IRSTI 16.01.11

Bilakhanova[1], A. Ydyrys[2], N. Sultanova[3]
[1,2,3]Suleyman Demirel University, Kaskelen, Kazakhstan

IDENTIFYING SPAM MESSAGES FOR KAZAKH LANGUAGE USING HYBRID MACHINE LEARNING MODEL

Abstract. This paper describes a spam detection system for Kazakh Language using Hybrid Machine Learning Model. The lack of spam detection systems in the Kazakh language calls for the need of a proposed system that can identify unwanted messages. The system integrates multiple Machine Learning algorithms to accurately classify spam and non-spam messages. The performance of the system is evaluated using metrics such as accuracy, precision, recall, and F1-score. Results show that the proposed solution outperforms existing spam detection techniques in terms of detecting spam with a low false positive rate and high accuracy. The findings of this research contribute to the development of effective spam detection systems for the Kazakh language and provide insights for future work in this field.

Keywords: Spam classification, spam detection, spam filtering methods, machine learning, data preprocessing for Kazakh language.

***

Аңдатпа. Бұл мақалада гибридті машиналық оқыту моделін қолдану арқылы қазақ тіліндегі спам анықтау жүйесі сипатталған. Қазақ тілінде спамды анықтау жүйелерінің жоқтығы спам хабарламаларды анықтай алатын ұсынылып отырған жүйені қажет етеді. Жүйе спам және спам емес хабарламаларды дәл жіктеу үшін бірнеше машиналық оқыту алгоритмдерін біріктіреді. Жүйенің өнімділігі accuracy (дәлдік), precision (дәлдік), recall (толықтық) және F1 ұпайы сияқты көрсеткіштер арқылы бағаланады. Нәтижелер көрсеткендей, ұсынылған шешім төмен жалған оң (false positive) көрсеткіші мен жоғары дәлдікпен анықтау бойынша қолданыстағы спамды анықтау әдістерінен асып түседі. Бұл зерттеудің нәтижелері қазақ тіліне арналған спамдарды анықтаудың тиімді жүйелерін жасауға септігін тигізеді және осы саладағы болашақ жұмыс туралы түсінік береді.

Түйін сөздер: Спам классификациясы, спамды анықтау, спамды фильтрлеу әдістері, машиналық оқыту, қазақ тіліне арналған деректерді алдын ала өңдеу.

***

Аннотация. В этой статье описывается система обнаружения спама для казахского языка с использованием модели гибридного машинного обучения. Отсутствие систем обнаружения спама на казахском языке вызывает потребность в предлагаемой системе, которая может идентифицировать нежелательные сообщения. Система объединяет несколько алгоритмов машинного обучения для точной классификации спамовых и не спамовых сообщений. Производительность системы оценивается с использованием таких показателей, как accuracy (точность), precision (точность), recall (полнота) и оценка F1. Результаты показывают, что предлагаемое решение превосходит существующие методы обнаружения спама с точки зрения обнаружения спама с низким уровнем ложных срабатываний и высокой точностью. Результаты этого исследования способствуют разработке эффективных систем обнаружения спама для казахского языка и дают представление о будущей работе в этой области.

Ключевые слова: Классификация спама, обнаружение спама, методы фильтрации спама, машинное обучение, предварительная обработка данных для казахского языка.

I. Introduction

Spam is the bulk sending of unwanted messages to multiple recipients. Despite over a decade of efforts to combat it, the prevalence of spam remains significant globally despite advancements in techniques for addressing it [1]. Traditionally, spam was used for promoting products and services to potential customers. However, it evolved into a tool for hacking and spreading viruses. To address this problem, various methods for detecting and filtering spam have been proposed by scientists and researchers. Following are the different categories of spam filtering methods: Case Base Spam Filtering Methods; Content Based Filtering Methods; List Based Filtering Methods; Heuristic or Rule Based Spam Filtering Methods; Adaptive Spam Filtering Methods [2].

Although the existing literature on spam detection mostly focuses on widely spoken languages, such as English, there is currently a gap in research on this topic, as there are limited resources and tools available for identifying spam messages in Kazakh language. Some potential research problems/gaps for identifying spam messages in Kazakh language include:

Lack of annotated datasets: There is a lack of annotated datasets of spam and non-spam messages in Kazakh language, which makes it difficult to train and evaluate machine learning models for spam detection. Developing such datasets would be an important first step in addressing this research problem.

Language-specific features: The linguistic features of Kazakh language may differ from other languages, and existing feature extraction methods may not be effective in identifying spam messages in Kazakh. Developing language-

specific feature extraction methods for Kazakh language could be a research problem to address.

Limited use of machine learning: Developing and evaluating different machine learning approaches for spam detection in Kazakh language could be a potential research problem.

