Computer Science • 23(3) 2022 https://doi.org/10.7494/csci.2022.23.3.4622

Zekeriya Anil Güven
Banu Diri
Tolgahan Çakaloğlu

IMPACT OF N-STAGE LATENT DIRICHLET ALLOCATION ON ANALYSIS OF HEADLINE CLASSIFICATION

Abstract *Data analysis becomes difficult when the amount of the data increases. More specifically, extracting meaningful insights from this vast amount of data and grouping it based on its shared features without human intervention requires advanced methodologies. There are topic-modeling methods that help overcome this problem in text analyses for downstream tasks (such as sentiment analysis, spam detection, and news classification). In this research, we benchmark several classifiers (namely, random forest, AdaBoost, naive Bayes, and logistic regression) using the classical latent Dirichlet allocation (LDA) and n-stage LDA topic-modeling methods for feature extraction in headline classification. We ran our experiments on three and five classes of publicly available Turkish and English datasets. We have demonstrated that, as a feature extractor, n-stage LDA obtains state-of-the-art performance for any downstream classifier. It should also be noted that random forest was the most successful algorithm for both datasets.*

Keywords topic modeling, headline classification, machine learning, text classification, latent Dirichlet allocation, data analysis

Citation Computer Science 23(3) 2022: 375–394

Copyright © 2022 Author(s). This is an open access publication, which can be used, distributed and reproduced in any medium according to the Creative Commons CC-BY 4.0 License.

1. Introduction

Information systems enable many companies to utilize large amounts of data in order to make their decisions. Data needs to be structured in order to use it as a knowledge base for decision-making; however, this is not a trivial process – especially in the text domain. Textual data with large dimensions is difficult to structure and analyze; therefore, text-mining methods have been developed. Text mining is the process of extracting insightful information from generally unstructured text; hence, next-generation online platforms can take advantage of different text-mining techniques like natural language processing (NLP), information retrieval, text classification, clustering, topic modeling, and text summarization. Also, text-mining techniques can be adopted in risk management, social media analysis, business intelligence, and fraud detection [10].

Topic modeling is one of the application areas of text mining. Topic models are machine-learning (ML) algorithms that are used to latent thematic structures from large corpora. Latent thematic structures are automatically derived from the statistical properties of the documents; hence, prior labeling or annotation is not required [28]. Topic modeling is used in many areas like sentiment analysis, spam detection, and headline classification. Latent Dirichlet allocation (LDA), latent semantic indexing, and the hierarchical Dirichlet process are examples of topic-modeling methods.

In this research, headline classification was performed by ML methods when using topic-modeling methods. As a feature extractor, LDA was chosen over other techniques since it provides a reliable representation of extensive text data in order to find different topics and their densities. In addition to the classical LDA (1-LDA), n-stage LDA (n-LDA) [13] was also used in the pipeline to show its impact. The n-LDA method reduces the word count in a corpus, as it allows for the deletion of words that have less weight than the threshold value within the topics. Thus, more successful topic modeling can be done with fewer corpora as compared to 1-LDA. The success of the ML methods was investigated by showing the positive effect of n-LDA in this study for both English and Turkish datasets. The contributions of this research can be summarized as follows:

- The effect of the ML algorithms was studied when using topic models.
- The impact of the n-LDA model was analyzed and compared with the 1-LDA model for headline classification.
- The faster speed and more successful nature was shown for the developed n-LDA model as compared to the 1-LDA model. The language-agnostic manner in which the n-LDA models can be used was also proven.

The remainder of the article is structured as follows. The literature research on topic modeling and news headlines is explained under the title “Related Work” in Section 2. Section 3 describes the tools and datasets that were used for the data analysis

as well as the utilized feature-extraction and machine-learning methods. Section 4 specifies the outputs of the analysis of the LDA and ML methods on the datasets. Finally, the conclusions of this research are discussed.

2. Related work

In the literature research, there are many studies on the LDA topic model and news headlines. These studies are mentioned separately under two subheadings.

2.1. Topic modeling

García-Pablos et al. [11] described W2LDA, an unsupervised system based on LDA. This system performs aspect category classification, aspect-term and opinion-word separation, and sentiment polarity classification along with some other unsupervised methods and minimum configuration steps for any domain and language. The authors evaluated W2VLDA using customer reviews and compared it with other LDA-based methods. They also evaluated the performance using the aspect-based sentiment-analysis subset of the multilingual SemEval 2016 Task 5 dataset.

Akhtar et al. [1] aimed to summarize and analyze user reviews of hotels on the TripAdvisor website. They applied NLP techniques to reveal some important information. They categorized customer reviews and then used topic-modeling techniques to reveal hidden issues on classified data. They applied the LDA topic-modeling technique to identify hidden information and aspects and subsequently conducted sentiment analysis on classified sentences and summarizations.

Bastani et al. [3] proposed an approach that was based on LDA to analyze customer complaints from the Consumer Financial Protection Bureau (CFPB). The proposed LDA extracted confidential topics from consumer complaints, assigning a probabilistic mix of topics to consumer complaints. Their approach aimed to extract latent topics in the CFPB complaints and examine their associated trends over time.

Wang et al. [29] proposed a new framework for applying online product reviews for analyzing customer preferences for two competitive products. They extracted the critical topics of online reviews for two specific competitive products via LDA. They also used topic-difference analysis to show the unique topics of the two products. They identified the strengths and weaknesses of both products for their competition.

Du et al. [9] focused on tracking and extracting the hot topic of microblog posts. They proposed an improved topic-extraction model that was based on LDA (MF-LDA) to extract hot topics from microblog posts. They developed a hot-topic life-cycle model (HTLCM) to see the evolutionary trends of the topics. HTLCM identified whether or not an issue was a candidate hot topic. They proposed a hot topic-tracking algorithm that combined MF-LDA and HTLCM for measuring their success. Their algorithm has been more successful than LDA.

