ЖАРАТЫЛЫСТАНУ ЖӘНЕ ТЕХНИКАЛЫҚ ҒЫЛЫМДАР

NATURAL AND TECHNICAL SCIENCES

IRSTI 55.01.11

Makhul Maulen[1]
[1]Suleyman Demirel University, Kaskelen, Kazakhstan

COMPARISON OF DIFFERENT CLASSIFICATION MODELS FOR SENTIMENT ANALYSIS

Abstract. In this work, we explored sentiment analysis techniques of texts using the example of product comments in the Kazakh language. To do this, we used machine learning methods such as Naive Bayes, Random Forest, Logistic Regression and Support Vector Machine, as well as text processing tools: CountVectorizer and TfidfVectorizer. In the process of work, experiments were carried out with different configurations of models and parameters of vectorizers. To assess the quality of the models, we used accuracy, precision, recall and F1-score metrics. The research findings indicated that the application of machine learning techniques make it possible to achieve high accuracy in sentiment analysis of comments. The best results were obtained using the Support Vector Machine and TfidfVectorizer. This study can be used to further improve the systems for sentiment analysis of comments in the Kazakh language, which can be useful in monitoring public opinion in various areas, including business.

Keywords: Kazakh language, sentiment analysis, Naive Bayes, Random Forest, Support Vector Machine, Logistic Regression, Scikit-learn.

***

Андатпа. Бұл жұмыста біз қазақ тіліндегі өнімдерге түсініктемелер мысалында мәтіндердің көңіл-күйін талдау әдістерін зерттедік. Ол үшін біз Naive Bayes, Random Forest, Logistic Regression және Support Vector Machine сияқты машиналық оқыту әдістерін, сондай-ақ мәтінді өңдеу құралдарын қолдандық: CountVectorizer және TfidfVectorizer. Жұмыс барысында модельдердің әртүрлі конфигурацияларымен және векторизаторлардың параметрлерімен тәжірибелер жүргізілді. Модельдердің сапасын бағалау үшін біз accuracy, precision, recall және F1-score көрсеткіштерін қолдандық. Зерттеу нәтижесінде машиналық оқыту әдістері түсініктемелердің көңіл-күйін талдауда жоғары дәлдікке қол

жеткізуге мүмкіндік беретіні анықталды. Ең жақсы нәтижелер Support Vector Machine мен TfidfVectorizer көмегімен алынды. Бұл зерттеуді қазақ тіліндегі пікірлердің көңіл-күйін талдау жүйелерін одан әрі жетілдіру үшін пайдалануға болады, бұл әртүрлі салаларда, соның ішінде бизнесте қоғамдық пікірді бақылауда пайдалы болуы мүмкін.

Түйін сөздер: Қазақ тілі, көңіл-күйді талдау, Naive Bayes, Random Forest, Support Vector Machine, Logistic Regression, Scikit-learn.

***

Аннотация. В данной работе мы исследовали приемы анализа тональности текстов на примере комментариев к товарам на казахском языке. Для этого мы использовали методы машинного обучения, такие как Наивный Байес, Случайный лес, Логистическая регрессия и Метод опорных векторов, а также инструменты обработки текста: CountVectorizer и TfidfVectorizer. В процессе работы проводились эксперименты с различными конфигурациями моделей и параметрами векторизаторов. Для оценки качества моделей мы использовали метрики accuracy, precision, recall и F1-score. В результате исследования было выявлено, что методы машинного обучения позволяют добиться высокой точности анализа тональности комментариев. Наилучшие результаты были получены с использованием Метод опорных векторов, и TfidfVectorizer. Данное исследование может быть использовано для дальнейшего совершенствования систем анализа тональности комментариев на казахском языке, что может быть полезно при мониторинге общественного мнения в различных сферах, в том числе и в бизнесе.

Ключевые слова: Анализ тональности, Казахский язык, Наивный Байес, Случайный лес, Метод опорных векторов, Логистическая регрессия, Scikit-learn.

I. Introduction

In the modern world, a huge amount of information is stored and transmitted in text form. Extracting meaningful information from this amount of data is an important task that can be solved with the help of sentiment analysis of texts. Text sentiment analysis is the process of determining the emotional coloring of a text, which can be positive and negative. Such analysis can be useful in many areas such as marketing, social research, recommendation systems, and more. Various machine learning methods such as Naive Bayes, SVM, Random Forest and Logistic Regression are used to solve the text sentiment analysis problem. In this paper, we analyze the results

of applying these models to sentiment analysis of texts and compare their effectiveness.

