Computer Science • 27(1) 2026 https://doi.org/10.7494/csci.2026.27.1.6738

Güncel Sariman

A DEEP LEARNING DRIVEN TEXT CLASSIFICATION APPROACH WITH NAMED ENTITY RECOGNITION

Abstract *In natural language processing with text data, which forms the basis of the studies in the field of artificial intelligence, various studies such as semantics and natural language generation are carried out, especially the solution of classification problems. This study aims to analyze the effect of detected named entities on text classification performance to make the text preprocessing stage more effective. In order to reduce the analysis time and increase the performance, after the classical preprocessing stage, word filtering was performed with Named Entity Recognition according to the thresholds determined in the 5% and 10% ranges. Analysis was performed with various machine learning, deep-learning algorithms, Bidirectional Encoder Representations from Transformers (BERT), and the obtained results are discussed in the last part of the study. In the problem of classifying 50,000 news texts, 93% with a support vector machine (SVM) algorithm in statistical classification with machine learning, 87% with long short-term memory (LSTM), and 83% with BERT success was achieved. In the analyses performed with LSTM and BERT, although the model performances were numerically lower, it was observed that the semantic integrity was stronger in text classification and that the success increased in general after Named Entity Recognition (NER) filtering. Thus, it can be interpreted that the dataset that is passed through the NER filter according to the threshold values positively affects the model's success in terms of time and performance.*

Keywords Named Entity Recognition, Natural Language Processing, Language Model, Machine Learning

Citation Computer Science 27(1) 2026: 119–145

1. Introduction

Natural language processing (NLP) is a subcategory of artificial intelligence and linguistics. In natural language processing technology there are goals, such as strengthening human-machine communication, making sense of large data sets, and increasing the efficiency of services. Although there are various natural language processing studies in the literature, one of the most widely preferred techniques is the study of text classification. The main objective of text classification studies is to identify the essential features of the problem. The classification approach is used to solve many text-based problems such as text summarization, sentiment analysis, headline modeling, text translation, question and answer (chatbot) systems. Today, many sectors attach great importance to data and continue their data-driven development. Most of this data is unstructured text from sources such as emails, customer-support chat transcripts, social media comments, surveys, news articles, columns, articles, and documents. E-archive systems have become very popular in recent years in organizations, especially to find the desired information from retrospective documents and to identify the desired information from archive documents. Searching and identifying information using documents digitized with optical character recognition (OCR) is extremely laborious and time-consuming. In the news and columns that accumulate on websites and in various news agencies; classification techniques are frequently used to make it easier to access and categorize the information sought. Text classification is a technique used to automatically match predefined classes related to a given text document. Text classification is usually performed using machine learning (ML) techniques [43]. Analysis of text data is performed using vectorial representations after conversion into fixed-length numerical data.

1.1. Named Entity Recognition on text classification

Named Entity Recognition (NER) models are used to categorize named entities (proper names, organizations, places, time expressions, etc.) in text according to their meaning. It was first defined in 1995 at the MUC-6 (Message Understanding Conference) [10]. NER helps to extract information from texts and this method is called "information extraction". In this model, different categories of special expressions are highly favored, such as noun entity expressions (ENAMEX), temporal (TIMEX), and numeric (NUMEX) entity expressions. Making inferences by creating qualified NER lists simplifies natural-language processing. With the results given by NER methods, the data set can be classified without examining it individually. It can also be sorted into defined categories and related data can be easily identified. The NER method is highly preferred in areas such as news, chatbots, machine translation, healthcare, customer relationship management, and enterprise risk management. Extracting meaningful information from large text data, and generating short information from long sentence structures, were possible with statistical or rule-based structures based on word weights, despite their low success in previous years.

In recent years, NER tags have been used extensively with machine learning techniques to identify words that have strong meanings in sentences. In systems where syntactic data such as e-mails, news texts, and system logs are dense, the success of the NER system in information extraction processes such as keyword extraction, and categorization is at a very important point in terms of both time and performance [44]. Although NER algorithms have generally been used for categorization in previous studies, the use of NER techniques in hybrid solutions and as part of a process has become more common in recent years.

1.2. Research problem

Many parameters affect learning performance and model success on text classification algorithms running in software systems. Some of the factors that negatively affect the process are not updating the data set during the process, not preprocessing the data set properly, and not selecting appropriate parameters and algorithms during the training and testing phases. Since the data obtained in text classification studies are usually obtained from social media or various internet sources, studies are performed with short and syntactically unstructured sentences. This brings some difficulties in the model training phase. When working with big data, breaking sentences into words and representing them with numerical vectors requires more processing power and time. Although the main purpose of data preprocessing stages is to improve performance, it is known that some preprocessing stages do not contribute positively to model success and reduce performance [42, 45].

In this study, we want to investigate how the extraction of named entities from the data set in the preprocessing stage affects the performance of traditional and modern classification algorithms. By removing the entities that can be detected with our model from the dataset, the number of tokens will be reduced the analysis time will be shortened and unnecessary word vectors will not be included in the model. Thus, it is thought that the model's performance will be positively affected both in terms of time and success.

1.3. Contribution

In recent years, text classification studies have focused on the performance of new models and algorithms on different data sets [24]. Especially in recent years, the success of large language models has become an important point in terms of algorithm selection for commercial applications [28]. It is known that preprocessing stages also affect model performance along with the use of new algorithms to improve performance in text classification studies [11, 37]. In this study, the impact of entity names on the performance of the proposed model will be observed. Although the impact of preprocessing stages on classification problems has been presented in studies, the effect of named entities on text classification performance was found in the literature [29] in only one study.

However, in this study, only filtering, was performed and the data set was minimized. It was not run on any algorithm. Situmeang [46] observed the performance of preprocessing stages in named entity detection with the Conditional random fields (CRF) algorithm and discussed the contribution to the model. In other studies discussing the effect of preprocessing stages on classification performance, algorithms such as convolutional neural network (CNN), LSTM, artificial neural network (ANN), random forest (RF), etc. were used [41, 42, 45]. The method proposed in our study has been included in the literature but has not been compared with basic approaches since it has not been run on algorithms yet. The algorithms used in this study include both classical and modern approaches.

1.4. Paper organization

In the first part of the study general information is given, followed by a literature review in the second part, the materials used, and the method to be applied in the third part. In the next part of the study, the findings obtained are presented in tables and graphs, the impact of the study is discussed in the conclusion section and suggestions are made.

2. Related works

In this section, studies on text classification and Named Entity Recognition are reviewed. During the reviews based on machine learning, deep learning, and language model-based algorithms, attention was drawn to the success rates and techniques in NER-related studies.

Ali [5], aims to automate the process of categorizing job applications and resumes using NLP and ML techniques. Various ML algorithms and NLP techniques are evaluated to measure the accuracy of the work and a solution is proposed that provides better accuracy and reliability in different environments. In the study, SVM, Naive Bayes, K-Nearest, and logistic tegression (LR) classifiers were used to achieve the best performance with more than 96% accuracy on more than 960 parsed resume datasets.

Uslu & Akyol [51] classified Turkish news texts using machine learning methods. The data set consists of categorized news texts. The data set was analyzed with support vector classifier, Random Forest and Naive Bayes Classifiers, and the results showed that the Bayes algorithm was the most successful method, with an accuracy rate of 91

In the Szczepanek's study [50], models were created to classify textual data about historic flood levels obtained from the Internet using named entity recognition techniques and advanced text analysis, deep learning, and spatial analysis techniques. In this study, the use of shortest-distance matrices is proposed between place names in the text.The obtained distance matrix was also shared as open data. In the SD-NER model study the F1 score value reached 92% in binary classification tasks, better than naivebayes and CNN. In this model, a structure was created using both spatial

analysis and the NER technique. This model is also designed to be applied to data sets where spatial information is important.