Xing et al. [31] used LDA to identify underlying patterns in the language of students who were writing scientific arguments about a complex scientific phenomenon;

namely, the Albedo Effect. LDA was performed to electronically store the arguments that were written by the students regarding global temperatures. They showed that LDA could discover semantic patterns in specific science areas and could be used as a tool for content analysis.

Sommeria-Klein et al. [26] predicted that LDA could be used to separate environmental DNA samples into overlapping assemblages of co-formed taxa. They compared LDA's performance on the abundance and occurrence data and measured the robustness of the LDA decomposition by measuring its stability according to the initialization of the algorithm.

Li et al. [20] applied a harmonization procedure with the LDA model to combine two land cover crops with a 30-m resolution. The LDA model generated a harmonized class label for each pixel by statistically characterizing the land attributes from the class co-occurrences. They evaluated its success using the LDA model in conjunction with the more widely used error matrices or semantic similarity scores for compliance. They showed that using LDA with error matrices was more successful than using LDA.

Guven et al. [14] compared the proposed n-stage LDA method with other topic-modeling methods in a sentiment analysis of Turkish tweets. In the method, they suggested reducing the terms that were used in creating the model in LDA. The n-stage LDA method that they developed increased the accuracy value as compared to normal LDA.

2.2. Headline classification

Li et al. [19] proposed a text-based framework (namely, TBF) to estimate agricultural futures prices that took full advantage of information from online news to advance the forecasting of performance. They used a topic model called dependency decomposition sentence LDA to extract the influencing factors of agricultural futures from online headlines.

Kim et al. [17] measured the click-value of individual words in headlines; then, they proposed an LDA-based headline click-based topic model (HCTM) to identify words that could bring more clicks to the headlines. They showed that, by evaluating the topics and clicks together with HCTM, they could detect changes in the user interests in the topics.

Omidvar et al. [23] proposed four indicators to determine the quality of published headlines based on the number of clicks and their obtained lifetimes by a website log analysis. Then, they used a deep-learning model that could predict the quality of unpublished headlines.

Liu et al. [21] developed a news frame-detection approach and fine-tuned the BERT model to achieve multi-class classifications of news article titles. Their training and test evaluation used a new dataset of headlines that were related to gun violence in the United States. They compared their approach with previous methods and were more successful in automatic news frame detection.

Seifollahi and Shajari [25] identified essential words and appropriate senses within a somewhat textual headline content training. Their proposed word sense disambiguation method increased the accuracy that could be achieved when integrated with a system that predicts the directional movement of the EUR/USD exchange rate by the sentiment analysis of headlines.

Atzeni et al. [2] proposed an approach by using the contents in the Semantic Web domain to estimate real-valued sentiment scores. They evaluated their approaches on two different datasets that consisted of microblog messages and financial news headlines. They tested their approach by using feature sets that included lexical and semantic features as well as a combination of both.

Lu et al. [22] proposed an approach for headline classification based on multiple representational mixed models with ensemble and attention learning. They combined the character level by modeling it into the word level feature of the headlines and strengthened the semantic representation by reducing the effect of the word segmentation. They also used an attention mechanism to emphasize characters or words.

3. Methodology

The datasets and tools that were used are described in this section. Next, the feature-extraction methods and ML algorithms are explained. The feature-extraction methods were LDA and n-stage LDA in this research. Random forest (RF), AdaBoost (ADB), naive Bayes (NB), and logistic regression (LR) are described for the ML algorithms. The methodology of this study is shown in Figure 1.

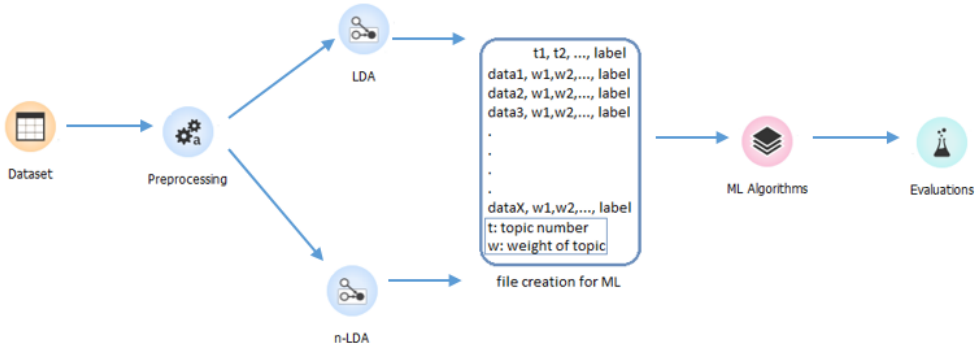

Figure 1. Methodology of this study

The topics are extracted by giving preprocessed datasets to the LDA methods. With the obtained topics, a file is created for the ML algorithms. For each headline, the attributes are created by calculating the weight values of each topic, and the label of the news is added in this file. Then, the headline classification is performed by the ML algorithms with this file.

3.1. Dataset description

The Turkish dataset[1] consisted of headlines from sites such as Mynet and Milliyet (see Fig. 2a). There were five news classes in the dataset, including economy, magazine, politics, sports, and life labels. There were 600 headlines for each news class. The dataset was trained and tested with three and five classes. One thousand eight hundred headlines were used for three classes (economy, life, sports), and three thousand headlines were used for five classes.

The English dataset[2] contained headlines from the UCI news and sports websites (see Fig. 2b). The dataset consisted of five classes, including business, entertainment, medicine, science&technology, and sports labels. There were 1000 headlines for each class. Two different datasets were used as three (entertainment, medicine, science&technology) and five classes during the training and test phases.

a) b)

Figure 2. Word clouds for datasets: a) Turkish dataset; b) English dataset

3.2. Tool description

Tools were used to implement the stemming operations and measure the success of the ML algorithms for the data that was obtained with the LDA models.

