Automatic Semantic Categorization of News Headlines using Ensemble Machine Learning: A Comparative Study

Raghad Bogery[1], Nora Al Babtain[2], Nida Aslam[3], Nada Alkabour[4], Yara Al Hashim[5], Irfan Ullah Khan[6]
Department of Computer Science
College of Computer Science and Information Technology
Imam Abdulrahman Bin Faisal University, Kingdom of Saudi Arabia

***Abstract*—Due to widespread availability of Internet there are a huge of sources that produce massive amounts of daily news. Moreover, the need for information by users has been increasing unprecedently, so it is critical that the news is automatically classified to permit users to access the required news instantly and effectively. One of the major problems with online news sets is the categorization of the vast number news and articles. In order to solve this problem, the machine learning model along with the Natural Language Processing (NLP) is widely used for automatic news classification to categorize topics of untracked news and individual opinion based on the user's prior interests. However, the existing studies mostly rely on NLP but uses huge documents to train the prediction model, thus it is hard to classify a short text without using semantics. Few studies focus on exploring classifying the news headlines using the semantics. Therefore, this paper attempts to use semantics and ensemble learning to improve the short text classification. The proposed methodology starts with preprocessing stage then applying feature engineering using word2vec with TF-IDF vectorizer. Afterwards, the classification model was developed with different classifier KNN, SVM, Naïve Bayes and Gradient boosting. The experimental results verify that Multinomial Naïve Bayes shows the best performance with an accuracy of 90.12% and recall 90%.**

Keywords—Natural language processing; feature engineering; word embedding; text classification; ensemble learning

I. Introduction

At present, due to low cost hand-held multimedia enabled devices along with the fast internet, huge amount of information is created and accessed daily. Internet is the main source of information and the integral part of individual's life. Due to the focus on mobility and the internet in the recent years and to reduce the paper waste, many news companies went online and changed the traditional way of printing newspapers and articles. Because of that there's a huge number of different articles in the news website databases. However, categorizing news article in its respective category manually is very difficult and time consuming. Automatic categorization of the news corpus will profit society in several ways. However, automatic categorization of the news headlines is a challenging task as the length of news varies. There is a need for an automated way to extract and access the news according to the user's interest. Hence, this paper aim to propose an automatic news headline categorization that is based on machine learning techniques.

NLP is mainly used to automatically categorize documents and speech by words count or frequency without considering the meaning behind the words. This method is useful for document or huge chunk of text categorizing. However, news headlines and descriptions are usually short, so such methods might not accurately categorize the articles in their respective category. Because of that, many researchers started exploring semantic classification instead of relying on single word meanings to achieve better accurate results.

In-order to achieve effective classification or clustering of news headlines, it is important to consider four methodologies that helps in the semantic analysis i.e. the background knowledge, word representation and feature vectorization technique, topic modeling, the similarity measure that might be used to assess the clustering algorithm. Table I summarizes these methodologies with a brief explanation and examples.

The remaining part of our study is organized as follow. Section 2 contains a literature review. Section 3 contains the description of the proposed methodology. Section 4 contains empirical studies that include dataset description, Section 5 contains the experimental setup and finally the last section contains the conclusion of our study.

TABLE. I. Methodologies to Enhance Clustering Algorithm [1] [2]

Method	Definition	Examples
Background knowledge	Background knowledge can help learning model to better understand the relationships, the context and the meaning of words.	Ontologies, WordNet, semantic networks, treasures, and taxonomies.
Topic Modeling	Models the topics of different documents.	Lantent Dirchlet Allocation algorithm (LDA), Lantent Semantic Indexing technique (LSI).
Word Representation	The words in NLP are usually represented in the vector space model, each vector represent a word against its occurrence in corpus. Vectors with single words are called Bag of Words (BOW).	BOW, term frequency, Tf-IDF, Word2Vec, GloVe.
Similarity Measures	Used to see wither two words have the same or the opposite meanings in the vector space.	Cosine similarity, Euclidean distance, Jacob similarity.

II. Literature Review

This section discusses the literature reviews of the most important studies related to our topic. These topics are feature engineering, such as word embedding and text summarization, clustering using different methods, and finally classification.