In this research, we will observe different machine learning models and algorithms to understand how they work and what factors can affect their performance. For our experiments, we use two datasets in Kazakh language: a training set of 209 comments and a test set of 100 comments. We will review the basic principles of how classification models work and run experiments to analyze their performance on our dataset. Our goal is to understand how to choose the best model for a classification problem in Kazakh language and how to optimize its performance.

II. Literature review

Sentiment analysis is mainly used in recommendation systems. May find good examples and practices of using sentiment analysis in papers about recommendation systems.

The paper [4] describes a method that uses sentiment analysis to recommend products and services based on user feedback. The authors propose a recommendation system that uses machine learning techniques such as sentiment analysis to detect user sentiment and determine how satisfied the user is with a product or service. Based on this analysis, the system recommends products and services that are most suitable for a particular user. The article also describes the use of sentiment analysis methods, such as classification and sentiment analysis methods, to detect the sentiment of reviews. Next, the authors describe how these methods are used to create machine learning models that can be used to recommend products and services. In general, the article presents a new method for creating recommendation systems based on sentiment analysis of user reviews. The authors argue that this approach can be useful for companies looking to improve their referral systems and increase customer satisfaction. In the next paper [5] proposes a book recommendation system that incorporates sentiment analysis to predict users' preferences. The authors collected data on users' book ratings and reviews from Goodreads, a popular social networking site for book lovers. They then performed sentiment analysis on the reviews to determine the users' emotional reactions to the books. The system uses a hybrid approach that combines content-based filtering and collaborative filtering to generate recommendations. The content-based filtering method uses the sentiment analysis results to calculate the similarity between books, while the collaborative filtering method makes use of the ratings given by similar users to generate recommendations. The authors evaluated the proposed system using a dataset of book reviews and ratings and compared it with other recommendation algorithms. The results showed that the proposed system outperformed the other algorithms in terms of recommendation accuracy. Overall, the proposed book recommendation system represents a contribution to the growing body of literature on sentiment-based recommendation systems

and provides insights into how sentiment analysis can be incorporated into the book recommendation process.

III. Method and Materials

Classification models are machine learning algorithms that are trained on labeled data to determine the class to which an object belongs. In classification problems, we have a data set, each element of which has its own class [10]. The task of a classification model is to learn how to classify new data based on the knowledge obtained from the training dataset. In our cases, classify reviews as positive or negative, and in some cases as neutral.

There are many different classification models [10][12], each with its own strengths and weaknesses depending on the problem to be solved. In our work, we will explore the Logistic Regression, Naive Bayes, Support Vector Machine and Random Forest models.

Logistic regression is a model that uses a linear combination of features to predict the likelihood of an object being assigned to a particular class. Logistic regression is a simple and fast model that is often used for binary classification problems.

Naive Bayes is a statistical model that is based on Bayes' theorem and assumes that each feature is independent of the others. This model is very fast and efficient, especially when dealing with large amounts of data.

Support Vector Machine (SVM) is a model that finds the optimal hyperplane to separate two classes in a multidimensional feature space. [10] SVM can handle data with many features and is an efficient model for classification problems with non-linear decision boundary.

Random forest is a model that builds many decision trees and combines their results to make more accurate predictions. Random Forest is a powerful and flexible algorithm that can handle large amounts of data and is noise tolerant.

In the comparison of models part, we will describe each model, their advantages and disadvantages for our study.

Data description. [8] Our experiment begins with data collection. To collect data, we used different methods, such as asking the opinion of acquaintances about books, films and music. We also used different platforms, such as a telegram bot that generated comments in the Kazakh language and collected data in online stores as a Kaspi of product reviews. So we collected 309 comments and used 209 for training the models and 100 for the test. Then, when our data was ready, we labeled each comment as positive (+) and negative (-), then removed punctuation marks and lowercase all comments and prepared these data for the experiment.

Data preprocessing. The vectorization method TfidfVectorizer() from the Scikit-learn library was used for preprocessing. TfidfVectorizer() is a feature extraction method for text data. It is used to transform a collection of documents (for example, a corpus of texts) into a feature matrix, where each

document is represented as a vector of numbers. For three models SVM, Random Forest and Logistic regression used this method and for Naïve Bayes model used CountVectorizer() method from the Scikit-learn library which takes into account only the frequency of occurrence of words in each comment. TfidfVectorizer() takes the following steps:

- Tokenization: splitting text into separate words (tokens).
- Stopword Removal: Removing common words that don't carry much meaning, such as "a", "an", "the", etc.
- Vectorization: converting text into a vector of numbers.