Stemming Library: While Zemberek[3] was performed for the Turkish dataset, the PorterStemmer[4] library was performed for English during the stemming process. Zemberek is an open-source Turkish NLP library, and it was completely developed in Java. PorterStemmer is a word-analysis tool that is installed with the NLTK library. It performs morphological analyses of words in English and determines the stem of each word.

[1] https://www.kaggle.com/anil1055/turkish-headlines-dataset
[2] https://www.kaggle.com/anil1055/english-headlines-dataset
[3] http://www.java2s.com/Code/Jar/z/Downloadzemberektr21jar.htm
[4] https://www.nltk.org/_modules/nltk/stem/porter.html

Orange Data Mining Tool: Orange[5] is open-source and free distributed data-analysis and ML software. This software contains libraries and components for data visualization, data analysis, etc.; it features flexibility and fast-working features. Components such as data preprocessing, feature scoring, feature filtering, modeling, model evaluation, and knowledge-discovery techniques are included in this software.

3.3. Feature-extraction methods

3.3.1. Latent Dirichlet allocation

LDA is a probabilistic model that uses Bayes on documents for topic modeling; it is an unsupervised clustering algorithm that automatically defines topics between text documents. Each topic indicates a multinomial distribution over the vocabulary [15]. LDA assumes that (1) a text document consists of various topics, (2) each topic consists of many words, and (3) the probability of each topic appearing in a particular text document is calculated. The topics are defined in large text documents by examining the rate of words that appear in a particular text document and an entire corpus [31]. With LDA, the terms in a document that are set to form a vocabulary are then performed to explore the hidden topics. Documents appear as a mix of topics with a probability distribution according to terms; then, each document is observed as a probability distribution over a set of topics [16].

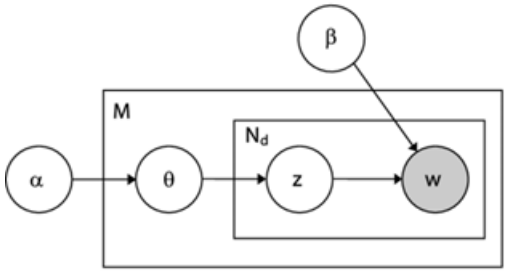

Figure 3. Structure of LDA [4]

The structure of LDA is shown in Figure 3 – the nodes represent random variables, and possible connections between nodes are represented by using edges. In Figure 3, α is the topic distribution per document, β is the word distribution per topic, θ is the topic distribution of a given document, z is the topic that is assigned to each word, and w indicates the observed word. The α and β parameters are sampled once a model is started, while θ parameter is sampled for each document [4].

It is crucial to determine an optimum topic number for LDA. The perplexity value does not express the semantic coherence of a topic [18]; hence, the coherence value is used to determine the topic number in this study. The topic coherence value

[5]https://orangedatamining.com/

scores a lone topic by calculating the degree of semantic similarity among the high-scoring words in the topic. This value helps to distinguish semantically interpretable topics [27].

3.3.2. n-stage latent Dirichlet allocation

Using the LDA algorithm, the n-stage LDA (n-LDA)[6] method was proposed in [12]. The proposed method was named n-stages because the n value is dynamic according to the size of the dataset that is used. This method aims to generate a faster and more successful topic model with fewer words in the dictionary. The processes of the n-stage method is shown in Figure 4, which shows that the threshold value calculation for each topic has an important role in the n-stage method.

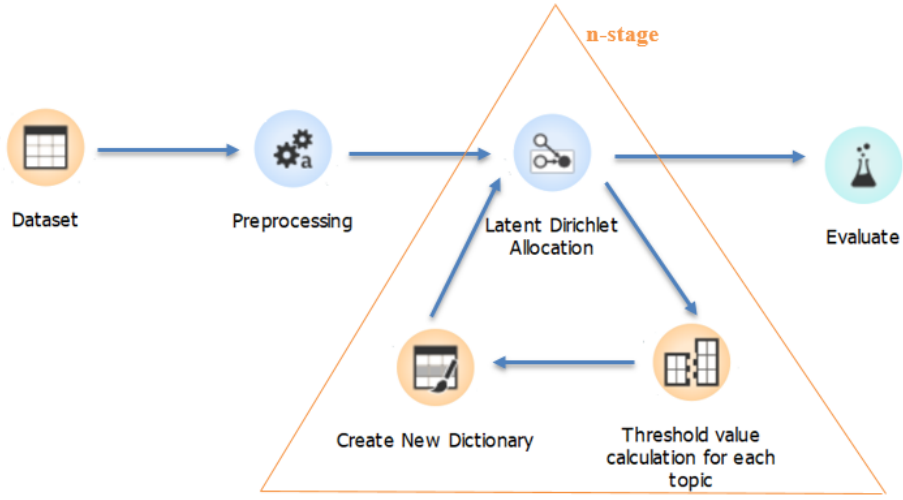

Figure 4. The Processes of n-stage method

The topics and the word-weight values of each topic are obtained with the LDA model. In the n-LDA method, the threshold value for each topic is calculated according to the weights of the words. The threshold value is calculated for each topic by proportioning the total weight of all of the words to the word count in the topic. Words with lower weights than the threshold value are removed for each topic; thus, a new dictionary is created with fewer words for the model. The LDA algorithm is remodeled with the new dictionary; the pseudocode of this method is shown in Figure 5.