A. Feature Engineering

One of the important steps in NLP is applying feature engineering for the text dataset, first to preserve the context of text and secondly to reduce the vector's dimensionality of the text. The word embeddings and semantic network like WordNet can help in preserving the context which will be discussed in the following sections, while text summarization can be used to reduce the dimensionality of text.

a) Word embedding: As the BOW, term frequency, TF-IDF; all represents the words in vector space model for the learning algorithm. However, they don't preserve the context and the relationships of the words in the documents. Word embedding can solve this problem as it models the semantics and relationships of words in a corpus using a vector with low dimensional as compared to the dimensional size of vocabularies in a corpus [1]. Several studies have been made on comparative analysis of word embedding techniques in various domains i.e. biomedical, twitter elections [3]. Twitter is one of the biggest source of individual opinion, news media [4]. There are two main approaches for words embedding that are known for their efficiency and accuracy and they are Word2Vec and GloVe.

A comparison study has been made for bio-medical NLP using Wikipedia, biomedical publications from PubMed and Medlist [5]. In 2013 Mikolov et. al [6], model was proposed for word2vec which has two architectures to learn and represent the words in the vector space. The first architecture is the continuous bag of words (CBOW) and second the skip-gram architecture. They performed word analogy to test their model, the word analogy was based on the semantic and syntactic questions that were produced by the authors. The authors observed a higher accuracy for both semantics and syntactic questions when both dimensional size and the number of training vocabulary increased. The methods that were proposed have the advantage of low computation time, but both consider the context locally in a document without making the advantages of the occurrence in different documents.

To overcome the previously mentioned problem, Jeffery et al. [7] proposed GloVe Model which stands for Global Vectors. The word's vector in the GloVe model is represented by not only considering the word co-occurrence probability in one document, but also considers the ratio co-occurrence probability across the documents. They tested their model on different tasks and conducted a comparison between CBOW and GloVe model and other baselines. They used the same testing approach as Mikolov et al. The performance of the model of the task analogy was increasing with the number of dimensions. In their comparison, they showed that the proposed model outperforms both architectures of word2vec in word analogy in semantic and syntactic questions. A comparative study to compare various feature engineering mechanisms for news articles and twitter tweets [8] . Continuous bag of words and skip-gram word embedding method using Convolutional neural network classifier. The experiments were conducted using these two word-embedding models and without any feature engineering approach for real news article and tweets. For the news article the CBOW achieved the highest accuracy while for the tweets Skip-gram outperform. A comparative study has been made by Jang et al. [9] on news article and news on social media in Korean language. The news was downloaded from NAVER a Korean news site, while twitter API was used to download the news from twitter.

b) Text summarization: Chi et al. [10] proposed a summarization model named Sentence Selection with Semantic Representation (SSSR). Through learning semantic sentence representation and implementing appropriate selection methods. SSSR also has two main parts which are sentence selection strategy and the sentence representation learning. Sentence selection strategy is to select a sentence that can rebuild the original document with the minimum falsification.

While in the semantic representation of sentences before implementing the selection strategy. Sentences can be represented using two representation the weighted mean of words embedding (SSSR-w) and deep coding (SSSR-d). The word embeddings were based on word2vec model, each word embedding is weighted based on the TF-IDF. Their experiment was conducted on DUC2006 and DUC2007 datasets they used Recall-Oriented Understudy for Gisting Evaluation (ROUGE) to evaluate the text summary results. Both models produce good results compared to other baselines using the F-measure metric, though SSSR-d has outperformed SSSR-w.

B. Clustering

Text clustering is considered as a challenging task, in the following sections three methodologies will be discussed for text clustering, dependency graph clustering, word embeddings clustering, and WordNet and lexical chains clustering.

a) Clustering using dependency graph: Asmaa K., et al. [11] proposed a way for reducing the problem that occurs from clustering using the traditional methods and in fact increase the clustering accuracy by using a method called dependency graph. A dependency graph represents one document, where each node is associated with a word and can be used as meta-data for the document. While semantic relations between words can be captured by using edges that are between their corresponding nodes, every edge has term weight based on TF-IDF. Dependency graph will affect the clustering result despite the clustering algorithm that is being used, and to display this K-means clustering algorithm was used to cluster the dataset. Where the number of clusters was 20 and so the value of K is 20. The number of correctly cluster documents was 188 out of 200 when using the dependency graph, while it decreases to 173 without the dependency graph.