Our goal in tackling these research problems is to enrich the current literature by proposing hybrid machine learning models that can detect spam messages in the Kazakh language.

II. Literature review

The literature review section of this paper provides a comprehensive overview of the existing research on spam filtering techniques utilizing ML algorithms. Various classification algorithms and techniques have been proposed in the literature, and this section critically evaluates the strengths and limitations of each approach. The classifiers evaluated in this section will be used further in our work to develop a novel spam filtering algorithm that combines the strengths of multiple classifiers while minimizing their weaknesses.

A. Naive Bayes Filtering Method

Naive Bayes spam filtering is a technique that uses Bayes' theorem. During the training process, the filter calculates and stores the "weight" of each word encountered in the text. A message is then classified as either "spam" or "non-spam" depending on whether its "weight" surpasses a set threshold. The formula for calculating the probability that a message includes a specific spam word is shown below [3]:

$$P(sp|w) = \frac{P(sp) * P(w|sp)}{P(sp) * P(w|sp) + P(nsp) * P(w|nsp)} \quad (1)$$

The probability that a message is spam given the presence of a specific word is represented by P(sp/w). The overall probability of a message being spam is represented by P(sp), while P(w/sp) indicates the probability of a certain word appearing in a spam message. The overall probability of a message being non-spam is represented by P(nsp), and the probability of a specific word appearing in a non-spam message is represented by P(w/nsp).

Overall, while Naive Bayes is a widely used and effective algorithm for spam filtering, it is not without its limitations. Naive Bayes assumes that the features used for classification are independent of each other. This assumption may not hold true for some features in real-world datasets, which can lead to inaccurate classification.

B. K-nearest neighbor

The K-nearest neighbor (KNN) algorithm is a well-used ML model for classification issues. It works by assigning the new message to the class that's most frequent among its k nearest neighbors, whose classifications are already known. The main challenge of this algorithm is that its computational cost is

higher compared to other ML algorithms [4]. The search for nearest neighbors requires comparing the object being classified with every object in the sample, requiring a large number of linear operations proportional to the sample size.

C. Support vector machine

The Support Vector Machine (SVM) algorithm is primarily used for binary classification (spam or non-spam), but can also be adapted for multi-classification problems. It works by finding the best boundary to separate positive and negative samples and determining if a new message is classified as spam or not [3]. Numerous studies have shown the effectiveness of SVM for spam filtering, with an accuracy of 96% [5]. However, it can be computationally expensive to train, sensitive to the choice of kernel function, prone to overfitting, and require tuning of hyperparameters.

D. Decision making tree

A Decision Tree (DT) is a hierarchical tree structure used for filtering spam messages. The below figure is a flow-chart of a DT.

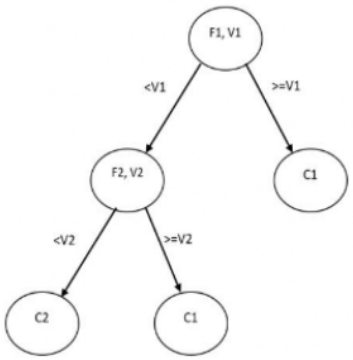

Figure 2. Flow-chart of a Decision Tree.

The DT process involves the use of input words (F), their frequencies (V), and labels (C) to determine if a text message is spam or non-spam. One of the biggest challenges in using DT is finding the optimal size of the final tree. In addition to this, the learning process also involves solving other problems such as selecting the most effective attribute for splitting, deciding when to stop learning, choosing the best pruning method to reduce the size of the tree, and estimating the accuracy of the resulting tree [7].

III. Methodology

In the following sections, we present methods and strategies, including data reading, preprocessing, and text classification workflow.

A. Dataset

The unique aspect or novelty of our work is that the dataset was meticulously gathered by us through manual means. Collecting the dataset manually allowed us to have full control over the quality and accuracy of the data, making the results of our work more reliable and trustworthy. This manual collection process was time-consuming and required significant effort, but it was necessary to ensure the validity and reliability of the data used in our work. The

dataset contains 2000 rows and 2 columns, where each message is classified as either non-spam (0) or spam (1). Out of the 2000 messages, 400 are labeled as spam and 1600 as non-spam. Given the limited amount of spam messages in real-life situations, having 400 spam messages, which constitute 20% of the dataset, is considered sufficient for the particular problem at hand.

B. Data Preprocessing

In ML preprocessing the dataset is crucial for transforming it into a computer-readable format. The following are some examples of the data preprocessing techniques used in our code. These steps help to improve the quality of the data and to make it more suitable for use in ML models.

