УДК 004.032.26
МРНТИ 06.81.23

KAZAKH NAMES GENERATOR USING DEEP LEARNING

NURMAMBETOV D., DAUYLOV S., BOGDANCHIKOV A.

Süleyman Demirel University

Abstract: *In recent years, sentiment analysis of e-mail messages or social media posts is becoming very popular. It can help people define if they are reading something positive or negative. On the same time, there are some services on the Internet that can help you find or create a new name. When processing the creation, they check the name in other popular languages, so your name does not mean inappropriate things in other languages. For this they bill for 25 thousand US dollars. If there are such services, then there is a demand. In this study, sentiment analysis of e-mails was implemented with using StanfordNLP [1] lemmatizer and classic machine learning algorithms as a classifier. It is applied to real e-mails from Russian speaking mailbox, which means there are both English and Russian messages. Thus, language identification is also added as preprocessing step. In this study only binary sentiment analysis was made, but it can be improved with adding several emotions to be detected. Then another model generates Kazakh names using neural networks, where all Kazakh names data has been collected through various websites. The sentiment analysis model gives 81% accuracy and the joint use of two models allow us to generate new Kazakh names, which are checked with Russian language if they mean something inappropriate. The result can be improved with checking with other languages.*

Key words: *Natural language processing, sentiment analysis, Deep Learning, Artificial Intelligence, Generate Names*

DEEP LEARNING ТЕХНОЛОГИЯСЫН ПАЙДАЛАНУ АРҚЫЛЫ ҚАЗАҚША АТАУЛАРЫНЫҢ ГЕНЕРАТОРЫ

Аңдатпа: *Соңғы жылдары электрондық поштаның хабарламаларын немесе әлеуметтік желілердегі хабарламаларды талдау өте қарқынды өсіп келеді. Бұл адамдарға жағымды немесе жағымсыз мәліметтерді оқып жатқандығын анықтауға көмектеседі. Сонымен қатар Интернетте жаңа атау табуға немесе жасауға көмектесетін бірнеше қызметтер бар. Шығарманы өңдеу кезінде олар басқа танымал тілдердегі атауды тексереді, сондықтан сіздің атыңыз басқа тілдердегі келеңсіздікті білдірмейді. Бұл үшін олар 25 мың АҚШ долларын талап етеді. Мұндай қызметтердің болуы, сұраныс тудырады. Осы зерттеуде StanfordNLP [1] лемматизаторы мен классикалық машиналарды оқыту алгоритмдерін классификатор ретінде қарастырып, электрондық поштаның пікірлеріне жүгіндік. Ол орыс тілінде сөйлейтін пошта жәшігіндегі нақты электрондық хаттарға қолданылады, яғни ағылшын және орыс тілдерінде де бар. Осыған орай тілдік сәйкестендіру, сондай-ақ алдын ала өңдеу қадамы ретінде қосылады. Зерттеу барысында тек бинарлық көңіл-күйге талдау жасалды, бірақ оны анықтауға бірнеше эмоциялар қосып жақсартуға болады. Содан кейін тағы бір модель нейрондық желілерді қолдана отырып, қазақша атауларды жасайды, мұнда барлық қазақ атаулары туралы мәліметтер әртүрлі веб-сайттар арқылы жиналады. Сезім талдауы моделі 81% дәлдік береді және екі модельдің бірігіп пайдаланылуы сәйкессіздік мағынаны білдірсе, онда орыс тілімен тексерілетін жаңа қазақша атауларды шығаруға мүмкіндік береді. Басқа тілдермен салыстыра тексергенде, нәтижесін жақсарта аламыз.*

Түйінді сөздер: *Тілдерді табиғи түрде өңдеу, сезім талдауы, тереңдетіп оқыту, жасанды интеллект, есімдер құру*