b) Clustering using word embeddings: In contrast, Juneja et al. [12], used word embeddings to improve text clustering results. They compared between the word embeddings algorithms. GloVe, CBOW, and skip-gram all of them have high dimensional space, however, GloVe has a higher dimensional space and time complexity. They used GloVe for text clustering since they were more concerned about accuracy rather than the time complexity. Their proposed methodology used T-SNE algorithm to reduce the dimensionality of the GloVe model for better understanding and visualizing the results of their work. In addition, it also helps in reducing the curse of dimensionality where the irrelevant words mask the relevant words. They also saved the words embedding in files to reduce the time complexity of the GloVe model and used k-means as the clustering algorithm. They also saved the words embedding in files to reduce the time complexity of the GloVe model and used k-means as the clustering algorithm. They tested their methodology on two datasets, one dataset showed an increase of error rate with the decrease of the number of dimensions, and it is due to the data loss. While the second dataset doesn't have a specific number of K and thus the result was acceptable.

c) Clustering using WordNet and lexical chains: Using word embedding isn't the only way to improve the semantic analysis of a corpus. Tingting Wei, et. al [13] used WordNet and lexical chains and a modified Word Sense Disambiguation (WSD) to propose a method that meaningfully cluster texts while reducing the text dimensions. They modified the WSD similarity measure by combining two methods to create a more accurate similarity measure which is Wu–Palmer measure based on the least common subsume (LCS) and Banerjee and Pedersen's measure based on mutual words in the word's definitions. After performing WSD using the modified similarity measure, and extracting core semantics using lexical chains, Tingting Wei, et al. performed clustering using Bisecting K-means by assigning K as the number of classes was previously known. They compared it against other methods without using lexical chains by the same clustering method. The used methods were Base (WSD is not performed while performing all basic preprocessing such as removing stop words), Disambiguated Concepts (WSD is performed as well as performing all basic preprocessing), Disambiguated Core Semantic (WSD is performed using lexical analysis as well as performing all basic preprocessing).

They have shown that disambiguated core semantic method that uses lexical chains produce the highest F1-measure and purity on three groups. These results prove that the proposed method not only produces purer clusters, but also decreases the computational cost by decreasing the text dimensions using lexical chains.

In the study WordNet was used to find the semantic relation between the words. The sense of the words has been found by first selecting the key word and then finding the related words by using the WordNet hierarchical semantic relations. For finding the sense of the word, not only the selected word but also the related words were considered i.e. hypernym, holonym etc. using WordNet. The experiments were performed on SENSEVAL-2 dataset and the achieved accuracy was 32%. WordNet and Senti-WordNet [14] was used to classify the news headlines based on semantic and sentiment. In the study [15], the multilayer model was used for Forex real market data news categorization, the model used the combination of WordNet and Senti-WordNet. Senti-WordNet was used to find out the polarity of the news i.e. positive or negative news for the market prediction and achieved the accuracy of 83.33%. The Nassirtoussi et al study was recently extended by Seifollahi et al. [16] by adding the WSD in the semantic analysis module in order to exploit the impact of WSD on results. The dataset was initially divided into two categories date and time. The interval of the news was 2 hours. The headlines were analyzed by using the proposed model to monitor the exchange rate to predict the P (increase in exchange rate) and N (decrease in exchange rate). The system outperforms the previous system in terms of accuracy and time.

C. Classification

Another task in NLP is text classification, in the following section different algorithms were used to for text classification.

