morphology (Matlatipov and Vetulani, 2009) is highly agglutinative, and this aspect makes it harder to rely on word embeddings: a single word can have more than 200 forms generated by adding suffixes, sometimes even an entire sentence in English language can be described by one word. . An example of how agglutinative the language is is shown in Figure 3.

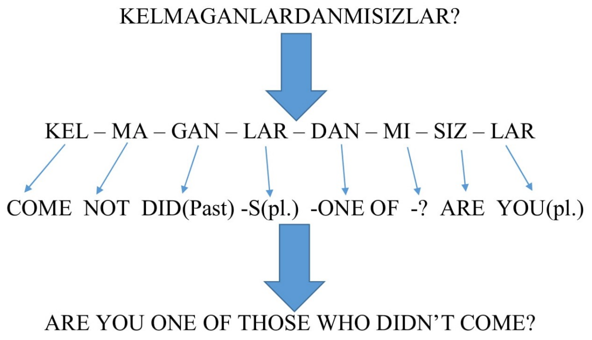

KELMAGANLARDANMISIZLAR?
KEL – MA – GAN – LAR – DAN – MI – SIZ – LAR
COME
NOT
DID(Past)
-S(pl.)
-ONE OF
-?
ARE
YOU(pl.)
ARE YOU ONE OF THOSE WHO DIDN'T COME?

Figure 3: An example of the agglutinative aspect of Uzbek language. Here we describe how just one Uzbek word can correspond to an entire sentence in English.

This agglutinative nature of Uzbek poses a major challenge for the definition of word embeddings. In our experiments, we could not associate a word-embedding to about 37% of words occurring in reviews. The reason for that was the noise of the reviews dataset we used, and which contained a large amount of misspelled words. Additionally, while our dataset contains only words in Latin alphabet, about the half of the word embeddings we used were in Cyrillic, decreasing the chance of the word to be found.

5. Conclusion and Future work

In this paper, we have presented a new Sentiment Analysis dataset for Uzbek language, collected from the reviews of Top 100 Android applications in Uzbekistan in Google Play Store. This dataset contains 4300 negative and positive reviews with a 3:4 ratio between the respective classes. It was manually annotated by two annotators, also considering the star rating provided by the reviewers. We also presented another new and relatively larger (20K) dataset of the same type, but this time it was automatically translated to Uzbek using Google Translate from an existing app review dataset in English language.

From the results of the experiments presented here, one can conclude that deep learning models do not perform better in sentiment classification than classic models for a low-resource language. We achieved our best accuracy (89.56%) on the translated dataset using a logistic regression model using word and character n-grams. The modern deep learning approaches have shown very similar results, without substantially outperforming classic ones in accuracy as they tend to do when used for resource-rich languages. We believe this to be due to lack of resources to feed the deep learning models: for example, the pre-trained word embeddings need to be enhanced (trained on a larger dataset) in order to benefit from the recent methods.

Our future work will be focused on creating more fundamental resources for the Uzbek language, such as tagged corpora, pre-trained word embeddings, lexicon and treebanks allowing us to build essential NLP tools, like part-of-speech taggers and parsers, which in turn can be used to improve sentiment analysis and other NLP tasks. An alternative to improve the deep learning models tested in this work would be to use character embeddings, which should be a good fit for an agglutinative language because they can capture information about parts of words and reduce the sparsity due to the large number of different words.

6. References

Abadi, Martín et al., 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Barnes, Jeremy, Roman Klinger, and Sabine Schulte im Walde, 2017. Assessing state-of-the-art sentiment models on state-of-the-art sentiment datasets. *arXiv preprint arXiv:1709.04219*.

Chakraborty, Koyel, Siddhartha Bhattacharyya, Rajib Bag, and Aboul Ella Hassanien, 2018. Comparative sentiment analysis on a set of movie reviews using deep learning approach. In *International Conference on Advanced Machine Learning Technologies and Applications*. Springer.

Chen, Yanqing and Steven Skiena, 2014. Building sentiment lexicons for all major languages. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Vol. 2: Short Papers)*. Baltimore, Maryland: Association for Computational Linguistics.

Chollet, François et al., 2015. Keras. `https://github.com/fchollet/keras`.

Dehkharghani, Rahim, Berrin Yanikoglu, Yucel Saygin, and Kemal Oflazer, 2017. Sentiment analysis in Turkish at different granularity levels. *Natural Language Engineering*, 23(4):535–559.

Dietrich, Ayse, 2018. Language policy and hegemony in the turkic republics. In Ernest Andrews (ed.), *Language Planning in the Post-Communist Era: The Struggles for Language Control in the New Order in Eastern Europe, Eurasia and China*. Cham: Springer International Publishing, pages 145–167.

Grave, Edouard, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov, 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Jang, Hayeon and Hyopil Shin, 2010. Language-specific sentiment analysis in morphologically rich languages. In *Coling 2010: Posters*. Beijing, China: Coling 2010 Organizing Committee.

Kaya, Mesut, Guven Fidan, and Ismail H. Toroslu, 2012. Sentiment analysis of Turkish political news. In *Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '12. Washington, DC, USA: IEEE Computer Society.

Matlatipov, Gayrat and Zygmunt Vetulani, 2009. Representation of Uzbek morphology in prolog. In *Aspects of Natural Language Processing. Lecture Notes in Computer Science*, volume 5070. Springer.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Powers, Ailab, David, 2011. Evaluation: From precision, recall and f-measure to roc, informedness, markedness & correlation. *J. Mach. Learn. Technol*, 2:2229–3981.

Sakenovich, Narynov Sergazy and Arman Serikuly Zharmagambetov, 2016. On one approach of solving sentiment analysis task for Kazakh and Russian languages using deep learning. In N. T. Nguyen, L. Iliadis, Y. Manolopoulos, and B. Trawiński (eds.), *Computational Collective Intelligence*. Cham: Springer.

Socher, Richard, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts, 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA.

Vilares, David, Miguel A. Alonso, and Carlos Gómez-Rodríguez, 2015. A syntactic approach for opinion mining on Spanish reviews. *Natural Language Engineering*, 21(01):139–163.

Vural, A. Gural, B. Barla Cambazoglu, Pinar Senkul, and Z. Ozge Tokgoz, 2012. A framework for sentiment analysis in Turkish: Application to polarity detection of movie reviews in Turkish. In *Computer and Information Sciences III*. Springer London, pages 437–445.

Yergesh, Banu, Gulmira Bekmanova, Altynbek Sharipbay, and Manas Yergesh, 2017. Ontology-based sentiment analysis of Kazakh sentences. In *International Conference on Computational Science and Its Applications*. Springer.

Zhang, Lei, Shuai Wang, and Bing Liu, 2018. Deep learning for sentiment analysis: A survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, 8(4).

Zimbra, David, Ahmed Abbasi, Daniel Zeng, and Hsinchun Chen, 2018. The state-of-the-art in Twitter sentiment analysis: A review and benchmark evaluation. *ACM Transactions on Management Information Systems (TMIS)*, 9(2):5.