He & Zhang [18] suggest an association rule mining method (ARMTNER) based on named entity recognition and text classification because the accuracy of named entity recognition methods is low in Chinese language and it is difficult to obtain different categories with classical data mining methods. At first, the TextCNN model is used to extract word vectors. Then, with the Bidirectional Long Short-Term Memory (BI-LSTM) model, the context between the texts is discovered and the general features of the text are extracted. Finally, text sequence tagging and entity recognition are performed. Text classification and entity classification were performed at two levels, and frequent itemsets were mined using association rules. Experimental results show that the method obtained an F1 score of 97.3% on the Chinese named entity recognition open-source data set, and frequent item sets improved performance by 0.279%.

Hemati & Mehler [19] describe an approach to the BioCreative V.5 challenge that aims to identify and classify chemically named entities in academic papers. The named entity recognition task is transformed into a sequence-ranking labeling problem, and a sequence-ranking labeling system is presented for solving this task. They performed various experiments on algorithms with hyperparameter optimization. They also introduced the LSTMVoter model, a two-stage recurrent neural network implementation that integrates optimized ranking labelers as a single ensemble classifier. The results show that LSTMVoter outperformed each tagger, achieving an F1-score of 90.04% in the BioCreative IV CHEMDNER corpus and 89.01% in the BioCreative V.5 corpus for identifying chemical entity mentions in patents. The study highlights the importance of chemical and biomedical name recognition, demonstrating the potential for machine learning techniques to improve this task.

Ali et al. [6] proposed a new LSTM-based model to solve the problem of named entity recognition in Arabic texts. They explained that the LSTM technique is suitable for NER detection in sequential texts, and a pretrained word embedding technique is used to train the inputs. The ANERcorp data set was used to evaluate the model. According to the results of LSTM, GRU, BLSTM, and BLSTM+char Model, the BLSTM+char model achieved the highest F-Score result, with a success rate of 88%.

Suat-Rojas et al [47] developed a method that detects Spanish traffic accident data on Twitter. In the study, after the data collection system was developed, the messages were classified according to whether they contained accident information or not. The named entity recognition system was also used to separate tweets containing place or location names from the accident data. For this model, geographical locations are also translated into place-names through a system to generate named entities. In the study, the combination of doc2vec and SVM achieved the best result for the classification of tweets, with a success rate of 96.8%. The best result for the system

proposal that detects the location with NER was obtained with 91% in the Spacy model.

Perera et al. [40] studied the recently popular Biomedical named entity recognition in 7 different models using textual data of food and dietary components. Especially the dictionary-based model, CRF, and the FoodCoNER hybrid model, which is a combination of these two models, as well as the deep-learning-based BERT, BioBERT, and Electra models were used in the study. We found that FooDCoNER not only leads to the best overall results when compared to deep language models but also that FooDCoNER is much more efficient in terms of training data runtime and sample size requirements.

In their study, Aydoğan & Karci [9] drew attention to the scarcity of Turkish text processing studies and created word vectors on 2 datasets using word embedding techniques. They performed classification on 1.5 million words using deep-learning techniques–CNN, RNN, LSTM, and GRU. The word embedding performances of these models were compared and their effect on accuracy rates was observed. GRU and LSTM models were found to be more successful than the other models, and preprocessed word vectors improved the performance by 5-7%.

Pankaj et al. [36] used reviews on smartphones that are sold on the Amazon website as a data set in an Amazon product reviews study. By performing sentiment analysis on this data, they categorized the comments into three categories: positive, neutral, and negative. They preprocessed and filtered the data and measured the accuracy of the results. Then they developed a fake news detection model in their study. News data from India was used in this model. The news is categorized into Crime, Treatment, Economy, Social, and Entertainment categories. All words in 2,773 news texts were vectorized with the term frequency-inverse document frequency (TF-IDF) algorithm. After the preprocessing and training stages, the highest success was measured with 87% accuracy in the news data tested with k-nearest neighbors (KNN), SVM, LR, and Naive Bayes algorithms [3].

Goel et al. [16] achieved 58% success in sentiment analysis using Twitter API with Sentiment140 dataset using Baseline, Naive Bayes Classifier, and support vector machine algorithms. It is predicted that the system improved by using the WordNet database will achieve more successful results in the sentiment analysis model.

Hou et al. [21] proposed a model for automatically identifying and classifying biomedical entities in text. For the named entity recognition task, they observed that a language model based on the BiLSTM-CRF architecture reduces false positives caused by words with multiple meanings and improves the performance of each subtask by sharing information between different entity names.

Dalkilic et al. [13] carried out a study on tagging proper names in Turkish texts as person, place, and organization entities. Data mining techniques were used for a data set of 30 different text files, taking 10 documents from the fields of politics, economics, and health. For Turkish, a rule-based method was developed using some grammatical rules of the language and tested on different types of documents, with

the highest success rate of 87% for place names. Although the lowest success rate is for person names, there is an 80% success rate.

Nemes & Kiss [34] used users' tweets about COVID-19 during the pandemic as a data set. The sentiment analysis performed in the study is enriched with information extraction and named entity recognition methods to obtain a more comprehensive result. The results obtained using the BERT method are compared with RNN, NLTK, and TextBlob algorithms. The results obtained in the study, which were enriched with various graphics, were analyzed over 500 tweets. H. B.

Patil and Patil [38] developed a statistical Entity Association Recognition model to classify entities in the Marathi language. A conditional random fields algorithm was used for this. The analysis of the FIRE-2010 data-set showed that the CRF algorithm had an F1-Score of 75%, and it was concluded that this language-specific information should be included in the algorithm.

Pavitha et al. [39] proposed a cosine similarity-based method for movie recommendation systems based on users' tendencies. In the study, a method based on movie reviews was proposed with SVM and Naive Bayes algorithms. As a result of the comparison of both algorithms, an accuracy rate of 98.33% for SVM and 97.33% for Naive Bayes was obtained.

AminiMotlagh et al. [7] analyzed sentiment analysis of Twitter data with SVM, K-nearest, Decision Tree and Naive Bayes machine learning algorithms. Among single classifiers and their combinations with ensemble methods, SVM achieved 3.53% and 7.41% of the performance on binomial and polynomial datasets, respectively. Although ensemble methods do not perform better than single methods, they can reduce the memorization problem, variance and generalization error of learning models. In another study, a new model for the use of large language models in vulnerability detection was proposed using graph structure and learning for context. An experimental study was conducted on three vulnerability detection datasets to evaluate its efficiency. It was observed that this study produced better results than the top 6 vulnerability detection systems. In terms of F1 scores, an improvement of 28% was found on the 3 data sets tested [28].

In general, when the literature is evaluated it is found that social media, websites and corporate data are used for text classification in natural-language processing studies. Additionally, in classification studies, the NER method has often been used for entity prediction in data using statistical approaches. In general, machine learning algorithms have been applied to pre-labeled NER data and the results are discussed. As a result of the reviews, there is no previous study that uses the NER method in preprocessing stages. Although there are studies on the problem of classification of news texts, there is no similar study and the same data set in the literature that uses the proposed model in our study. For this reason, the models used in various text analysis processes in the literature for the dataset in our study and their results are given in Table 1.