a) Classification using Different classifiers: Vishwanath et al. [17] proposed an improved term graph model and conducted a comparison between the KNN, term graph algorithm model and Naïve Bayes. The term graph model was used to preserve the semantics of the words in the datasets by using a weight in a directed graph for frequently co-occurring words. In this model, documents are treated as transactions and uses frequent item set mining algorithms. On the other hand, The KNN uses the vector space model which is based on TF-IDF and similarity measures to see whether one document belong to a certain class or not. The Naïve Bayes assign a probability for terms that belong to a certain class, the document, in the end, is classified to one class by the summation of each term probability for a certain class. They trained KNN, term graph model and Naïve Bayes on the dataset and then compared between the three model results. Among the three algorithms, the KNN outperformed both the term graph model and Naïve Bayes; term graph model has higher and closer accuracy to KNN, while Naïve Bayes has the worst accuracy between all the models. The proposed term graph model showed an improved result compared to the other baseline term graph model. A study has been made on 130000 news article consists of 8 categories using Naive Bayes, Support Vector machine and ANN model [2]. Features were selected using chi-square and LASSO. LASSO enhanced the accuracy of Naive Bayes while SVM achieved highest accuracy with chi-square feature selection. Another study [18] was made to identify the fake news using deep semantic structural model and improved Recurrent Neural Network on twitter dataset with 99% accuracy. Experiments were conducted using individual DSSM, LSTM and the hybrid approach the combination of DSSM and LSTM. Three experiments were conducted i.e. first by dividing the data equally among training and testing, secondly by dividing into 80-20 and finally with 75-25 division. The hybrid model achieved highest outcome with 75-25 data sampling division for training and testing. Pambudi et al. [19] classified the Indonesian news into multi-class classification using Pseudo Nearest Neighbor (PNNR). The PNNR was initially proposed for the binary classification and was later extended for multiclass as well. Several proximity functions were used, and

Cosine proximity similarity measure produced the highest results as compared to Manhattan and Euclidian.

b) Classification using WordNet: The study aimed to produce a model to predict suicidal thoughts by collecting data from Twitter using Twittr4J and used Weka as a data mining tool [13]. This paper also implements its own algorithm that calculates the semantic similarity between the collected data depending on a semantic analysis resource using WordNet. They manually constructed a vocabulary related with suicide and then collected data from Twitter. After that they applied IB1, J48, CART, SMO and Naïve Bayes algorithms to perform the classification. Then they improved their results using semantic analysis based on WordNet. The precision of the algorithms is shown in Table II based on the precision of the algorithms that were used.

TABLE. II. RESULTS FROM SUICIDE PREDICTION MODELS

Algorithm	IB1	J48	CART	SMO	Naïve Bayes
Precision (tweets with risk of suicide)	71%	81.2%	83.1%	89.5%	87.5%
Precision (tweets without risk of suicide)	63%	75.4%	66.7%	70%	61%

Finally, in word embedding GloVe and word2vec both have their own strength and weakness, GloVe is designed for preserving the context in a large corpus with multiple documents while the word2vec is designed for preserving context in one document with multiple records. SSSR-w and SSSR-d both used in word representation in the SSSR model, while SSSR-d tends to have a better performance over SSSR-w. On the other hand, all the cited literature in clustering section preserve the context of words by using either a dependency graph, a semantic graph or a word embedding. While in the classification, they used a either TF-IDF or WordNet to represent the words and to preserve its semantics. As most of the traditional classification algorithms doesn't work with the word embeddings technique, we can conclude with this gap which is how to combine between the word embeddings and the classical classification algorithms while maintain a high accuracy.

III. Description of the Proposed Techniques

Our methodology comprises of several stages which are preprocessing, feature engineering using word2vec, classification using K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Multinomial Naïve Bayes classifiers and finally training the model using boosting classifier.

A. Preprocessing Techniques

The preprocessing stage is very important for the semantic analysis. The dataset used in our study contains two columns that might help in future news headlines prediction, those two columns were combined and cleaned. As it is important for our model to maintain the semantics, thus returning each word to its root is important e.g. play and playing is the same word. Therefore, there was a trade-off between using stemming and lemmatization to improve our model. Stemming is a technique that removes suffix, it is simple and uses less computational time however it might lead to over-stemming errors due to its simplicity. On the other hand, Lemmatization uses the relationship between words and depends on WordNet to return the root of the word. This indicates that Lemmatization ensures less error but take more computational time. Since over-stemming might give wrong results for our classifier, we preferred the lemmatization over the stemming.

B. Feature Engineering Using Word2Vec

As mentioned previously, Word2Vec is a word embedding technique that uses a shallow neural network to represent the words in the vector space based on their context. There are two approaches to this technique, continuous bag of words (CBOW) and the skip-gram, as explained previously. This feature engineering method was used because it preserves the words semantics while lowering the dimensionality by dropping words that appear less than min_count, which is a hypermeter of the Word2Vec model. It also has two more important hypermeters, the dimensionality size of the word vector and the maximum distance between the current and predicted word within a sentence. In this study, the words were mapped to its produced vectors into a dictionary which was pipelined with the classification model using a TF-IDF vectorizer. Both Word2Vec approaches were tested with the classifiers.