[6]https://github.com/anil1055/n-stage_LDA

```
initialization gensim library;
tokenList ← your token list for your dataset;
topicNumber ← your topic number for your model;
model ← load your LDA model wtih gensim;
topics ← Words-weights info list for each topic in the model;
totalWords ← ∅ ;                                  // total word list for new LDA model
for topic in topics:
    duoInfo ← Seperate duo word-weight in the topic;
    totalWeight ← 0 ;                             // Total weight value for each topic
    wordList ← ∅ ;                                // word list for each topic
    weightList ← ∅ ;                              // weight list for each topic
    for i in range(len(duoInfo)):
        word ← Seperate word in duoInfo[i];
        weight ← Seperate word's weight value in duoInfo[i];
        if weight ≠ "0.0001":
            Add word to wordList;
            Add weight to weightList;
            totalWeight ← totalWeight + weight;
        else:
            break;
        thresholdWeight ← totalWeight/len(weightList);
        for j in range(len(weightList)):          /* Adding words to totalWords according to
         meanWeight of topic */
            if weightList[j] >= thresholdWeight:
                Add wordList[j] to totalWords
            else:
                break;
newTokenList ← remove tokens not on totalWords from tokenList;
Create new gensim corpus with newTokenList;
Create new gensim dictionary with newTokenList;
Create new LDA model with corpus, dictionary and topicNumber;
```

Figure 5. Pseudocode for n-stage LDA method

3.4. Machine-learning algorithms

3.4.1. Random Forest

RF is an ensemble-learning technique that ranks the importance of each estimator that is included in a model by generating multiple decision trees; it reduces the variance as compared to single decision trees [24]. Each node of a tree considers a different subset of randomly selected estimators; from this, the best estimator is selected and split on. Each tree is created using a different random bootstrap instance. The predictions for each variable are collected in all of the trees, and the mean square error of the out-of-bag predictions is calculated. The performance of each RF is evaluated according to the mean square error, and the percentage of the variance is explained [8].

3.4.2. Logistic Regression

LR is a method that is used in ML to create a model that can distinguish between instances from two or more classes. This method models the posterior probabilities of K classes through the linear functions of an input [6]; it starts with a training phase in which a prediction model is calculated based on the previously collected values for the predictive variables and corresponding results. This verification is carried out by evaluating the model in the specified common variables and comparing the output

with the known result. The model is considered to be successful when the model classification equals the result for most of the test data [5].

3.4.3. Naive Bayes

NB calculates the probability of a new sample that belongs to a class based on the assumption that all of the attributes that are given to a class are independent of each other. This assumption arises from the need to estimate the multivariate probabilities from a set of training data. In general practice, most combinations of attribute values are either unavailable or do not have enough numbers in the training data; thus, the direct estimation of each multivariate probability is not reliable. NB solves this problem with the conditional independence assumption [7].

3.4.4. AdaBoost

A boosting algorithm is a classifier that combines the bootstrap and bagging methods in an effort to combine several weak classifiers into one robust one; the ADB algorithm is representative of such an algorithm. Unlike other boosting algorithms, the ADB algorithm is a type of recurrent algorithm that sets a learning pattern according to the errors that are returned by weak learners. The main idea of this algorithm is to combine the weak learners that are created during each iteration in order to develop a strong learner; hence, it is essential to know how weak students will be weighted and combined [30].

4. Experiments

Under this heading, the experiments in this study are explained step by step. The preprocesses that was applied to the datasets, the results of the analysis of the LDA models, and the success of the ML methods are examined separately under the following subheadings.

4.1. Data preparation

Before the datasets were given to the model, the preprocesses were necessary – removing the punctuation marks, converting all of the texts to lowercase letters, defining the Turkish characters (only for the Turkish dataset), and removing the stopwords and numbers were all performed during the preprocessing phase. Then, the dictionary was created with the stems of the words. As a result of this preprocessing, the volume of the used data was decreased. Figure 6 indicates an example of the English preprocesses; the impact of these preprocesses can be examined in the figure as follows:

- The text was converted to lowercase letters; thus, the same words that were written differently (upper and lower cases) could be evaluated as a single form.
- Since there were large number of unnecessary punctuation marks in the tokens, the punctuation marks were deleted.

- Stopwords such as conjunctions, pronouns, etc. that did not affect the classification were removed from the texts; thus, the volume of the used data was decreased.
- The same root words with different suffixes should be evaluated as single words; therefore, the stemming operation was applied to all of the terms in the datasets.

Figure 6. Example for English preprocessing

4.2. Analysis of LDA models

Before training the LDA model, it was important to determine the optimal number of topics; therefore, the coherence values were calculated first in order to determine the topic number for each dataset (6, 9, 12, ..., 30 for the dataset with three classes, and 10, 15, 20, ..., 50 for the dataset with five classes). The topic number with the highest coherence values was used for the LDA model. The coherence values and topic numbers of the datasets are shown in Table 1. According to this table, the number of topics that were found for the whole dataset were the same as in the n-LDA models.

Table 1
Coherence values and topic numbers of datasets

	Turkish dataset		English dataset	
Class	Coherence	Topic Number	Coherence	Topic Number
3	0.512	6	0.473	30
5	0.493	30	0.458	40

Then, the 1-LDA model was generated with the obtained coherence values. Using the word weights of the topics that were generated by the model, n-LDA was performed twice (2-LDA and 3-LDA). The word counts that were used to generate the models are shown in Table 2. The word count that was used in 1-LDA decreased considerably in the other stages. The running times of the models are also shown in Figure 7. Since the word counts decreased, the running times decreased as the stage numbers increased.