Table 1
Text analysis studies with news dataset

References	Data Set	Model	Performance
[33]	News Data Set [32]	CNN &NN	%89 F1-Score
[25]	News Data Set [32]	WSTC	%75 F1-Score
[32]	News Data Set [32]	Exploratory Data Analysis	–
[48]	News Data Set [32]	Topic Modeling- LSTM	0.923-R
[2]	News Data Set [32]	Classification-Naive Bayes	%90-F1-Score

3. System architecture

In this section of the study, the model architecture, the stages and used data set are explained. After stop words extraction, lemmatization, and stemming, the data set of news texts was passed through the NER filtering module and trained with machine learning in nine different algorithms, deep-learning algorithm LSTM (long short-term memory) architecture and BERT (Bidirectional Encoder Representations from Transformers) language model, and the best classification results were tried to be obtained with different parameters.

3.1. Dataset

Data collection is the first stage of this study. Between 2012 and 2018, 50,000 news texts were used from the HuffPost news site in three categories with the most consistent distribution out of 200,000 texts [32]. The data set contains columns such as category, title, authors, news text, summary, and date. Figure 1 shows filtered data set distribution.

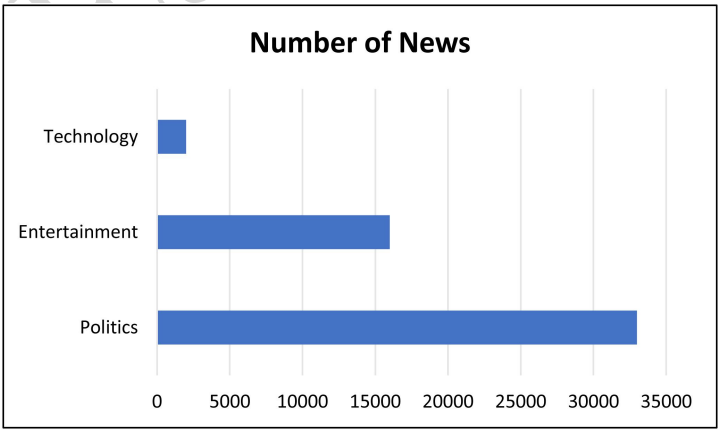

Figure 1. Filtered data set distribution

3.2. Data cleaning

Data cleaning consists of removing noisy data and extracting the data that are needed for analysis. The importance of this analysis can also be determined with data visualization tools. Within the scope of this study on the dataset:

1. Punctuation marks have been removed.
2. Stopwords in the data set were extracted according to the NLTK library and the stopwords list that we created (yourselves, itself, whom, those, etc.).
3. Web addresses were cleaned.
4. News texts were separated into words.
5. Words were separated into their roots.

3.3. Data preprocessing

In recent years, natural language processing studies are mostly based on statistical approaches with machine learning and deep-learning algorithms. For the analysis of the resulting text dataset, the words are digitized and made ready for the learning algorithm. In these methods, known as word embedding, words are digitized and represented in a vector space.

There are 2 basic approaches known in the literature. The first of these approaches is the TFIDF method, which is a frequency-based approach that digitizes based on proximity to a word [14]. This scoring model, which is widely used in the process of word digitization, evaluates the frequency of occurrence of terms in the document. This calculation method, known as Term FrequencyInverse Document Frequency (TFIDF), is a statistically calculated weighting factor that indicates the importance of a term in a document.

The TFIDF value is the multiplication of TF (i.e. how many times a word occurs in a sample document) and IDF (i.e., the inverse of how many times a word appears in the entire sample set) [23]. The formula for TF and IDF are given in Equation 1 and Equation 2, according to [4].

$$tf\,(term, document) = \frac{f(term, document)}{\sum_{term\prime \in document} f(term', document)} \tag{1}$$

$$idf(term, allDocuments) = \ log\frac{N}{df(t)} \tag{2}$$

The performance of artificial neural-network models in digitizing textual data has led to new methods due to the limitations of classical word-scoring methods.

In this method, a neural network model was developed that uses one word to predict other words by looking at the data set [31]. The approach based on the

neural network model is known as the Word2Vec model. This model is created by training 1.6 billion words per day.

In this approach word vectors are constructed in 2 ways: the continue bag-of-words (CBOW) and the Skip-Gram model. The CBOW approach is a technique that allows one to vectorize the word in the middle of the sentence by looking at the words near it, and the Skip-Gram model is a technique that vectorizes the other words next to it according to the middle word [8]. The Word2Vec model training within the scope of the study determined the best result, according to the parameters in Table 1.

The min-count parameter is the minimum number of occurrences of the word in the corpus. Vector size is the size of the vector to be created for each word. Window is the number of neighboring words that will directly affect the vector computation of the target phrase. Workers parameter is the number of cores to be used for parallel processing. Sg means that the algorithm will be trained with the Skip-Gram method.

The BERT language model, revealed by Google in 2018, is one of the pretrained language models that has been successful in many NLP tasks recently. BERT is pretrained for large-scale text data, combining word representations and sentence representations in a large transformer [52]. It evaluates the sentence both from left to right and from right to left.

In this way, it plans to better extract the meaning and the relationships between the words, and this pays off in the results. When digitizing a word, the BERT Tokenizer structure captures differences such as polysemy better than other embedding methods. It captures contextual word embeddings and other forms of information resulting in more accurate feature representations; thus it provides a better model performance [30].

3.4. Proposed model

In this section of the paper a diagram is given for the proposed model. With a new approach for the words obtained after cleaning, tagging, and word embedding on the data set, it was thought that the analysis would be more successful by removing entities such as date, place, time, proper name, and city names, which were thought to have no effect on the meaning for text classification. Studies have shown that parameters such as entity type and number affect the classification success, and NER filtering is used during the preprocessing stages [35, 49].

Accordingly, after the preprocessing stages, SPACY [1] library was used to detect and extract the entities from the database. In the process of extracting entities from product comments, a ratio was determined between the number of words in the sentence and the words labeled as entities. This ratio was determined as a percentage by dividing the number of entities in the sentence by the number of words in the sentence, and this calculation method is shown in Equation 3.

$$FilteredNER = \frac{Total\ Entity\ Number\ in\ Sentence}{Total\ Token\ Number\ in\ Sentence} * 100 \tag{3}$$

The purpose of this equation is to determine the percentage ratio of the number of entities in the sentence to the total number of words in the sentence. If this ratio is greater than the threshold value, the sentence is not included in the text classification process. This word elimination is expected to improve training and testing performance. In this model, analysis was performed both by extracting entities and running the entire data set.

In this study, classification was performed with classical machine learning, deep learning, and language model. The frequency-based TFIDF method to digitize words in the machine learning model, the Word2Vec word embedding method in the deep-learning model, and the BERT Tokenizer word embedding method were used in the language model. In the classification problem with three different models, the neural network model was first used in addition to classic machine-learning algorithms.
For the machine learning model (Model-1), which is frequently used to classify news texts according to their categories, nine algorithms were used. The algorithms used in [5, 40] research obtained successful results in their studies. The names of the used algorithms are given in Table 2.

Table 2
Algorithms for Model-1

The Algorithms		
Random Forest	K-Nearest Neighbour	SGD Classifier
Support Vector Machine	Bernoulli Naive Bayes	MLP Classifier
Logistic Regression	Gaussian Naive Bayes	Decision Tree

In the proposed Model-2, an approach based on the LSTM algorithm is presented using the Word2Vec word embedding method. LSTM is an extension of the recurrent neural network (RNN) [20] as a long short-term memory network model. The LSTM presents memory cells consisting of several types of gate units, including forget gate, input gate, and output gate in each recurrent body. LSTM learns to discard previous information, refreshes the current input as a new memory, and generates state and output information after learning. The LSTM model can capture information about the meaning of words in context [26]. In this model, word representations for the classification problem are created with the neural network model and then trained with the LSTM model, and performance analysis is performed

In the last model (Model-3), the solution of the classification problem is presented by using the language model. Language models try to predict masked words in a given text. The masked data can be a single word or a sentence. The Bert (Bidirectional Encoder Representations from Transformers) [15] language model was trained with internet resources using the encoder network of the transformer architecture by the Google brain team. BERT, one of the pretrained language models, gets good results for text classification problems. It benefits from the attention mechanism, which is effective in detecting general features of related words in a sentence or document [12]. After the words were vectorized with BERT embedding, performance analysis was

performed by fine-tuning. The results obtained from the three models are listed in the findings section with success rates. Figure 2 shows the diagram of the study.