ГЕНЕРАТОР КАЗАХСКИХ ИМЕН С ИСПОЛЬЗОВАНИЕМ DEEP LEARNING

Аннотация: *В последние годы анализ настроений сообщений электронной почты или сообщений в социальных сетях становится очень популярным. Это может помочь людям определить, читают ли они что-то положительное или отрицательное. В то же время в Интернете есть несколько служб, которые могут помочь вам найти или создать новое имя. При обработке создания они проверяют имя на других популярных языках, поэтому ваше имя не означает неуместные вещи на других языках. За это они выставляют счет на 25 тысяч долларов США. Если есть такие услуги, то есть спрос. В этом исследовании был проведен анализ настроений электронной почты с использованием лемматизатора StanfordNLP [1] и классических алгоритмов машинного обучения в качестве классификатора. Он применяется к реальным электронным письмам из русскоязычного почтового ящика, что означает наличие как английских, так и русских сообщений. Таким образом, идентификация языка также добавляется в качестве шага предварительной обработки. В этом исследовании был проведен только анализ бинарных настроений, но его можно улучшить, добавив несколько обнаруживаемых эмоций. Затем другая модель генерирует казахские имена, используя нейронные сети, где все казахские имена были собраны через различные веб-сайты. Модель анализа настроений дает точность 81%, а совместное использование двух моделей позволяет нам генерировать новые казахские имена, которые проверяются на русском языке, если они означают что-то неуместное. Результат может быть улучшен путем проверки с другими языками.*

Ключевые слова: *обработка естественного языка, анализ настроений, глубокое обучение, искусственный интеллект, генерация имен*

Introduction

Sentiment analysis is also called polarity detection, when the objective is to define if the text is positive or negative. It comprises of data collection, language identification, part-of-speech (POS) tagging led to stemming or lemmatization and polarity or emotions detection steps. And for each of them there are a lot of different approaches and algorithms. Basically, creating a new approach in any of these steps is a contribution to Natural Language Processing (NLP). Or its application to a new area.

Mostly, NLP tasks are solved using deep learning methods. Deep learning is a subset of machine learning, which, in turn, is a subset of artificial intelligence (AI). Artificial intelligence is a method that allows a machine to simulate human behavior. Machine learning is a method of applying AI using algorithms, trained on data. And finally, deep learning is a type of machine learning based on the structure of the human brain in terms of deep learning on artificial neural networks. Nowadays, deep learning uses many areas, for example, with the support of customers, they begin to replace people, in practice, you don't even know who is talking to you in the medical care they use to get the most accurate result, because the computer is trained with a lot of data.

The aim is to create an approach in names generator, as well as build Kazakh names dataset by collecting from different resources like books, websites, forums and so on, which will contain only Kazakh names. The objectives are to make a research in terms of related works and methods in names generation, create a robust and effective algorithm based on sequence models and then check the names with sentiment analysis. Reviewing results of testing accuracy on sentiment and generation of names are also the part of results discussion.

The idea of creating a name generator arose from two things: an opportunity to make an

impact on development of NLP application on Kazakh language and the creation of the service based on deep learning algorithms to be described in this paper. Selecting or creating a new name is presented as a web-service, and there are some of them already available to generate names in Kazakh. From early childhood throughout our lives, we have not heard a single word as often as our own name. This can motivate people to use such services, as well as generating business names.

The objectives of this study were to create an approach in sentiment analysis for e-mail messages within working communication, which gives quite good results in terms of accuracy. And then test the trained model on the generated Kazakh names.

Dataset

Dataset for sentiment analysis part was parsed from outlook, messages were cropped due to delete previous conversation which can influence on algorithm. Also, all additional information as signature, sender name was removed from the dataset. It was only 100 mails, while getting more data could improve the accuracy of sentiment analysis.

Data for Kazakh names generation was collected from several websites [2-5]

Literature review

During the study, papers describing the approaches in sentiment analysis with different application were reviewed.