C. Classification KNN, SVM and Naïve Bayes classifiers

The following section discussed the different classifiers that were used which are the K-Nearest Neighbor, Support Vector Machine, and Naïve Bayes classifiers.

a) K-Nearest Neighbor (KNN) Classifier: K-Nearest Neighbor (KNN) is a supervised learning algorithm that classifies data based on the training data using a similarity measure. It is one of the simplest supervised learning algorithms. This classifier was chosen because, despite its simplicity, it performs well. Moreover, in contrast to eager learners such as Naïve Bayesian, it is a lazy learner that stores the training data for future predictions and doesn't generate rules from them, so it doesn't require prior knowledge. KNN works by searching for the nearest similar K neighbor points in the training data and count their majority voting to predict the unknown class. In other words, it simply matches the unknown class attributes with the training data attributes and looks for the closest match. Because of that, its training time is short since it simply stores the training data. On the other hand, the testing time in KNN is usually far longer than the training time because it needs to compute the K neighbor voting for every test data [1].

From the previous description, KNN relies heavily on its training data, any noise can influence the prediction. Furthermore, a huge amount of training data will take time to test. Finally, the K value is also very important since it defines how many neighbors the algorithm consider while classifying. Usually, K is an odd number to avoid evenly split voting. As mentioned, KNN computes how similar the neighbor points using a similarity measure (distance measures). This study will use the Euclidian distance measure. The Euclidian is defined in equation (1) [1].

$$d(X,Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \qquad (1)$$

Where X= (x_1, x_2,…, x_i) are the set of attributes for the first data and Y= (y_1, y_2,..., y_i) for the second data. The result of d(X,Y) coordinate is plotted and compared to its neighbors.

b) Support Vector Machine (SVM) Classifier: The Support Vector Machine classifiers is one of the best machine learning techniques and outperform in most of the cases. Support Vector Machine classifiers were first introduced by Corinna Cortes and Vladivir Vapnik in 1995 [20]. It is a learning algorithm that works for both classification and regression problems. This classifier was selected because of its high performance even though it has a huge computational time [15]. The goal of SVM is to find the optimal separating hyperplane that gives the maximum separation margin between the hyperplane and the nearest points of both classes. For the set of training data that are shown in (2):

$$\{(x^1, y^1), \ldots., (x^m, y^m), \quad x \in R^n, y \in \{1, -1\} \tag{2}$$

A hyperplane can be found to separates the two classes. A hyperplane is shown in equation (3):

$$\langle w, x\rangle + b = 0 \tag{3}$$

It can be said that the hyperplane is separating the classes efficiently if the distance between the nearest point and the hyperplane is maximum.

There are some parameters that affect the result of the SVM classifier. The first one is the Regularization (C) parameter, lower value of C turns a high error rate that is given for the training set and the hyperplane margin will be large which means a smaller decision function. On the other hand, a higher value of C turns low error rate that is given for the training set and the hyperplane margin will be small. The second parameter is the gamma, it defines how far the effect of a single training point reach. If the gamma is large that means the point that is close will be used for calculation while a small value of gamma means that points that are far will be used for calculation.

c) Naïve Bayes Classifier: Multinomial Naïve Bayes classifier has a version for the textual document classification based on word count. Multinomial naïve bays classifier or known as term frequency or raw term frequency tf, is an approach for characterizing text based on number of times a term t appears in document d as shown in equation (4).

$$tf\ (t, d) \tag{4}$$

A multinomial naïve Bayes is known to be simple to implement, but very efficient since it assumes that the features are mutually independent, which is why it is one of our choices in the set of classifiers to implement. In practice, usually the term frequency tf is normalized by dividing it over the document length (or the sum of the number of terms in the document) n_d as seen in equation (5).

$$normalized\ term\ frequency = \frac{tf(t,d)}{n_d} \tag{5}$$

Using the term frequency, we can estimate the maximum-likelihood from the training data to find the class-conditional probabilities, where equation 6 shows the calculations needed to find this estimation.

$$\rho(x_i|w_j) = \frac{\sum tf(x_i, d\epsilon w_j) + \alpha}{\sum N_{d\epsilon w_j} + \alpha \cdot V} \tag{6}$$

Where

x_i represents *A word from a particular sample in the feature vector* x

$\sum tf(x_i, d\epsilon w_j)$ represents *the total sum of the term frequencies of a specific word* x_i *from the document in the training sample d that belongs to the class* w_j.