Table 2
Unique word counts of generated models

	Turkish dataset		English dataset	
LDA models	3 class	5 class	3 class	5 class
1-LDA	2052	3133	2878	5489
2-LDA	486	499	1316	1389
3-LDA	404	464	1153	1285

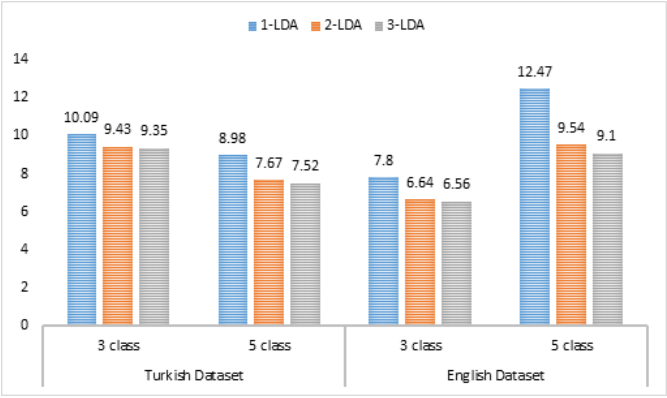

Figure 7. Running times of generated models

Label	Word	1-LDA	2-LDA	3-LDA	Label	Word	1-LDA	2-LDA	3-LDA
Economy	art	0.033	0.056	0.06	Business	company	0.041	0.067	0.169
Sports	takım	0.147	0.117	0.158	Entertainment	final	0.054	0.072	0.134
Politics	başkan	0.130	0.105	0.208	Sci&tech	release	0.155	0.035	0.22
Magazine	oyun	0.028	0.034	0.039	Sports	team	0.192	0.123	0.546
Life	kaza	0.132	0.185	0.295	Medicine	test	0.171	0.175	0.272

Figure 8. Example word weights for each model (Sci&tech: science&technology)

As a result of the application of 1-LDA and n-LDA, the topics with words and the weight values of the words were obtained. When the most suitable topic for any sentence is calculated in 1-LDA, the values of the topics can be close. But when the weight of the word that indicates the topic increases with n-LDA, it will be assigned to the appropriate topic with a more distinct difference. Example word weights for each model are shown in Figure 8. This figure shows that the weights of some words increased (takim, kaza, etc. for Turkish; final, test, etc. for English). With the

increases to these word weights, a more accurate classification can be made during the testing phase.

```
initialization gensim library;
topicNumber ← your topic number for your model;
model ← load your LDA model wtih gensim;
tokenList ← your token list in your model;
topics ← Words-weights info list for each topic in the model;
totalWeights ← Total weights value for each topic in all dataset;
for tokens in tokenList:
    topicWeights ← ∅ ;            // Total weight value to tokens for each topic
     for i in range(topicNumber):
        topicWeights[i] ← 0 ;              // All weight value to 0 before process
    for token in tokens:
        index ← 0;
        for topic in topics:
            for word,weight in topics: /* Searching token in each topic          */
                if token = word: /* Update related topic weight                  */
                    topicWeights[index] ← topicWeights[index] + weight;
            index ← index + 1;
    Add topicWeights to totalWeights list
labelList ← Label values in the dataset;
eachLabel ← Data count for each label;
texts ← "";
count ← 0 ;
index ← 0;
for weights in totalWeights: /* Writing text with label and weights           */
    for weight in weights:
        texts ← texts + ",";
    texts ← texts + labelList[index];
    count ← count + 1;
    if count = eachLabel:
        index ← index + 1;
        count ← 0 ;
Save texts to file;
```

Figure 9. Pseudocode for file creation

Finally, the weight of each headline was calculated for each topic according to the generated models. All of the topic weights were obtained by summing up the weight values of each topic that contained the terms in each news headline. A CSV file that contained the weights and class labels was obtained for each headline in the topics; that is, for each headline sentence, there were weight values that were equal to the topic number. The created file was used for the analysis of the ML algorithms. The pseudocode for the creation of the file is shown in Figure 9.

4.3. Analysis of ML algorithms

The success of the generated LDA models was analyzed using the RF, ADB, LR, and NB algorithms. The data that was obtained from the LDA models was made proper for these ML algorithms with the file creation that is shown in Figure 9. For these ML algorithms, the weight value of each topic was the attribute, and the tag of the data was the label; thus, the ML algorithms could be used for training with this file. The RF, ADB, LR, and NB algorithms were performed by the Orange data-mining tool[7].

[7]https://orangedatamining.com/

The evaluation criteria were determined as F1-measure values for the ML methods. The effects of the 1-LDA and n-LDA models were analyzed step by step for all of the datasets.

The results of the Turkish dataset with three classes are shown in Figure 10. While the stage number of the LDA model increased, the F1-measure value of the ML algorithms also increased. The AdaBoost algorithm achieved the highest F1-measure (with 93.45% in 3-LDA).

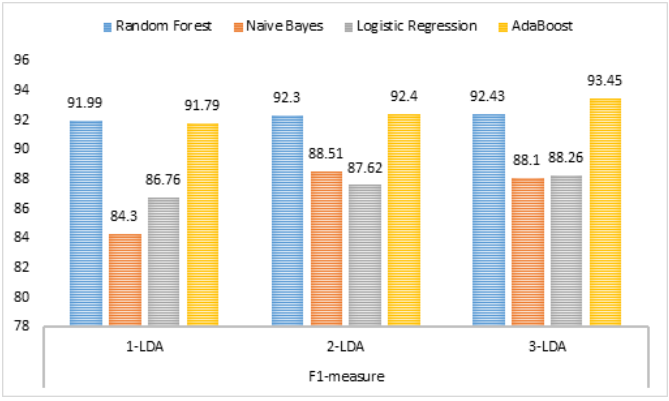

Figure 10. Results of three-class Turkish dataset

Figure 11 shows the F1-measure values of the ML algorithms in which the Turkish dataset with five classes was used. As a result of the increase in the stage number of the LDA model, the F1-measure of the ML algorithms increased. The highest F1-measure was obtained by the RF method (with 88.91% in 2-LDA).