The first question to be answered was - what if a system making sentiment analysis already exists. There is no work related to sentiment analysis in Russian specifically, but in paper "SentiCorr: Multilingual Sentiment Analysis of Personal Correspondence" [6] a multilingual sentiment analysis system called SentiCorr was presented. It is based on four-stage approach - language identification using Graph-based approach for Language Identification (LIGA) algorithm for short texts, POS tagging using TreeTagger, subjectivity detection using AdaBoost and polarity detection using Rule-Based algorithm to create an Emissive Model on patterns. They also developed an Outlook plug-in allowing people to test the system. In the paper a classification was implemented into objective, negative and positive.

Also, the question of bipolarity or multi-polarity of messages is important too. In the paper above analysis was made on phrases level thus messages could be bipolar. It could be a good point to check in this study.

A paper "A psychological based analysis of marketing email subject lines" [7] applies a similar approach for creating 40 tips to help you get an ideal e-mail subject. It provided emotional analysis, subjectivity analysis and sentiment analysis of real e-mails and got insights from each e-mail's labels in terms of attention, impression or interest it caused.

A paper "Sentiment Analysis for Automated Email Response System" [8] used messages between students and teacher to provide a comparative analysis on sentiment analysis classifier. According to it, RNN achieved the most accuracy on the dataset, and was used in auto-response system. In our study, we also can provide a comparison of different algorithms to define the most suitable for the data.

There are not so much algorithms, which support POS-tagging for Russian language. StanfordNLP algorithm used in this study was also described in the paper "Universal Dependency Parsing from Scratch". It is recurrental neural network-based algorithm, which can lemmatize in more than 50 languages. Also, there is no sentiment analysis related paper using smileys as a part of sentences and parentheses were used in this study as an additional indicator of message sentiment.

There are no Kazakh name generator works, which describe the proposed algorithm and methods. One of the latest updated papers is "Generating Thematic Chinese Poetry using Conditional Variational Autoencoders with Hybrid Decoders" [2]. The presented methodologies of utilizing grouping-to-arrangement models with consideration frequently produce non-topical sonnets. They present a novel restrictive variational autoencoder with a half and half decoder adding

the deconvolutional neural systems to the general intermittent neural systems to completely learn theme data by means of inactive factors. This methodology fundamentally improves the importance of the created sonnets by speaking to each line of the sonnet in a setting touchy way as well as in a comprehensive manner that is exceptionally identified with the given watchword and the educated theme. A proposed enlarged word2vec model further improves the musicality and evenness. Tests show that the created sonnets by the methodology described in the paper are generally fulfilling with directed guidelines and reliable subjects, and 73.42% of them get an Overall score no under 3 (the most elevated score is 5). In this study we will use sentiment analysis to estimate the quality of generated text.

One of the companies in Ukraine called KOLORO - Brand Design [10] in their blog had written how their branding process works. They come to a decision that helping the name of your children is like branding, the name should mean something, somehow related to parents' past, cultural values, your wishes, sounds good, doesn't mean anything bad in other languages and even uses statistics with a forecast. They also write interesting articles like how to not name children. The whole process looks like creating a name, checking uniqueness, comparing it in any language and making sure that it's not profanity or insult at the end lawyers check that name excludes the possibility of coincidence with the registered trademark. Also, they mentioned that in Switzerland this service costs $28000.

It is shown that in the similar works authors present the model to generate text and evaluate it comparing with some benchmark and in this study the result of generation will be checked with sentiment analysis in Russian language, whether the name can mean something inappropriate.

Methods and materials

The pipeline used in this study for sentiment analysis is as following: to apply language identification first (to know which corpora to use) using langdetect library[11], removing punctuation using TextBlob [12] and stop-words using nltk [13] library, lemmatization using StanfordNLP algorithm[1], tokenization using CountVectorizer and TF-IDF as described here "Twitter Sentiment Analysis using NLTK, Python" [14] and applying LogisticRegression classifier. Parentheses with spaces prior were not considered as stop-words since it can be used as additional emotional indicator.

According to "Why do Russians use parentheses instead of smileys?" [15] in Russian written communication parentheses are commonly used for emotions expression, and detection of emotions from smileys is also an advantage of this study.