α represents *Smoothing parameter*.

$\sum N_{d\epsilon w_j}$ represents *the total sum of all the term frequencies N in the training dataset d that belong to the class* w_j.

V represents *the vocabulary size that is in the training set*.

Then we can use the product of the likelihoods of individual words in the document to give us the class conditional probability of encountering a word x, as shown in equation 7.

$$\rho(x|w_j) = \rho(x_1|w_j) \cdot \rho(x_2|w_j) \cdot \rho(x_3|w_j) \cdot \ldots \cdot \rho(x_n|w_j) = \prod_{i=1}^{m} \rho(x_i|w_j) \tag{7}$$

d) Improving the Model Using Ensemble Learning: Ensemble learning is used to increase the classifier accuracy and reduce the variance and bias, it follows different approaches including bagging, boosting, stacking and voting. The basic idea behind the ensemble learning is using multiple classifiers to improve the model’s prediction. In this study gradient boosting classifier will be used to improve the model.

e) Model Improved by Gradient Boosting Classifier: The Gradient boosting classifier combine many weak classifiers, the number of classifiers indicates how many times the model will be trained. In each training phase the misclassified instances will be given higher weight to reclassify them correctly. This can be an advantage for the imbalanced classes in our dataset and will decrease the number of misclassified classes in each training iterations.

IV. Empirical Studies

This section gives a brief description of the dataset characteristics. Also, describes the experimental setup that have been done to produce the models using the different selected classifiers K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Multinomial Naïve Bayes classifiers and finally improving the model using boosting classifier. As well as the followed optimization strategy.

A. *Description of Dataset*

The learning of any model relies on the nature and condition of the data used. In our study News Category Dataset obtained from Kaggle website have been used. This dataset is about collected news headlines from the year 2012 to 2018 obtained from HuffPost. It contains 202,372 records and 6 attributes, namely category, headline, authors, link, short description, and date. The target is the category of the headlines, containing 41 classes as shown in the Fig. 1. After cleaning the data and dropping records with empty cells, the dataset contains 200,746 records. In our study 3 classes were used containing the most records, as shown in Table III.

TABLE. III. FILTERED DATASET CLASSES

Class	Records
Travel	9887
Style & Beauty	9649
Parenting	8677

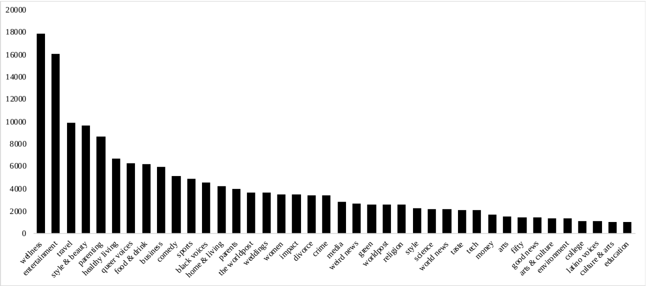


Fig. 1. News Category Dataset Class label distribution

V. EXPERIMENTAL SETUP

A. Dataset Cleaning

To clean this dataset, we started by removing the stop words, emojis and numbers. Next the missing values in the dataset were removed. The words were lemmatized using through stem.wordnet from nltk library. After that we applied word tokenization.

B. Feature Extraction using Word2Vec

After cleaning, tokenizing, and combining the data, the Word2Vecotrization was performed on the combined features using both CBOW and Skip-Gram Word2Vec methods. Both models were initialized with different measures as shown in Table IV. Based on the similarities between words using the cosine similarity measure, the CBOW3 model was chosen for the classification. For example, Fig. 2 shows the cosine similarity between the words 'health' and 'care'. Finally, the CBOW3 model's words were mapped to its vector as a dictionary to be used for classification.