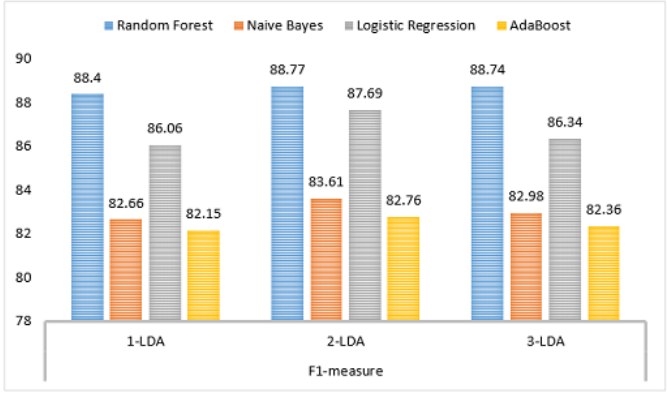

Figure 11. Results of five-class Turkish dataset

The F1 measure of the ML algorithms was analyzed for the English dataset. The F1-measure results for the English dataset with three classes are shown in Figure 12. With the application of the n-stage to the model, the F1-measure value increased. The most successful ML algorithm was RF (with 90.77% in 3-LDA).

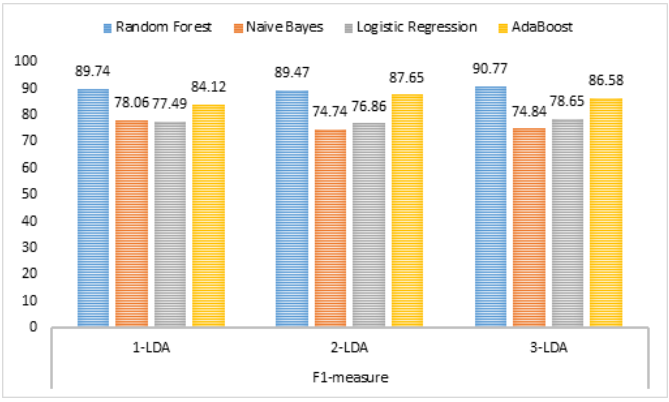

Figure 12. Results of three-class English dataset

The results of the English dataset with five classes are shown in Figure 13. When the algorithms were analyzed, the success of the ML algorithms grew with increasing stage numbers. The RF algorithm was the best method (with an 81% F1-measure in 3-LDA).

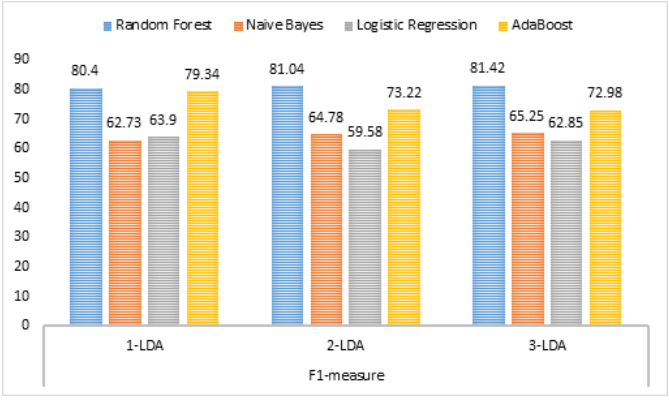

Figure 13. Results of five-class English dataset

The F1-measure and AUC values for all of the LDA models and ML methods are given in Table 3. The values of F1 and AUC were linearly proportional. The AUC value indicates the area under the ROC curve. The table indicates that the LDA

model with the lowest F1-measure and AUC values was 1-LDA. The F1-measure and AUC values increased with the n-stage LDA models.

Table 3
F1-measures and AUC values of datasets

Methods		Turkish dataset				English dataset			
		3 class		5 class		3 class		5 class	
LDA	ML	F1	AUC	F1	AUC	F1	AUC	F1	AUC
1-LDA	RF	91.99	0.98	88.45	0.97	89.74	0.97	80.4	0.95
	NB	84.3	0.95	82.66	0.97	78.06	0.91	62.73	0.87
	LR	86.76	0.97	86.06	0.97	77.49	0.91	63.9	0.89
	ADB	91.79	0.95	82.15	0.89	84.12	0.88	79.34	0.95
2-LDA	RF	92.3	**0.983**	**88.77**	**0.981**	89.47	0.96	81.04	0.95
	NB	88.51	0.97	83.61	0.97	74.74	0.89	64.78	0.89
	LR	87.62	0.97	87.70	0.98	76.86	0.92	59.58	0.86
	ADB	92.4	0.97	82.76	0.89	87.65	0.95	73.22	0.93
3-LDA	RF	92.43	0.98	88.75	0.97	**90.77**	**0.978**	**81.42**	**0.958**
	NB	88.1	0.97	82.98	0.96	74.84	0.91	65.25	0.89
	LR	88.26	0.97	86.34	0.97	78.65	0.93	62.85	0.87
	ADB	**93.45**	0.97	82.37	0.89	86.58	0.94	72.98	0.93

5. Discussion

In this study, the success of the n-LDA method when classifying news headlines was examined. The proposed n-LDA method with the classical LDA was compared with the running times and F1-measures of the machine-learning methods.

Figure 14 shows the changes of the running times and F1-measure values as the stage number of the LDA model increases. As the number of stages increases, the running time decreases as the word count in the dictionary shrinks. This is because the word count that is used is decreased at each stage, so the word weights are usually increased. In addition, the success of the F1-measure also increases as the number of stages of the LDA model increases. These situations can be seen in Figure 14.

The n-LDA models were also applied to datasets in two different languages. The F1-measure of ML algorithms was analyzed in both datasets, and the success of these algorithms increased. As a result, it was proven that n-LDA models can be used; hence, it is a language-agnostic model. As a result, the n-LDA model can be used in different languages such as Arabic, French, and Spanish.