TfidfVectorizer() uses the TF-IDF (term frequency–inverse document frequency) method, which takes into account the frequency of occurrence of a word in a document and the overall frequency of occurrence of a word in a collection of documents.

TF-IDF takes into account the fact that words that appear frequently in each document may not be of great importance for classification, while rare words that occur only in some documents may be of great importance. Applying TfidfVectorizer() to text data allows to take into account the importance of words in classification and improve the accuracy of the model.

To train classification models, the Scikit-learn library was used, which provides many tools for machine learning, including text data classification. Scikit-learn also provides many other classification models and data science tools, making it one of the most popular libraries for machine learning and data analysis in the scientific community. To train the models, we first splitted the data into train and test. Then we trained each model using the SVC, RandomForestClassifier, LogisticRegression and MultinomialNB methods from the Scikit-learn library.

Evaluation. The next stage of our study was to evaluate the performance of the algorithm. We used several metrics to measure the quality of the model: accuracy, precision, recall, and F1-score.

- Accuracy - a metric that measures the proportion of correctly classified objects from the total number of objects.
- Precision is a performance metric that calculates the ratio of correctly predicted positive instances to the total instances predicted as positive.
- Recall it is a metric that measures the proportion of correctly identified positive cases among all actual positive cases.
- F1-score is a commonly used metric in cases where the classes are imbalanced. It is calculated as the harmonic mean of precision and recall.

All these metrics calculated using the accuracy_score, precision_score, recall_score and f1_score functions from the scikit-learn library.

IV. Discussion and Results

The results showed that the SVM model performed best across all metrics (accuracy, recall, F1-measure, and accuracy), reaching an accuracy of

93%. This is because the SVM is a more complex model capable of dealing with non-linear class boundaries. Logistic regression and Naive Bayes also gave good results, but slightly outperformed by SVM. Logistic regression gave an accuracy of 88%, and Naive Bayes also gave 88%. On the other hand, the Random Forest model showed the lowest accuracy - 83%, which may be due to the fact that this model does not work very well with text data and requires a large number of trees to achieve good results. So, we can conclude that the SVM model is the most effective for classifying text data in a given data set, but it is also worth considering the features of each model and the context of use. But our experiment did not end with these results, to test the models we asked two people for their opinion. What category are test comments classified as positive or negative and compared the results of the models with the opinions of people.

Table 1. Comparing the accuracy of various machine learning algorithms

Algorithm	accuracy	precision	recall	F1-score
SVM	93%	93%	96%	94%
Random Forest	83%	81%	92%	86%
Logistic regression	88%	85%	96%	90%
Naïve Bayes	88%	95%	83%	88%

For addition, we have created a histogram to study the comparative frequency of the use of words in positive and negative comments. The more words in the comment, the higher the column on the histogram. In many cases, positive comments are more likely to contain fewer words than negative comments. Because users describe every moment that they don't like. If they like the content or product, then just briefly leave reviews. But in our case, you can see that positive comments contain more words than negative comments. This suggests that users who leave positive comments tend to be more verbose and provide more detailed feedback than those who leave negative comments. This finding may have implications for how we analyze and interpret user feedback in the future, as it suggests that positive comments may require more attention and analysis to fully understand and address user needs and preferences.

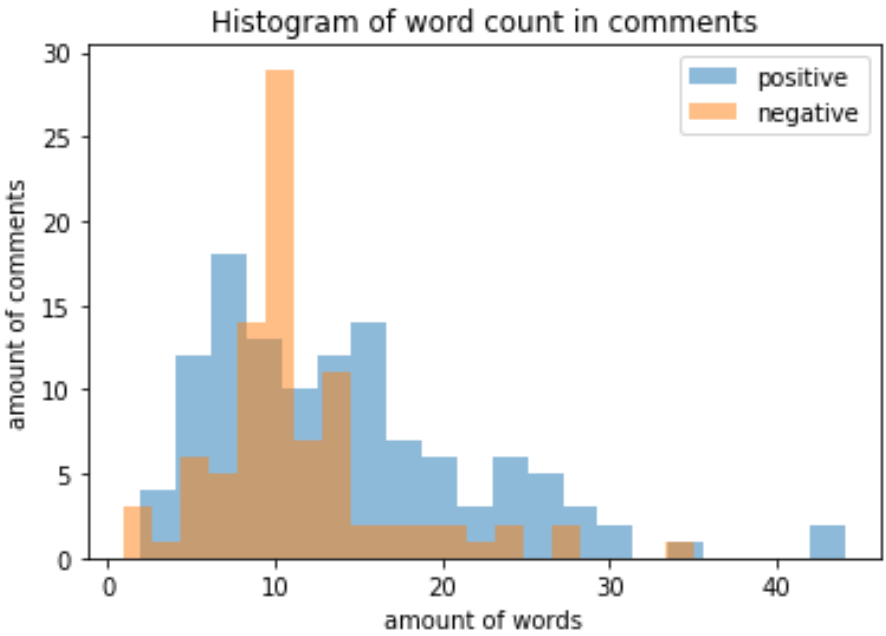

Figure 1. Histogram showing the amount of words in positive and negative comments.