3.5. Model training

The Sklearn library was used to test the study with machine learning algorithms. In the data set of 50,000 lines of news texts, unnecessary fields were removed and the news texts were divided into training and test sets. The rates that gave the best results in the analyses were determined as 80% training and 20% test data. The thresholds specified in the NER filtering module are determined from the worst to the best result obtained as a result of the classification. Figure 2 shows the model's flowchart on text classification.

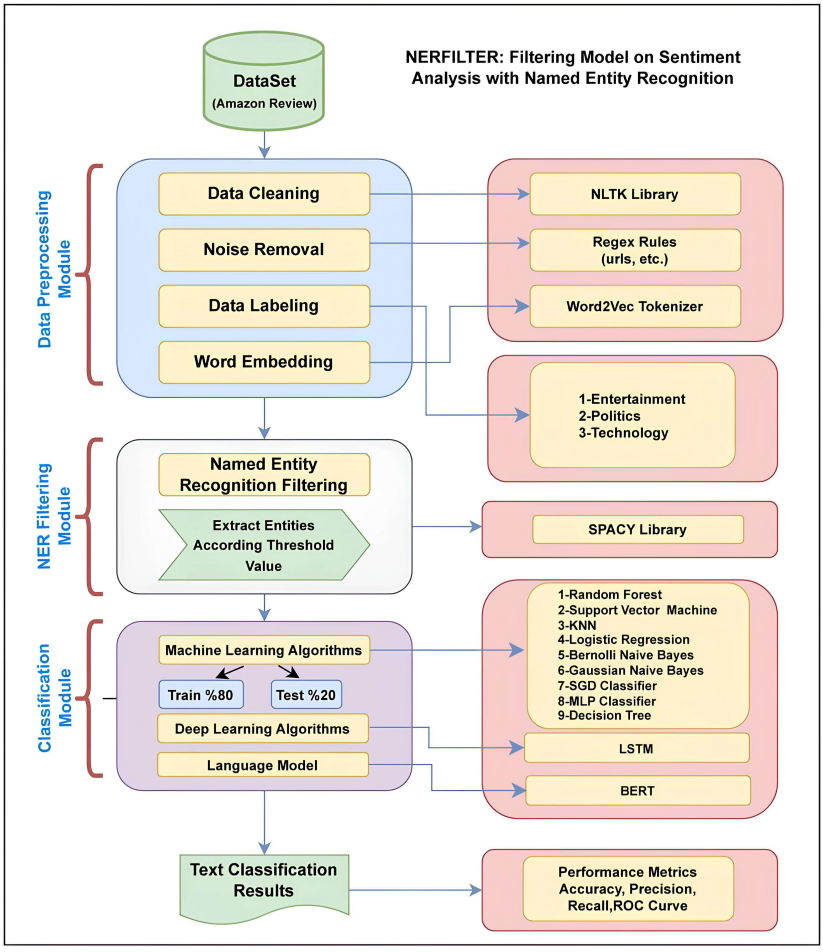

Figure 2. NER Filtering Model FlowChart on Text Classification

3.6. Model evaluation metrics

Model evaluation metrics play a key role in evaluating the accuracy and performance of a trained model. My literature review shows that researchers focused primarily on AUC, accuracy, precision, recall and F1-score. It is noteworthy that the AUC tends to differentiate between the classes of a data set. The higher the AUC the better the performance of a model that distinguishes between positive and negative classes. Furthermore, the confusion matrix measures the precision of all classification techniques. The confusion matrix has four distinct values: true positive (TP), false positive (FP), true negative (TN) and false negative (FN). A false positive of the confusion matrix is called Type 1 error, and false negative is called Type 2 error. Several approaches are used to evaluate a model's accuracy. For example, TP, TN, FP, and FN are the main determinants of the model's performance. These also determine the precision, recall, and f1-score that I used in following section [27].

4. Experimental results

In this study, a performance-enhancing model is proposed for news text classification. The effects of the entities identified in the data set on performance are measured by three sub-techniques in the model. The performance of the classification problem was analyzed using classic machine learning, deep learning, and language modeling. In the 50.000 lines of news text data set the effect of named entities on the classification success was analyzed with the determined threshold values.

Table 3
Number of Words Excluded from Analysis According to Thresholds

Context	Total Word	Total NER	Threshold
50878	377483	39376	10%
		40164	5%
		0	0%

The descriptions of the identified entity names and their distribution according to categories are given in Table 4. After the entity names were detected according to the thresholds in Table 5, they were removed from the data set and their effect on classification was measured.

After the preprocessing stages‘ the data set was divided into 80% training and 20% test set. The words used for the analysis with machine learning algorithms were vectorized with the TfIdf method. The Word2Vec word embedding method was used for deep-learning analysis, and the LSTM algorithm was used for training. The parameters used are given in Tables 5 and 6.

Table 4
Descriptions of Entity Names and Their Distribution According to Thresholds

Named Entities	Explanation	Threshold 10%	Threshold 5%
PERSON	People, including fictional	11634	11885
CARDINAL	Figures do not fall into any other genre.	4105	4252
NORP	Nationalities, religious or political groups.	3828	3940
DATE	Dates or periods.	3802	3863
GPE	Cities, states, or countries	3065	3153
ORG	Companies, agencies, institutions, etc.	1706	1777
ORDINAL	"First," "second," etc.	1149	1196
TIME	Times smaller than a day.	763	769
MONEY	Monetary values, including unit.	132	132
PERCENT	Percentage, including "%".	92	92
QUANTITY	Measurements such as weight or distance.	40	41
LOC	Non-GPE locations, mountain ranges, bodies of water.	19	19
PRODUCT	Objects, vehicles, foods, etc. (not services).	15	17
FAC	Buildings, airports, highways, bridges, etc.	13	15
LAW	Named documents made into laws.	5	5
WORK_OF_ART	Titles of books, songs, etc.	2	2
EVENT	Named hurricanes, battles, wars, sports events, etc.	1	1
LANGUAGE	Any named language.	1	1

Table 5
Word Embedding Parameters in LSTM

Size	300
Window	13..5
Min_count	1
sg	0 (CBOW)
Epoch	30

Table 6
Training Parameters in LSTM

Activation function	RELU, Softmax
Loss function	Cross-Entropy
Optimization function	Adam
Number of texts	50878
Batch size	256
Epoch	10

The parameters that were given in Table 7 were used in the analysis using the language model.

Table 7
Training Parameters in BERT

Activation function	RELU, Softmax
Tokenizer	Distilbert-Base-Uncased
Loss function	Cross-Entropy
Optimization function	Adam
Number of texts	50878
Batch size	64
Epoch	1

4.1. Machine learning algorithms result

The results of the analysis using machine learning algorithms are given in Tables 8, 9, and 10.