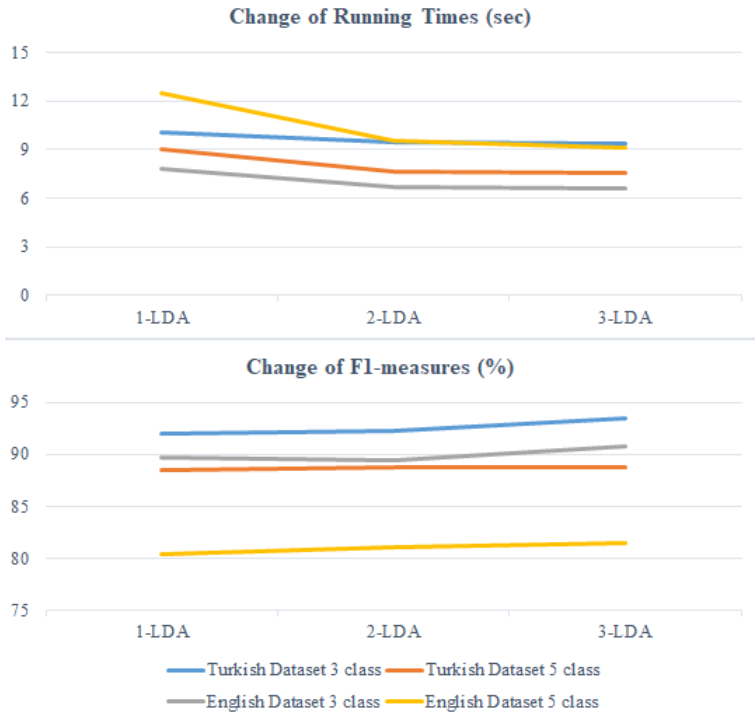

Figure 14. Changes of running times and F1-measure values according to LDA models for all datasets

6. Conclusion and future work

The 1-LDA and improved n-stage LDA models were analyzed for headline classification in this research. These approaches were compared using three- and five-class Turkish and English datasets that are available to the public. The 1-LDA model achieved the lowest success in the ML algorithms. While the n-stage LDA models were generated with fewer words, it was proven that the success of the ML algorithms increased with this model. The highest F1-measure values in both of the datasets were obtained when using the n-stage LDA models. RF was the most successful algorithm in all of the datasets except for the Turkish dataset with five classes (the most successful here was AdaBoost). In addition, it was proven that n-stage LDA models are generated faster than the 1-LDA model is.

For future work, evaluating the n-LDA models will be considered in languages other than Turkish and English. This is aimed at different studies such as spam detection, sarcasm detection, tag recommendation, document summarization, and sentiment analysis. In addition, n-LDA models can be used in different fields such as medical and geological data.

References

[1] Akhtar N., Zubair N., Kumar A., Ahmad T.: Aspect based Sentiment Oriented Summarization of Hotel Reviews, *Procedia Computer Science*, vol. 115, pp. 563–571, 2017. doi: 10.1016/j.procs.2017.09.115.

[2] Atzeni M., Dridi A., Reforgiato Recupero D.: Fine-Grained Sentiment Analysis on Financial Microblogs and News Headlines. In: *Semantic Web Challenges. SemWebEval 2017*, pp. 124–128, Communications in Computer and Information Science, vol. 769, Springer, Cham, 2017. doi: 10.1007/978-3-319-69146-6_11.

[3] Bastani K., Namavari H., Shaffer J.: Latent Dirichlet allocation (LDA) for topic modeling of the CFPB consumer complaints, *Expert Systems with Applications*, vol. 127, pp. 256–271, 2019. doi: 10.1016/j.eswa.2019.03.001.

[4] Blei D.M., Ng A.Y., Jordan M.I.: Latent Dirichlet allocation, *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[5] Bonte C., Vercauteren F.: Privacy-preserving logistic regression training, *BMC Medical Genomics*, vol. 11, 2018. doi: 10.1186/s12920-018-0398-y.

[6] Chen H., Gilad-Bachrach R., Han K., Huang Z., Jalali A., Laine K., Lauter K.: Logistic regression over encrypted data from fully homomorphic encryption, *BMC Medical Genomics*, vol. 11, 2018. doi: 10.1186/s12920-018-0397-z.

[7] Chen S., Webb G.I., Liu L., Ma X.: A novel selective naïve Bayes algorithm, *Knowledge-Based Systems*, vol. 192, 2020. doi: 10.1016/j.knosys.2019.105361.

[8] Darst B.F., Malecki K.C., Engelman C.D.: Using recursive feature elimination in random forest to account for correlated variables in high dimensional data, *BMC Genetics*, vol. 19, 2018. doi: 10.1186/s12863-018-0633-8.

[9] Du Y.J., Yi Y.T., Li X.Y., Chen X.L., Fan Y.Q., Su F.H.: Extracting and tracking hot topics of micro-blogs based on improved Latent Dirichlet Allocation, *Engineering Applications of Artificial Intelligence*, vol. 87, 2020. doi: 10.1016/j.engappai.2019.103279.

[10] Ferreira-Mello R., André M., Pinheiro A., Costa E., Romero C.: Text mining in education, *WIREs Data Mining and Knowledge Discovery*, vol. 9(6), 2019. doi: 10.1002/widm.1332.

[11] García-Pablos A., Cuadros M., Rigau G.: W2VLDA: Almost unsupervised system for Aspect Based Sentiment Analysis, *Expert Systems with Applications*, vol. 91, pp. 127–137, 2018. doi: 10.1016/j.eswa.2017.08.049.

[12] Guven Z.A., Diri B., Cakaloglu T.: n-stage Latent Dirichlet Allocation: A Novel Approach for LDA, *CoRR*, 2021. doi: 10.48550/arXiv.2110.08591.