Stopwords are used to eliminate words that are too common and lack significant information. In the Kazakh language, “Shylau” (Шылау) can be attributed to such words. These service words do not have an independent lexical meaning and are not independent members of the sentence. The string punctuation in Python is predefined in the string module and consists of all characters as a string. It is utilized to eliminate punctuation from a given sentence. By removing punctuation, the model can focus on the relevant words in the text rather than the irrelevant punctuation marks. The process of reducing inflectional endings from words is called lemmatization. The root word in lemmatization is known as the lemma. The kaz-nlp tool was used in our work for morphological processing, including lemmatization. This set of tools addresses a range of Natural Language Processing (NLP) problems. To lemmatize words, sentences must be split into individual tokens and passed to the "tag-sentence" method [8].

The above and other methods such as lowercasing, tokenization, removal of links "http", "bitly", "www", and numbers were combined into a single preprocess-text function that can be utilized as a pipeline for preprocessing Kazakh datasets. The pipeline can be applied to each individual value in the dataset, resulting in preprocessed data.

C. Text Classification Workflow

Our work involves using the Term Frequency Inverse Document Frequency (TFIDF) vectorization technique on preprocessed data to determine the significance of each word in a text. This conversion from text to numerical representation is necessary for ML algorithms to effectively process the data. Next, we split the data into training and testing sets, with the goal of training the model on the training set and evaluating its performance on the test set. After import and training, classifiers can make predictions, which can be stored in a variable. In our work, we utilized the following algorithms: SVM, KNN, Multinomial Naive Bayes (MNB), DT, Random Forest Classifier (RFC), and Logistic Regression (LR). All of these classifiers have readily available Python packages, making them easy to implement and integrate into our work.

IV. Results

A. Non-hybrid Machine Learning Models Results

This section evaluates performance of non-hybrid standalone algorithms through accuracy, precision, recall and F1-score measurements. To assess the classifier's accuracy and precision, we compare the predictions to the actual correct labels. Accuracy in the context of spam messages is computed by dividing the number of correctly predicted instances by the total number of predictions made by the classifier, while precision measures the classifier's ability to accurately identify non-spam messages. Recall is calculated as the number of true positive predictions of spam messages divided by the total number of actual spam messages present in the dataset. F1 score is a combined metric that balances precision and recall by taking the average of their harmonic mean. The table below shows the result of the metrics for each classification method on the test dataset:

Table 1. Result metrics (Accuracy, Precision, Recall, F1-Score) for non-hybrid models.

	Accuracy	Precision	Recall	F1-Score
SVM	0.975758	0.944444	0.708333	0.708333
KNN	0.963636	0.875000	0.583333	0.700000
RFC	0.960606	1.000000	0.458333	0.628571
LR	0.960606	1.000000	0.458333	0.628571
MNB	0.954545	1.000000	0.375000	0.545455
DT	0.939394	0.590909	0.541667	0.565217

Based on the obtained metrics, we can observe the following: SVM has the highest accuracy and a relatively high f1-score, which balances precision and recall. RFC, LR and MNB have a high precision, but relatively low recall, which means they are good at avoiding false positives (all the positive predictions made by these models are correct) but not as good at detecting all the spam messages. DT has a relatively low precision and f1-score compared to other algorithms, which indicates that it is not performing as well. SVM and KNN have the best overall performance for spam detection based on these metrics.

B. Hybrid Machine Learning Models Results

A non-hybrid models (SVM, KNN, etc.) are simpler, faster, and easier to interpret and implement, making them a good option for simpler problems. But they use a single ML algorithm to make predictions, therefore they are limited by the performance of the chosen algorithm. A hybrid ML model for spam detection is a model that combines two or more individual ML algorithms to make predictions. The advantage of a hybrid ML model is that it can combine the strengths of different algorithms to improve the overall performance of the model. For example, one algorithm may have a high precision but a low recall, while another algorithm has a low precision but a high recall. By combining these algorithms, a hybrid model can achieve a good balance between precision and recall, resulting in a higher overall performance. In our work, we try to combine algorithms (SVM, MNB, DT, RFC, KNN, LR) of each with each to create a hybrid ML model for improved performance in spam detection using

the VotingClassifier method, which combines the predictions of the base classifiers using hard or soft voting. Hard voting is a simple voting scheme that takes the majority vote of the base classifiers. For example, SVM and KNN predicts the message as spam and RFC predicts it as not spam, then the majority vote would be to classify the message as spam, since two out of the three classifiers predicted it as spam. If you have an even number of classifiers in your ensemble, as in our case, and the number of votes for each class is equal, the majority vote could be ambiguous. In such cases, the VotingClassifier method could use soft voting instead of other tie-breaking strategies for hard voting to resolve the ambiguity (randomly, weighted voting scheme, etc.)[9]. In soft voting, each classifier assigns a probability score to each class label (spam or non-spam class), rather than a binary prediction of either 0 or 1, and chooses the class with the highest probability. In this case, it would depend on the exact probability scores assigned by the classifiers. If SVM assigned a much higher probability to the spam class than KNN assigned to the non-spam class, then the soft voting would predict the message as spam. The table below presents the evaluation metrics for the hybrid models.