Table 8
NER Analysis Results without Filtering

Category	Algorithm	Precision	Recall	F1 Score	Support	Accuracy	Auc
Entertainment	**SVM**	**0.92**	**0.88**	**0.90**	**4834**	**0.93**	**0.97**
Politics		**0.93**	**0.97**	**0.95**	**9770**		
Tech		**0.85**	**0.61**	**0.71**	**660**		
Entertainment	MLPClassifier	0.91	0.86	0.89	4793	0.92	0.97
Politics		0.93	0.96	0.94	9878		
Tech		0.69	0.67	0.68	593		
Entertainment	BernolliNB	0.91	0.88	0.90	4833	0.92	0.97
Politics		0.93	0.96	0.94	9763		
Tech		0.82	0.53	0.65	668		
Entertainment	MultinominalNB	0.91	0.87	0.89	4754	0.91	0.97
Politics		0.91	0.97	0.94	9851		
Tech		0.94	0.31	0.46	659		
Entertainment	RandomForest	0.89	0.85	0.87	4880	0.90	0.96
Politics		0.91	0.95	0.93	9746		
Tech		0.78	0.55	0.65	638		
Entertainment	Decision Tree	0.85	0.84	0.84	4847	0.88	0.87
Politics		0.91	0.93	0.92	9753		
Tech		0.70	0.57	0.63	664		
Entertainment	KNN	0.89	0.57	0.70	4826	0.81	0.85
Politics		0.80	0.97	0.87	9797		
Tech		0.80	0.25	0.38	641		

Table 9
Analysis Results According to 10% Threshold Value

Category	Algorithm	Precision	Recall	F1 Score	Support	Accuracy	Auc
Entertainment	**SVM**	**0.89**	**0.85**	**0.87**	**1936**	**0.92**	**0.97**
Politics		**0.93**	**0.96**	**0.95**	**4930**		
Tech		**0.84**	**0.52**	**0.65**	**208**		
Entertainment	MLPClassifier	0.90	0.84	0.87	1938	0.92	0.97
Politics		0.93	0.96	0.95	4949		
Tech		0.67	0.62	0.64	187		
Entertainment	MultinominalNB	0.89	0.82	0.85	1939	0.91	0.96
Politics		0.92	0.97	0.94	4952		
Tech		0.97	0.32	0.48	183		
Entertainment	RandomForest	0.88	0.80	0.84	1959	0.91	0.95
Politics		0.91	0.96	0.94	4930		
Tech		0.80	0.46	0.59	185		
Entertainment	BernolliNB	0.89	0.84	0.86	1974	0.91	0.96
Politics		0.92	0.97	0.94	4904		
Tech		0.95	0.35	0.51	196		
Entertainment	Decision Tree	0.86	0.78	0.82	1972	0.89	0.85
Politics		0.91	0.95	0.93	4917		
Tech		0.66	0.47	0.55	185		
Entertainment	KNN	0.88	0.67	0.77	1933	0.87	0.89
Politics		0.87	0.97	0.92	4960		
Tech		0.84	0.31	0.46	181		

Table 10
Analysis Results According to 5% Threshold Value

Category	Algorithm	Precision	Recall	F1 Score	Support	Accuracy	Auc
Entertainment	**SVM**	**0.90**	**0.86**	**0.88**	**2020**	**0.93**	**0.96**
Politics		**0.94**	**0.97**	**0.95**	**5073**		
Tech		**0.89**	**0.61**	**0.72**	**218**		
Entertainment	MultinominalNB	0.92	0.84	0.88	2009	0.92	0.97
Politics		0.92	0.98	0.95	5098		
Tech		0.98	0.31	0.47	204		
Entertainment	BernolliNB	0.88	0.85	0.86	1942	0.91	0.96
Politics		0.93	0.96	0.94	5150		
Tech		0.92	0.36	0.52	219		
Entertainment	RandomForest	0.89	0.81	0.84	2011	0.91	0.95
Politics		0.92	0.97	0.94	5088		
Tech		0.84	0.46	0.59	212		
Entertainment	MLPClassifier	0.89	0.83	0.86	1999	0.91	0.96
Politics		0.93	0.96	0.94	5119		
Tech		0.67	0.63	0.65	193		
Entertainment	Decision Tree	0.87	0.78	0.82	2037	0.89	0.86
Politics		0.90	0.95	0.93	5041		
Tech		0.74	0.52	0.61	233		
Entertainment	KNN	0.88	0.66	0.75	2007	0.87	0.90
Politics		0.86	0.97	0.91	5102		
Tech		0.85	0.37	0.52	202		

4.2. Deep learning results

The results of the analysis using the Deep Learning algorithm LSTM and the Word2Vec word embedding algorithm are given in Tables 11,12 and 13.

Table 11
NER Analysis of Results Without Filtering

Category	Algorithm	Precision	Recall	F1 Score	Support	Accuracy	Auc
Entertainment	LSTM	0.87	0.83	0.85	6429	0.84	0.92
Politics		0.82	0.94	0.88	7182		
Tech		0.89	0.26	0.41	1067		

Table 12
Analysis of Results According to 10% Threshold Value

Category	Algorithm	Precision	Recall	F1 Score	Support	Accuracy	Auc
Entertainment	LSTM	0.76	0.78	0.77	1988	0.86	0.90
Politics		0.90	0.92	0.91	4905		
Tech		0.64	0.10	0.17	181		

Table 13
Analysis of Results According to 5% Threshold Value

Category	Algorithm	Precision	Recall	F1 Score	Support	Accuracy	Auc
Entertainment	LSTM	0.80	0.78	0.79	1971	0.87	0.91
Politics		0.91	0.94	0.92	5146		
Tech		0.66	0.21	0.31	194		

4.3. Language model results

The results of the analysis using the BERT language model are given in Tables 14, 15 and 16.

Table 14
NER Analysis of Results Without Filtering

Category	Algorithm	Precision	Recall	F1 Score	Support	Accuracy	Auc
Entertainment	Bert	0.82	0.83	0.82	6434	0.80	0.91
Politics		0.78	0.88	0.83	7185		
Tech		0.93	0.11	0.20	1067		

Table 15
Analysis of Results According to 10% Threshold Value

Category	Algorithm	Precision	Recall	F1 Score	Support	Accuracy	Auc
Entertainment	Bert	0.86	0.57	0.68	1955	0.83	0.91
Politics		0.83	0.97	0.89	4907		
Tech		0.88	0.11	0.19	212		

Table 16
Analysis of Results According to 5% Threshold Value

Category	Algorithm	Precision	Recall	F1 Score	Support	Accuracy	Auc
Entertainment	Bert	0.68	0.82	0.74	2025	0.83	0.91
Politics		0.91	0.85	0.88	5106		
Tech		0.71	0.27	0.39	180		

5. Results and discussion

In this study, three different models were executed to classify the data set using NER Filtering. First, classic machine-learning algorithms, then LSTM, and finally the BERT language model were used to solve the classification problem. The results of the best performance analysis results are given in Table 17, loss graph and complexity matrix for the model executed with various techniques, and the parameters are given in Figures 3, 4, and 5. For all three approaches it was found that more successful results were obtained as the number of filtered entities increased. With the classic

machine-learning algorithms, support vector machine (SVM) with 5% NER filter value gives the best results in terms of analysis time, roc curve, and accuracy. Figure 3 shows Roc and the precision-recall curve.