Sentiment analysis in this study consists of:

1) Tokenizing a message using TextBlob library [12]. It has words function, which tokenizes text into words with default tokenizer function. But it is also available to use any nltk tokenizers as custom tokenizer, which suits your needs. Thus, TweetTokenizer from nltk was used, since it defines parentheses as smileys, and it could help in analysis of russian text.

2) After forming a word sequence, we remove stop-words using standard corpus from nltk library. But as a preprocessing step Langdetect is used to define English messages, which will be processed with English stop-words, and for other e-mails we use Russian stop-words.

3) Lemmatization is realized using StanfordNLP library with English corpora applied to English messages, and Russian one for Russian messages.

4) Apply CountVectorizer to get a matrix representing all the words in the document. It is shown below in Fig. 1:

5) To count TF-IDF weights for each token, which will be used as an input vector for the classifier:

• Tf(d,t) (Term frequency) is defined as the number of occurrence of the term t in document d

• Idf(t) (Inverse document of frequency) is defined as log(D/t), where D: Total number of documents and t: Number of documents with the term.

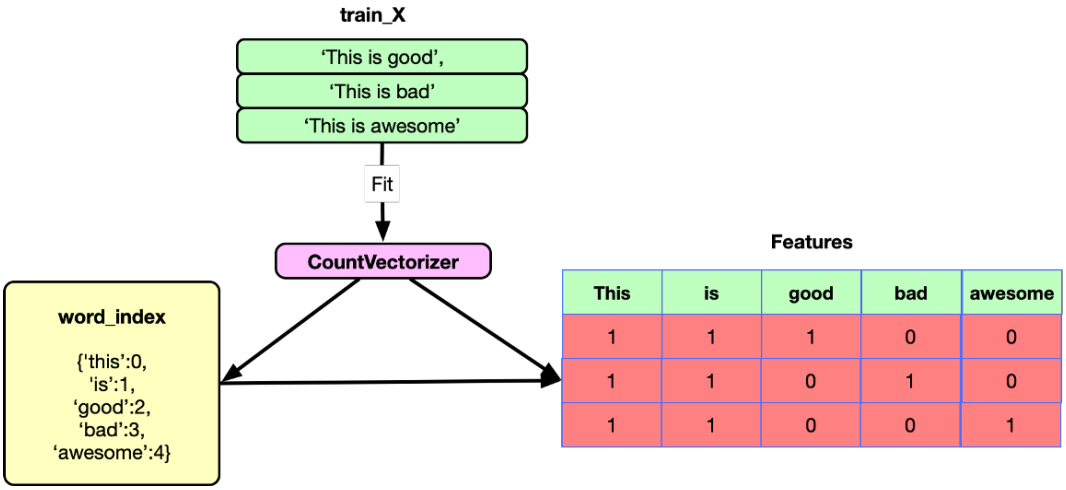

Figure 1. CountVectorizer sparse matrix demonstration

What Tf-Idf transformer does is returns the product of Tf and Idf which is the Tf-Idf weight of the term.

6) To apply StandardScaler [16], which standardizes features by removing the mean and scaling to unit variance. The standard score of a sample x is calculated as: z = (x - u)/s where u is the mean of the training samples and s is the standard deviation of the training samples.

7) To apply LogisticRegression as a classifier, since it is the most common algorithm for binary tasks.

The model for names generation is one-layer LSTM with preprocessing which includes norming the name and one-hot encoding with Kazakh alphabet. It is also added with dropout with given probability 0.1 to avoid overfitting. It includes the steps:

• Preparation – helper function to get random pairs of (category, line). The category here is just a one-hot vector, just like the vector input. For each timestep (that is, for each letter in a preparation word) the contributions of the system will be (class, current letter, shrouded state) and the yields will be (next letter, next concealed state). So, for each preparation set, we'll need the classification, a lot of info letters, and a lot of yield/target letters. Since we are foreseeing the following letter from the present letter for each timestep, the letter sets are gatherings of back to back letters from the line - for example for "ABCD<EOS>" we would make ("A", "B"), ("B", "C"), ("C", "D"), ("D", "EOS"). The classification tensor is a one-hot tensor of size <1 x n_categories>. When preparing we feed it to the system at each timestep - this is a plan decision, it could have been incorporated as a major aspect of starting a concealed state or some other procedure.