Table 2. Result metrics (Accuracy, Precision, Recall, F1-Score) for hybrid models.

	Accuracy	Precision	Recall	F1-Score
SVM + KNN	0.981818	0.875000	0.875000	0.875000
SVM + MNB	0.981818	1.000000	0.750000	0.857143
SVM + DT	0.969697	0.791667	0.791667	0.791667
SVM + RFC	0.972727	0.894737	0.708333	0.790698
SVM + LR	0.975758	0.863636	0.791667	0.826087
MNB + KNN	0.978788	0.904762	0.791667	0.844444
MNB + LR	0.981818	0.950000	0.791667	0.863636
MNB + DT	0.978788	0.869565	0.833333	0.851064
MNB + RFC	0.954545	0.680000	0.708333	0.693878
RFC + LR	0.972727	0.894737	0.708333	0.790698
RFC + DT	0.969697	0.818182	0.750000	0.782609
RFC + KNN	0.966667	0.740741	0.833333	0.784314
KNN + LR	0.963636	0.875000	0.583333	0.700000
KNN + DT	0.963636	0.875000	0.583333	0.700000
DT + LR	0.945455	0.650000	0.541667	0.590909

Based on the given results, the hybrid model "SVM + KNN" has the best performance in terms of accuracy, precision, recall, and f1-score. This combination achieved an accuracy of 0.9818 which means that the model correctly classified 98.18% of the spam messages. The precision of 0.875, meaning that the model is not classifying many non-spam messages as spam, recall of 0.875, indicating that 87.5% of the actual spam messages were correctly classified as spam, and F1-score of 0.875. The findings indicate that hybrid

models surpass existing non-hybrid spam detection methods in terms of accurately identifying spam with a low rate of false positives.

B. Conclusion

The results show that both the SVM and KNN performed well compared to other models such as RFC, LR, MNB, and DT based on all four metrics. The combination of SVM and KNN further improved their performance, which supports the conclusion that combining two well-performing models leads to even better results. This suggests that the hybrid model is better suited for this problem and dataset being evaluated, and will result in better results compared to using a single model. Future work aims to expand the dataset, which is expected to enhance its performance. Improving the processing of the Kazakh language will not only benefit research in this specific language, but also other languages with similar challenges. Overall, there is significant potential for future work to further improve the processing of the Kazakh language.

References

1 Kaspersky Lab, "Global spam volume as percentage of total e-mail traffic from 2011 to 2021".
URL: https://www.statista.com/statistics/420400/spam-email-traffic-share-annual.

2 Emmanuel G.D., Joseph S.B., Haruna Ch., Shafi'i M.A., Adebayo O.A., Opeyemi EA., "Machine learning for email spam filtering: review, approaches and open research problems", Heliyon, Volume 5, Issue 6, 2019.

3 Ganiev S.K., Khamidov Sh.J., Olimov I.S., "Analysis of machine learning methods for filtering spam messages in email services", In International Conference on Information Science and Communications Technologies (ICISCT), November 04-06, 2020.

4 Mansoor RASA, Nathali D.J., Muhana M.A, "A Comprehensive Review on Email Spam Classification using Machine Learning Algorithms", In International Conference on Information Networking (ICOIN), January 13-16, 2021.

5 Niken L.O., Eko H.R., Christy A.S., DRIM Setiadi, "Comparison of Multinomial Naïve Bayes Classifier, Support Vector Machine, and Recurrent Neural Network to Classify Email Spams", In International Seminar on Application for Technology of Information and Communication (iSemantic), September 19-20, 2020.

6 Fahima H., Mohammed N.U., Rajib K.H., "Analysis of Optimized Machine Learning and Deep Learning Techniques for Spam Detection", In IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), April 21-24, 2021.

7 Ahmed I.T; Safaa S.I., "An Intelligent Hybrid Technique of Decision Tree and Genetic Algorithm for E-Mail Spam Detection", In Ninth International Conference on Intelligent Computing and Information Systems (ICICIS), December 08-10, 2019.

8 Makhambetov O., Makazhanov A., Sabyrgaliyev I., Yessenbayev Zh., "KazNLP: NLP tools for Kazakh language". Last modified on Nov 21, 2018. URL: https://github.com/makazhan/kaznlp.

9 Cornelio, C., Donini, M., Loreggia, A. et al. Voting with random classifiers (VORACE): theoretical and experimental analysis. Autonomous Agents and Multi-Agent Systems 35, May 21, 2021.