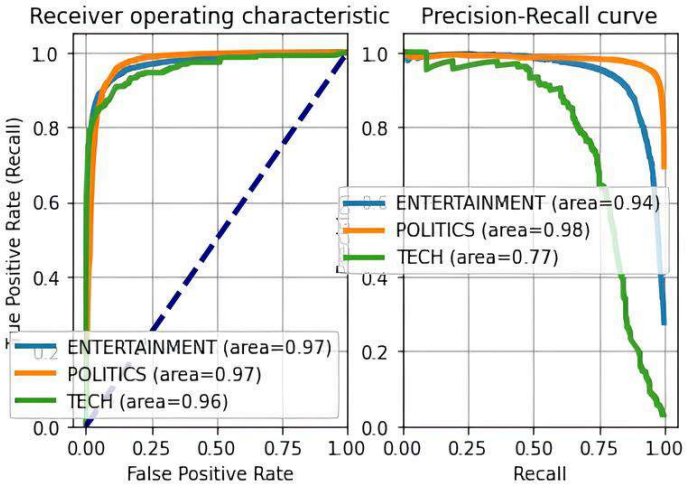

Figure 3. Roc and Precision-Recall Curve of SVM Algorithm with 5% NER Filter Value

The Word2Vec word embedding algorithm was used in the analysis with the deep-learning algorithm LSTM. The 5% NER filter value gives the best result in terms of analysis time, roc curve, and accuracy. Figure 4 shows the Roc and precision-recall curve for LSTM.

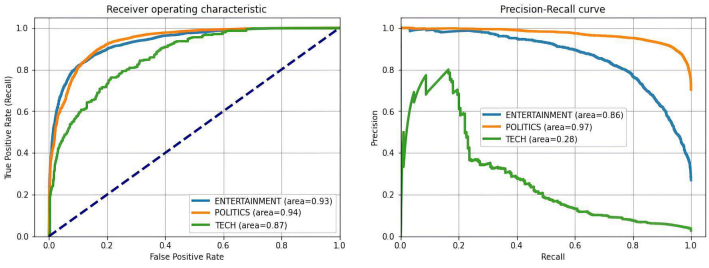

Figure 4. Roc and Precision-Recall Curve of LSTM Algorithm with 5% NER Filter Value

In the analysis with language models the BERT algorithm was used. In terms of analysis time, roc curve, and accuracy, 10% and 5% NER filter values give the best results at similar rates. Figure 5 shows the Roc and precision-recall curve for the BERT language model.

It is seen that the achieved success is higher in the analysis with classic machine-learning algorithms. In the analysis with deep learning and language models, it is

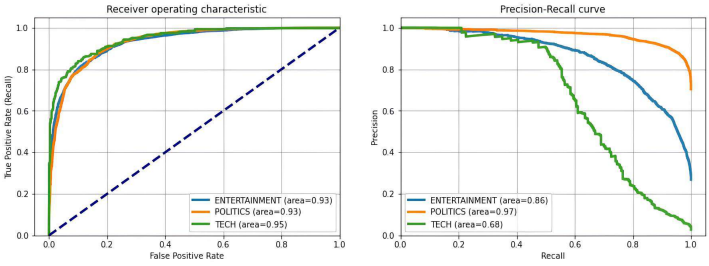

Figure 5. Roc and Precision-Recall Curve of the BERT Algorithm with 5% and 10% NER Filter

understood that the NER effect, which is the main purpose of the study, is better measured, although the model performance is low. The main reason for the low performance of these models is known to be the inadequate data set. Large data sets show more successful results in deep-learning analysis [17, 22].

Since classification problems are solved with a statistical approach in classic machine-learning studies, the semantic relationships between words are not taken into account, and the extraction of words defined as NER is not very effective. In the analysis, the success of some algorithms decreased after NER filtering and it shows that the performance of algorithms that offer a statistical approach decreases when the number of words decreases.

However, since the main factor in deep learning and language models is based on working with neural networks and the relationship of words to each other, it is observed that the success of the models increases as the number of filtered entities increases. Table 17 shows the effect of the algorithms used in the study before and after using NER.

Table 17
Impact of NER Filtering on Model Success

Algorithms	Acc with NER Filtering (%5)	Auc with NER Filtering (%5)	Acc Without NER Filtering	Auc Without NER Filtering	Prediction Time(s) with NER Filtering	Prediction Time(s) Without NER Filtering	Status
SVM	0.93	0.96	0.93	0.97	120	129	Neutral
MultinominalNB	**0.92**	**0.97**	0.91	0.97	**118**	**126**	**Positive**
BernolliNB	0.91	0.96	0.92	0.97	122	127	Negative
RandomForest	**0.91**	**0.95**	0.90	0.96	**117**	**122**	**Positive**
MLPClassifier	0.91	0.96	0.92	0.97	118	123	Negative
Decision Tree	**0.89**	**0.86**	**0.88**	**0.87**	**119**	**123**	**Positive**
KNN	**0.87**	**0.90**	**0.81**	**0.85**	**118**	**120**	**Positive**
LSTM	**0.87**	**0.91**	**0.84**	**0.92**	**123**	**130**	**Positive**
Bert	**0.83**	**0.91**	**0.80**	**0.91**	**126**	**133**	**Positive**

When the results are analyzed it is determined that although the SVM algorithm has high accuracy rates in classification success, the filtering has no effect (neutral) on the result obtained as a result of removing the words detected as entities. BernolliNB and MLPClassifier algorithms, on the other hand, showed a decrease in performance

with NER filtering, that is, a negative change was detected. In neural network-based deep learning and language models, it was found that as the number of filtered entities increased the success increased, and there was a positive change.

6. Conclusion and future works

The classification method is frequently used to generate meaningful data from large data sets. In our study, which aimed to increase the classification performance by extracting the entities detected in the data set, the data labeled in various categories were run with classic and modern algorithms, and significant results were obtained. The performance of the classification problem was further improved by identifying entity names in the preprocessing process. Although the Acc values of classic machine-learning algorithms were higher with NER filtering, it was observed that the model successes with deep-learning algorithms based on semantic integrity and the pretrained language model increased with filtering, but the model successes remained lower.

In this study the classification performance of news texts extracted from a news website was measured with a new approach. In the new approach, I observed in the experiments of this study and in the literature [35, 49] that removing all entities in preprocessing regardless of the number of entities in a sentence fails to correctly classify it. However, the success of the model was slightly improved by eliminating entities below a certain threshold value in the sentence. These thresholds were determined by trial and error. When the threshold was 0 all assets were included in the analysis. In this way I measured the loss of performance between the exclusion and inclusion of assets. At 5% and 10%, certain assets were eliminated according to the formula in Equation 3. In tests above 10% the model performance gets becomes worse and worse. The Spacy [1] library was used to detect the entities. In classic machine-learning algorithms, tfidf is used to separate words into vectors, while the CBOW algorithm is used in the Word2Vec model for the deep-learning phase and the BERT Tokenizer algorithm is used in languages model for word embedding.

When the results of the study are evaluated, it is observed that the filtering process contributes to the classification success in the analysis performed by extracting the words detected as NER from the dataset. Statistical machine learning algorithms showed the highest classification success. It has been observed that deep learning and the pretrained language model also increase the classification success with the filtering process and provide more semantically accurate classification. Although the results of the LSTM and BERT models are not too bad, their success is lower because the word similarities and convergence between neural networks cannot be done properly after the entities are removed. This is because LSTM and BERT run the Word2Vec word vector algorithm and the neural network model.

NER filtering increased success in six algorithms used in the study. However, as can be seen in Table 17, it negatively affected the model's success in two algorithms.

Although the classification success remained the same for SVM and LSTM, BERT benefited more from the filtering.

Although classification problems are known as categorization alone, improvements in the results of neural network-based models are very important in natural language processing studies such as text summarization, predicting the next word, correcting ambiguity, and text generation. In this context, it is observed that the results of our study are better than the achievements in NER studies using classical neural networks [53]. Although there are studies on the effect of preprocessing processes on classification performance in text classification studies, there is no study that includes entity names in the preprocessing process, and there is only one study that only performs filtering. For this reason, our study differs from existing studies in terms of model performance and novelty.