• Training the network - we are making a forecast at each progression, so we are computing loss at each progression. The advantages of autograd permits you to just entirely ignore these losses at each progression and make a back-propagation. Preparing is the same old thing - consider training a lot of times and hold up a couple of moments, printing the present time and loss each print_every iteration, and keeping a store of a normal loss for every plot_every iteration in all_losses for plotting later.

• Sampling the network - to test we give the system a letter and ask what the following one is, feed that in as the following letter, and rehash until the EOS token.

Data and results

First, we implemented a neural network for generating names in Kazakh. So, after running the learning process for 200000 iterations, it shows good results. Figure 1 shows the training loss converging to the value close to zero, which means that training was successful and finally loss (which is Cross Entropy Loss in this study) is low enough.

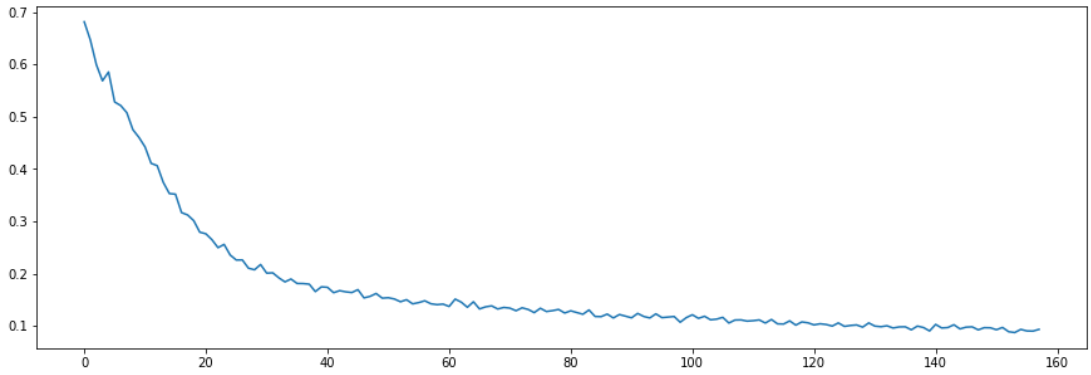

Figure 2. Training loss of Kazakh Names Neural Network

Then we generate some number of names. For example, as an input, we have entered the name in the Kazakh language, which generates pretty good names starting from each letter in the name (Figure 2). For the future, we can label each name by category for man or woman's name or it is unisex.

С – Саира
А – Ақын
Н – Нұртабаи
Ж – Жанар
А – Аиман
Р – Раира

Figure 3. Result of generated name based on input "САНЖАР"

The sentiment analysis model was trained on real outlook data and then it was tested and gave 81% accuracy achieved on validation set, which can be increased with using other non-linear classification algorithms. Also, it can give better result with more data, which must be manually labeled or adding some cross-validation for parameters. In Fig. 2 here are classification report [17], confusion matrix [18] and accuracy for validation set.

	precision	recall	f1-score	support
0	0.93	0.84	0.88	44
1	0.42	0.62	0.50	8
micro avg	0.81	0.81	0.81	52
macro avg	0.67	0.73	0.69	52
weighted avg	0.85	0.81	0.82	52

```
[[37  7]
 [ 3  5]]
0.8076923076923077
```

The validation set was also formed from e-mail messages. In comparison with other related works, it seems a similar result, since in [8] it was achieved from 81.5% to 87% accuracy depending on model for sentiment analysis with using 40 000 samples from open datasets in English. But using 10 000 it gives only approximately 50% accuracy.

And after adding our generated names to the sentences and defining its sentiments, we have a system, which is able to check generated names whether it means anything inappropriate.