[13] Güven Z.A., Diri B., Çakaloğlu T.: Emotion Detection with n-stage Latent Dirichlet Allocation for Turkish Tweets, *Academic Platform Journal of Engineering and Science*, vol. 7(3), pp. 467–472, 2019. doi: 10.21541/apjes.459447.

[14] Güven Z.A., Diri B., Çakaloğlu T.: Comparison of n-stage Latent Dirichlet Allocation versus other topic modeling methods for emotion analysis, *Journal of the Faculty of Engineering and Architecture of Gazi University*, vol. 35(4), pp. 2135–2146, 2020. doi: 10.17341/gazimmfd.556104.

[15] Hoffman M.D., Blei D.M., Bach F.: Online Learning for Latent Dirichlet Allocation. In: *NIPS'10: Proceedings of the 23rd International Conference on Neural Information Processing Systems – Volume 1*, pp. 856–864, 2010.

[16] Jelodar H., Wang Y., Yuan C., Feng X., Jiang X., Li Y., Zhao L.: Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey, *Multimedia Tools and Applications*, vol. 78, pp. 15169–15211, 2019. doi: 10.1007/s11042-018-6894-4.

[17] Kim J.H., Mantrach A., Jaimes A., Oh A.: How to Compete Online for News Audience: Modeling Words that Attract Clicks. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1645–1654, 2016. doi: 10.1145/2939672.2939873.

[18] Li C., Duan Y., Wang H., Zhang Z., Sun A., Ma Z.: Enhancing Topic Modeling for Short Texts with Auxiliary Word Embeddings, *ACM Transactions on Information Systems*, vol. 36(2), pp. 1–30, 2017. doi: 10.1145/3091108.

[19] Li J., Li G., Liu M., Zhu X., Wei L.: A novel text-based framework for forecasting agricultural futures using massive online news headlines, *International Journal of Forecasting*, vol. 38(1), pp. 35–50, 2020. doi: 10.1016/j.ijforecast.2020.02.002.

[20] Li Z., White J.C., Wulder M.A., Hermosilla T., Davidson A.M., Comber A.J.: Land cover harmonization using Latent Dirichlet Allocation, *International Journal of Geographical Information Science*, vol. 35(2), pp. 348–374, 2021. doi: 10.1080/13658816.2020.1796131.

[21] Liu S., Guo L., Mays K., Betke M., Wijaya D.T.: Detecting Frames in News Headlines and Its Application to Analyzing News Framing Trends Surrounding U.S. Gun Violence. In: *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pp. 504–514, 2019. doi: 10.18653/v1/k19-1047.

[22] Lu Z., Liu W., Zhou Y., Hu X., Wang B.: An Effective Approach for Chinese News Headline Classification Based on Multi-representation Mixed Model with Attention and Ensemble Learning. In: *Natural Language Processing and Chinese Computing. NLPCC 2017*, Lecture Notes in Computer Science, vol. 10619, pp. 339–350, Springer, Cham, 2018. doi: 10.1007/978-3-319-73618-1_29.

[23] Omidvar A., Pourmodheji H., An A., Edall G.: Learning to Determine the Quality of News Headlines. In: *Proceedings of the 12th International Conference on Agents and Artificial Intelligence – Volume 1: NLPinAI*, pp. 401–409, 2020. doi: 10.5220/0009367504010409.

[24] Probst P., Boulesteix A.L.: To Tune or Not to Tune the Number of Trees in Random Forest, *Journal of Machine Learning Research*, vol. 18, pp. 1–18, 2018.

[25] Seifollahi S., Shajari M.: Word sense disambiguation application in sentiment analysis of news headlines: an applied approach to FOREX market prediction, *Journal of Intelligent Information Systems*, vol. 52, pp. 57–83, 2019. doi: 10.1007/s10844-018-0504-9.

[26] Sommeria-Klein G., Zinger L., Coissac E., Iribar A., Schimann H., Taberlet P., Chave J.: Latent Dirichlet Allocation reveals spatial and taxonomic structure in a DNA-based census of soil biodiversity from a tropical forest, *Molecular Ecology Resources*, vol. 20(2), pp. 371–386, 2020. doi: 10.1111/1755-0998.13109.

[27] Stevens K., Kegelmeyer P., Andrzejewski D., Buttler D.: Exploring Topic Coherence over Many Models and Many Topics. In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 952—961, 2012. https://aclanthology.org/D12-1087.

[28] Syed S., Spruit M.: Full-Text or Abstract? Examining Topic Coherence Scores Using Latent Dirichlet Allocation. In: *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 165–174, 2017. doi: 10.1109/DSAA.2017.61.

[29] Wang W., Feng Y., Dai W.: Topic analysis of online reviews for two competitive products using latent Dirichlet allocation, *Electronic Commerce Research and Applications*, vol. 29, pp. 142–156, 2018. doi: 10.1016/j.elerap.2018.04.003.

[30] Xiao L., Dong Y., Dong Y.: An improved combination approach based on Adaboost algorithm for wind speed time series forecasting, *Energy Conversion and Management*, vol. 160, pp. 273–288, 2018. doi: 10.1016/j.enconman.2018.01.038.

[31] Xing W., Lee H.S., Shibani A.: Identifying patterns in students' scientific argumentation: content analysis through text mining using Latent Dirichlet Allocation, *Educational Technology Research and Development*, vol. 68, pp. 2185–2214, 2020. doi: 10.1007/s11423-020-09761-w.

Affiliations

Zekeriya Anil Güven
Ege University, Turkey, zekeriya.anil.guven@ege.edu.tr

Banu Diri
Yildiz Technical University, Turkey, diri@yildiz.edu.tr

Tolgahan Çakaloğlu
Walmart Global Tech, USA, jackalhan@gmail.com

Received: 30.12.2021
Revised: 06.06.2022
Accepted: 27.08.2022