In future studies, based on this method, it is aimed to perform similar analyses on larger datasets with more powerful computer components and increase performance in terms of time and performance. The algorithms used in this study have been run with larger datasets in the literature; the model parameters were fine-tuned to adapt to larger datasets. It is also planned to extract and run disambiguating NER word types from this dataset in future studies.

Conflict of interest

No author associated with this paper has disclosed any potential or pertinent conflicts that may be perceived to have impending conflict with this work.

References

[1] spaCy · Industrial-strength Natural Language Processing in Python, https://spacy.io/, 2024. Accessed: Apr. 21, 2024.

[2] Ahmed J., Ahmed M.: Online News Classification Using Machine Learning Techniques, *IIUM Engineering Journal*, vol. 22(2), pp. 210–225, 2021. doi: 10.31436/iiumej.v22i2.1662. Accessed: Apr. 19, 2024.

[3] Ahmed J., Ahmed M.: Classification, Detection and Sentiment Analysis using Machine Learning over Next Generation Communication Platforms, *Microprocessors Microsyst*, p. 104795, 2023. doi: 10.1016/j.micpro.2023.104795. Accessed: Apr. 20, 2024.

[4] Aizawa A.: An information-theoretic perspective of tf-idf measures, *Information Processing & Management*, vol. 39(1), pp. 45–65, 2003. doi: 10.1016/s0306-4573(02)00021-3. Accessed: Apr. 20, 2024.

[5] Ali I., Mughal N., Khan Z.H., Ahmed J., Mujtaba G.: Resume Classification System using Natural Language Processing and Machine Learning Techniques, *Mehran University Research Journal of Engineering and Technology*, vol. 41(1), pp. 65–79, 2022. doi: 10.22581/muet1982.2201.07. Accessed: Apr. 20, 2024.

[6] Ali M., Tan G., Hussain A.: Bidirectional Recurrent Neural Network Approach for Arabic Named Entity Recognition, *Future Internet*, vol. 10(12), p. 123, 2018. doi: 10.3390/fi10120123. Accessed: Apr. 20, 2024.

[7] AminiMotlagh M., Shahhoseini H., Fatehi N.: A reliable sentiment analysis for classification of tweets in social networks, *Social Network Analysis and Mining*, vol. 13(1), 2022. doi: 10.1007/s13278-022-00998-2. Accessed: Apr. 20, 2024.

[8] Asudani D.S., Nagwani N.K., Singh P.: Impact of word embedding models on text analytics in deep learning environment: a review, *Artificial Intelligence Review*, 2023. doi: 10.1007/s10462-023-10419-1. Accessed: Apr. 19, 2024.

[9] Aydoğan M., Karci A.: Improving the accuracy using pre-trained word embeddings on deep neural networks for Turkish text classification, *Physica A: Statistical Mechanics and its Applications*, vol. 541, p. 123288, 2020. doi: 10.1016/j.physa.2019.123288. Accessed: Apr. 20, 2024.

[10] Bahçeci S.B.: Doğal Dıl İşleme'nin Alt Dalı: Varlık İsmi Tanıma, 2024. https://safaburakbahceci29.medium.com/doal-dil-ilemenin-alt-dali-varlik-ismi-tanima-eeb9f4551f06. Accessed: Apr. 20, 2024.

[11] Camacho-Collados J., Pilehvar M.T.: On the Role of Text Preprocessing in Neural Network Architectures: An Evaluation Study on Text Categorization and Sentiment Analysis, https://arxiv.org/abs/1707.01780, 2024. Accessed: Apr. 19, 2024.

[12] Chen X., Cong P., Lv S.: A Long-Text Classification Method of Chinese News Based on BERT and CNN, *IEEE Access*, vol. 10, pp. 34046–34057, 2022. doi: 10.1109/access.2022.3162614. Accessed: Apr. 19, 2024.

[13] Dalkilic F.E., Gelisli S., Diri B.: Named Entity Recognition from Turkish texts. In: *2010 IEEE 18th Signal Processing and Communications Applications Conference (SIU)*, IEEE, 2010. doi: 10.1109/siu.2010.5653553. Accessed: Apr. 20, 2024.

[14] Deerwester S., Dumais S., Furnas G., Landauer T., Harshman R.: Indexing by Latent Semantic Analysis, *Journal of the American Society for Information Science*, vol. 41(6), pp. 391–407, 1990. doi: 10.1002/(SICI)1097-4571(199009)41:6⟨391::AID-ASI1⟩3.0.CO;2-9.

[15] Devlin J., Chang M.W., Lee K., Toutanova K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, https://arxiv.org/abs/1810.04805, 2018. Accessed: Apr. 19, 2024.

[16] Goel A., Gautam J., Kumar S.: Real time sentiment analysis of tweets using Naive Bayes. In: *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, IEEE, 2016. doi: 10.1109/ngct.2016.7877424. Accessed: Apr. 20, 2024.

[17] Hang F., Xie L., Zhang Z., Guo W., Li H.: Research on the application of network security defence in database security services based on deep learning integrated with big data analytics, *Int J Intell Netw*, 2024. doi: 10.1016/j.ijin.2024.02.006. Accessed: Apr. 19, 2024.

[18] He B., Zhang J.: An Association Rule Mining Method Based on Named Entity Recognition and Text Classification, *Arabian Journal for Science and Engineering*, 2022. doi: 10.1007/s13369-022-06870-x. Accessed: Apr. 20, 2024.

[19] Hemati W., Mehler A.: LSTMVoter: chemical named entity recognition using a conglomerate of sequence labeling tools, *Journal of Cheminformatics*, vol. 11(1), 2019. doi: 10.1186/s13321-018-0327-2. Accessed: Apr. 20, 2024.

[20] Hochreiter S., Schmidhuber J.: Long Short-Term Memory, *Neural Computation*, vol. 9(8), pp. 1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735. Accessed: Apr. 19, 2024.

[21] Hou G., Jian Y., Zhao Q., Quan X., Zhang H.: Language model based on deep learning network for biomedical named entity recognition, *Methods*, 2024. doi: 10.1016/j.ymeth.2024.04.013. Accessed: Apr. 22, 2024.

[22] Koppe G., Meyer-Lindenberg A., Durstewitz D.: Deep learning for small and big data in psychiatry, *Neuropsychopharmacology*, vol. 46(1), pp. 176–190, 2020. doi: 10.1038/s41386-020-0767-z. Accessed: Apr. 19, 2024.

[23] Leelawat N., *et al.*: Twitter data sentiment analysis of tourism in Thailand during the COVID-19 pandemic using machine learning, *Heliyon*, vol. 8(10), p. e10894, 2022. doi: 10.1016/j.heliyon.2022.e10894. Accessed: Apr. 20, 2024.

[24] Li K., Kang C.: Deep feature extraction with tri-channel textual feature map for text classification, *Pattern Recognition Letters*, 2023. doi: 10.1016/j.patrec.2023.12.019. Accessed: Apr. 19, 2024.

[25] Li M., Zhu J., Yang X., Yang Y., Gao Q., Wang H.: CL-WSTC: Continual Learning for Weakly Supervised Text Classification on the Internet. In: *WWW '23: ACM Web Conference 2023*, ACM, 2023. doi: 10.1145/3543507.3583249. Accessed: Apr. 19, 2024.

[26] Liang M., Niu T.: Research on Text Classification Techniques Based on Improved TF-IDF Algorithm and LSTM Inputs, *Procedia Computer Science*, vol. 208, pp. 460–470, 2022. doi: 10.1016/j.procs.2022.10.064. Accessed: Apr. 19, 2024.