V. Conclusion

In this study, four classification models - Logistic regression, Naive Bayes, Random Forest and SVM - were analyzed on a dataset consisting of comments. Accuracy, recall, F1-measure and accuracy were used as performance metrics. The results showed that the SVM model gave the best results across all metrics, which is due to its ability to work with non-linear class boundaries. Logistic regression and Naive Bayes also showed good results, but slightly outperformed by SVM. The Random Forest model showed the lowest accuracy, which may be due to its limitations in working with text data.

Thus, depending on the task, it is possible to use different classification models for working with text data. However, in this case, the SVM model showed the best results for all metrics. But when we compared the results of the models with human opinions there were some differences. The SVM model indicated some comments as negative but according to people they are positive, also some comments were indicated by the model as negative but according to people these are positive comments. Since there were good results in the SVM and Naive Bayes models, we compared them. The Naive Bayes fit more well with people's opinions. In the course of the study, when comparing people's opinions about comments with the results of models, people found it difficult to classify as positive or negative because some comments are similar to neutral. Since we don't have a lot of data, we rarely see such comments. But for

future work, we can collect neutral comments and add them to our data, then train them too.

In general, the use of the Scikit-learn library in combination with the TfidfVectorizer text vectorization method allows to process text data and train classification models with high accuracy and efficiency.

It is also worth noting that our study was conducted on comments in the Kazakh language in order to determine their emotional coloring, which is associated with the task of sentiment analysis. In recent years, with the growth in the number of generated texts on the Internet, sentiment analysis has become an important task in the field of machine learning, as it allows to automatically determine the attitude of people to certain objects or phenomena. The use of machine learning methods for the analysis of the Kazakh language can greatly simplify and speed up the process of processing large amounts of text data in this language. Thus, the results of our study can be useful for further development and improvement of models for the classification and analysis of texts in the Kazakh language.

References

1 B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up? sentiment classification using machine learning techniques,” arXiv preprint cs/0205070, 2002.

2 B. Liu, “Sentiment analysis and opinion mining,” Synthesis lectures on human language technologies, vol. 5, no. 1, pp. 1–167, 2012.

3 B. Pang, L. Lee et al., “Opinion mining and sentiment analysis,” Foundations and Trends® in information retrieval, vol. 2, no. 1–2, pp. 1–135, 2008.

4 Amel Ziani, Nabiha Azizi, Didier Schwab and Monther Aldwairi, "Recommender System Through Sentiment Analysis", Conference: 2017 2nd International Conference on Automatic Control.

5 Mounika, A., Saraswathi, S. (2021). Design of Book Recommendation System Using Sentiment Analysis. In: Suma, V., Bouhmala, N., Wang, H. (eds) Evolutionary Computing and Mobile Sustainable Networks. Lecture Notes on Data Engineering and Communications Technologies, vol 53. Springer, Singapore.

6 R. S. Jagdale, V. S. Shirsat, and S. N. Deshmukh, “Sentiment analysis on product reviews using machine learning techniques,” in Cognitive Informatics and Soft Computing: Proceeding of CISC 2017. Springer, 2019, pp. 639–647.

7 Francesco Ricci, Lior Rokach, Bracha Shapira, Paul B. Kantor “Recommender Systems Handbook”. Springer Science+Business Media, LLC 2011.

8 “Practical Recommender system” by Kim Falk Copyright © 2019 by Manning Publications Co.
9 Benjamin Bengfort, Rebecca Bilbro, and Tony Ojeda (2018). Applied Text Analysis with Python Enabling Language-Aware Data Products with Machine Learning. Published by O’Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.
10 Programming Collective Intelligence by Toby Segaran (2017). Published by O’Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.
11 Nitin Indurkhya and Fred J. Damerau (2010). “Handbook of Natural language processing” second edition, International Standard Book Number-13: 978-1-4200-8593-8 (Ebook-PDF).
12 Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning (Integrated Series in Information Systems, 36) 1st ed. 2016 by Shan Suthaharan