Conclusion

We have created a system, which generates Kazakh names and we have implemented sentiment analysis of mostly Russian e-mails with using of StanfordNLP lemmatizer and Logistic Regression as a classifier. It is applied to real e-mails from the mailbox, and to the sentences including the generated names. In this study only binary sentiment analysis was made, but it can be improved with adding several emotions to be detected. It gave 81% on validation set but can be improved with adding more data. Also, custom tokenizer can be used for splitting text to words, with adding all specifications of Russian e-mail text or abbreviations specific for e-mail communication. The system can generate new name for you in Kazakh and then check if it means anything in Russian, which can have bad sentiment. In future, it can be added with checking within other languages, so that generated name will be appropriate to use.

REFERENCES

1. Peng Qi, Timothy Dozat, Yuhao Zhang and Christopher D. Manning. 2018. Universal Dependency Parsing from Scratch in Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, pp. 160-170
2. Нұргүл Абай. Балаға ең жиі қойылатын ТОП-20 есімнің мағынасы немесе ат қоярда нені ұмытпаған жөн. Sputniknews.kz. Nov 25, 2018. https://sputniknews.kz/society/20181013/7589294/bala-esim-top-20.html
3. Накипов Мұхамедәлі Асанұлы. Қазақша есімдердің тізімі. Bilim-All.kz. March 12, 2018. https://bilim-all.kz/esimder/all
4. Айнаш Ануарбек. Қазақша қыз есімдері мен олардың мағынасы. April 11, 2017. Yvision.kz. https://yvision.kz/post/763198
5. Stan.kz. Қазақы есімдер. Ұлыңызға қандай есім бердіңіз. Stan.kz. May 12, 2018. https://stan.kz/kazaky-esimder-ulynyzga-kanday-esim-b/
6. Erik Tromp; Mykola Pechenizkiy, "SentiCorr: Multilingual Sentiment Analysis of Personal Correspondence", 2011 IEEE 11th International Conference on Data Mining Workshops, 2011.
7. R. Miller; E.Y.A. Charles, "A psychological based analysis of marketing email subject lines", 2016 Sixteenth International Conference on Advances in ICT for Emerging Regions (ICTer), 2016.
8. Muhammad Babar Abbas; Mukarram Khan, "Sentiment Analysis for Automated Email Response System", 2019 International Conference on Communication Technologies (ComTech), 2019
9. Xiaopeng Yang, Xiaowen Lin, Shunda Suo, Ming Li. Generating Thematic Chinese Poetry using Conditional Variational Autoencoders with Hybrid Decoders. Arxiv Sanity Preserver. 5 Mar 2020. https://arxiv.org/abs/1711.07632v4
10. Анна Слёз. Как выбрать имя ребенку. Koloro brand Design Blog. Dec 4, 2019. https://koloro.ua/blog/brending-i-marketing/sozdanie-imeni-rebenky.html
11. Port of Nakatani Shuyo's language-detection library, Feb 16, 2020 https://pypi.org/project/langdetect/
12. Steven Loria, TextBlob: Simplified Text Processing, April 26, 2020. https://textblob.readthedocs.io/en/dev/
13. Pratima Upadhyay, Removing stop words with NLTK in Python, March 30, 2017. https://www.geeksforgeeks.org/removing-stop-words-nltk-python/
14. Mohamed Afham, "Twitter Sentiment Analysis using NLTK, Python", towardsdatascience, 2019
15. OLEG YEGOROV, "Why do Russians use parentheses instead of smileys?", RBTH, 2017. Available: https://www.rbth.com/lifestyle/326858-why-russians- use-parentheses
16. Jeff Hale, Scale, Standardize, or Normalize with Scikit-Learn, Mar 4, 2019. https://towardsdatascience.com/scale-standardize-or-normalize-with-scikit-learn-6ccc7d176a02
17. A Ydobon, How to interpret a Classification Report, Jan 25, 2020. https://medium.com/@a.ydobon/justforfunpython-how-to-interpret-a-classification-report-189edc487460
18. Abhishek Sharma, Confusion Matrix in Machine Learning, Dec 13, 2019. https://www.geeksforgeeks.org/confusion-matrix-machine-learning/