[27] Liu L., Shen J., Zhang M., Wang Z., Tang J.: Learning the joint representation of heterogeneous temporal events for clinical endpoint prediction. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018. doi: 10.1609/aaai.v32i1.11307.

[28] Lu G., Ju X., Chen X., Pei W., Cai Z.: GRACE: Empowering LLM-based software vulnerability detection with graph structure and in-context learning, *Journal of Systems and Software*, p. 112031, 2024. doi: 10.1016/j.jss.2024.112031. Accessed: Apr. 19, 2024.

[29] Mahmood M.: Stop Words and Named Entity Recognition (NER) Filtering for Airline Sentiment Text PreProcessing, 2024. https://blog.devgenius.io/stop-words-and-named-entity-recognition-ner-filtering-for-airline-sentiment-twitter-dataset-text-52c3643fcac9. Accessed: Apr. 22, 2024.

[30] McCormick C.: BERT Word Embeddings Tutorial · Chris McCormick, https://mccormickml.com/2019/05/14/BERT-word-embeddings-tutorial, 2019. Accessed: Apr. 20, 2024.

[31] Mikolov T., Chen K., Corrado G., Dean J.: Efficient Estimation of Word Representations in Vector Space, https://arxiv.org/abs/1301.3781, 2013. Accessed: Apr. 20, 2024.

[32] Misra R.: News Category Dataset, https://arxiv.org/abs/2209.11429, 2022. Accessed: Apr. 19, 2024.

[33] Misra R., Arora P.: Sarcasm detection using news headlines dataset, *AI Open*, vol. 4, pp. 13–18, 2023. doi: 10.1016/j.aiopen.2023.01.001. Accessed: Apr. 19, 2024.

[34] Nemes L., Kiss A.: Information Extraction and Named Entity Recognition Supported Social Media Sentiment Analysis during the COVID-19 Pandemic, *Applied Sciences*, vol. 11(22), p. 11017, 2021. doi: 10.3390/app112211017. Accessed: Apr. 20, 2024.

[35] P. P.: Text Preprocessing in Natural Language Processing (NLP), https://www.linkedin.com/pulse/text-preprocessing-natural-language-processing-nlp-prema-p-jurmc/, 2024. Accessed: Apr. 19, 2024.

[36] Pankaj, Pandey P., Muskan, Soni N.: Sentiment Analysis on Customer Feedback Data: Amazon Product Reviews. In: *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, pp. 320–322, 2019. doi: 10.1109/COMITCon.2019.8862258.

[37] Patil H.B., Patil A.S.: Evaluating the Effect of Preprocessing Tools for Marathi Text Retrieval, *Procedia Comput Sci*, vol. 233, pp. 902–908, 2024. doi: 10.1016/j.procs.2024.03.279. Accessed: Apr. 19, 2024.

[38] Patil N., Patil A., Pawar B.V.: Named Entity Recognition using Conditional Random Fields, *Procedia Computer Science*, vol. 167, pp. 1181–1188, 2020. doi: 10.1016/j.procs.2020.03.431. Accessed: Apr. 20, 2024.

[39] Pavitha N., *et al.*: Movie Recommendation and Sentiment Analysis Using Machine Learning, *Global Transitions Proceedings*, 2022. doi: 10.1016/j.gltp.2022.03.012. Accessed: Apr. 20, 2024.

[40] Perera N., Nguyen T.T.L., Dehmer M., Emmert-Streib F.: Comparison of Text Mining Models for Food and Dietary Constituent Named-Entity Recognition, *Machine Learning and Knowledge Extraction*, vol. 4(1), pp. 254–275, 2022. doi: 10.3390/make4010012. Accessed: Apr. 20, 2024.

[41] Salur M.U., Aydin I.: The Impact of Preprocessing on Classification Performance in Convolutional Neural Networks for Turkish Text. In: *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*, IEEE, 2018. doi: 10.1109/idap.2018.8620722. Accessed: Apr. 19, 2024.

[42] Schilling K.G., *et al.*: Influence of preprocessing, distortion correction and cardiac triggering on the quality of diffusion MR images of spinal cord, *Magnetic Resonance Imaging*, 2024. doi: 10.1016/j.mri.2024.01.008. Accessed: Apr. 19, 2024.

[43] Sebastiani F.: Machine learning in automated text categorization, *ACM Computing Surveys*, vol. 34(1), pp. 1–47, 2002. doi: 10.1145/505282.505283.

[44] Shelke R., Vanjale S.: Recursive LSTM for the Classification of Named Entity Recognition for Hindi Language, *Ingénierie des systèmes d'information*, vol. 27(4), pp. 679–684, 2022. doi: 10.18280/isi.270420. Accessed: Apr. 20, 2024.

[45] Siino M., Tinnirello I., Cascia M.L.: Is text preprocessing still worth the time? A comparative survey on the influence of popular preprocessing methods on Transformers and traditional classifiers, *Information Systems*, vol. 121, p. 102342, 2024. doi: 10.1016/j.is.2023.102342. Accessed: Apr. 19, 2024.

[46] Situmeang S.: Impact of Text Preprocessing on Named Entity Recognition Based on Conditional Random Field in Indonesian Text, *Mantik*, vol. 6(1), pp. 423–430, 2022.

[47] Suat-Rojas N., Gutierrez-Osorio C., Pedraza C.: Extraction and Analysis of Social Networks Data to Detect Traffic Accidents, *Information*, vol. 13(1), p. 26, 2022. doi: 10.3390/info13010026. Accessed: Apr. 20, 2024.

[48] Sun J., Gloor P.: "Towards Re-Inventing Psychohistory": Predicting the Popularity of Tomorrow's News from Yesterday's Twitter and News Feeds, *Journal of Systems Science and Systems Engineering*, vol. 29(6), pp. 823–839, 2020. doi: 10.1007/s11518-020-5470-4. Accessed: Apr. 19, 2024.

[49] Sun W., Liu S., Liu Y., Kong L., Jian Z.: Named Entity Recognition Networks Based on Syntactically Constrained Attention, *Applied Sciences*, vol. 13(6), p. 3993, 2023. doi: 10.3390/app13063993. Accessed: Apr. 19, 2024.

[50] Szczepanek R.: A Deep Learning Model of Spatial Distance and Named Entity Recognition (SD-NER) for Flood Mark Text Classification, *Water*, vol. 15(6), p. 1197, 2023. doi: 10.3390/w15061197. Accessed: Apr. 20, 2024.

[51] Uslu O., Özmen Akyol S.: Türkçe Haber Metinlerinin Makine Öğrenmesi Yöntemleri Kullanılarak Sınıflandırılması, *ESTUDAM Bilişim*, vol. 2(1), pp. 15–20, 2021.

[52] Wang Q., Liu P., Zhu Z., Yin H., Zhang Q., Zhang L.: A Text Abstraction Summary Model Based on BERT Word Embedding and Reinforcement Learning, *Applied Sciences*, vol. 9(21), p. 4701, 2019. doi: 10.3390/app9214701. Accessed: Apr. 19, 2024.

[53] Yaseen U., Langer S.: Neural Text Classification and Stacked Heterogeneous Embeddings for Named Entity Recognition in SMM4H 2021. In: *Proc. Sixth Social Media Mining Health (#SMM4H) Workshop Shared Task*, Assoc. Comput. Linguistics, Mexico City, Mexico, 2021. doi: 10.18653/v1/2021.smm4h-1.14. Accessed: Apr. 19, 2024.

Affiliations

Güncel Sarıman
Mugla Sitki Kocman University Information System Engineering Department
Mentese/Mugla, guncelsariman@mu.edu.tr

Received: 19.11.2024
Revised: 04.12.2024
Accepted: 15.08.2025