C. Experimental Setup of K-Nearest Neighbor (KNN) Classifier

Firstly, a pipeline was used to combine the KNN classifier with the Tf-idf Vectorizer Word2Vec CBOW3 dictionary. Secondly, the dataset was split using the stratified technique to 70% for training data and 30% for testing data. Thirdly, a brute force grid search method of 5-fold was used on the training data to find the best odd K values between 7 and 15, resulting in 5 * 5 = 25 fits. Fourthly, the folds results were plotted as shown in Fig. 3. Lastly, the KNN classifier was validated using the unseen testing data using the best K parameter as found by the grid search previously.

D. Experimental Setup of Support Vector Machine (SVM) Classifier

For the SVM classifier the first step is to create a pipeline that combines the SVM classifier with the Tf-idf Vectorizer Word2Vec CBOW3 dictionary. The second step is to select several C and gamma parameters to be tested in the next step. The third step is to perform a 3-fold cross validation grid search on a data that was split to 70% testing a 30% training. Finally, the result of all the classifiers with the best fit parameter will computed.

E. Experimental Setup of Naïve Bayes Classifier

The multinomial Naïve Bayes has many forms such as multinomial, Gaussian, as well as Bernoulli. However, since our main goal is to preform multinomial Naïve Bayes on our dataset. The following steps have been used:

- Make sure that you have already split your data into testing and training
- Make an object from the multinomialNB() class
- Pipeline the object by the following code:
- MultiNB = Pipeline([('vect', TfidfVectorizer()), ('clf', MultinomialNB())])
- Then use It to train the data by fitting it to the object
- Define a predicted value to compare with: predicted = MultiNB.predict(X_test)
- And finally find its accuracy by finding the mean the values where the predicted = the test.

```
----------- CBOW ---------------

CBOW1 Cosine similarity between 'health' and 'care':0.938

CBOW2 Cosine similarity between 'health' and 'care':0.936

CBOW3 Cosine similarity between 'health' and 'care':0.917

----------- SkipGram (SG)---------------

SG1 Cosine similarity between 'health' and 'care': 0.835

SG2 Cosine similarity between 'health' and 'care': 0.785

SG3 Cosine similarity between 'health' and 'care': 0.795
```

Fig. 2. Cosine Similarity between the Words Example.

TABLE. IV. WORD2VEC MODELS

	Model name	Window distance	Minimum word count	Vector dimension
CBOW	CBOW1	3	50	100
	CBOW2	5	100	70
	CBOW3	7	150	50
Skip Gram	SkipGram1	3	50	100
	SkipGram2	5	100	70
	SkipGram3	7	150	50

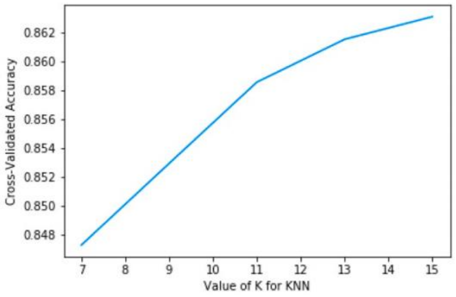


Fig. 3. 5-Fold Grid Search Results.

F. *Experimental Setup of Gradient Boosting Classifier*

The Gradient Boosting Classifier was used, which have different parameters that will help in tuning the boosting classifier. Those parameters are the base classifier, the number of estimators, the learning rate, the minimum sampling leaf (the number of samples to consider in the leaf) and split (the number of samples to consider when splitting the tree), the maximum depth for the classifying tree and finally the criteria of measuring the error in each iteration. All these parameters will also be tuned through grid search technique to select the best parameters that gives the optimal accuracy for the classifier. The base classifier was set on the default classifier (tree classifier) since the SVM and KNN doesn't work with classifier as a base classifier. Different parameters were tested to get the obtained results, which are listed in the following:

- Number of estimators: 50,100,200,300,400,500,600.
- Learning rate: 0.25, 0.1, 0.01,0.001.
- Maximum depth: 3,4,5,6,7,8.
- Minimum split sampling: 2,3,4,5,6,7,8.
- Minimum split leaf: 0.1,0.2,0.3,0.4.

Not all these parameters were included in the grid search, instead the best parameter from each was selected iteratively to reduce the computation time. The grid search was finally used with cross validation = 5 with the best selected parameters.

G. *Optimization Strategy*

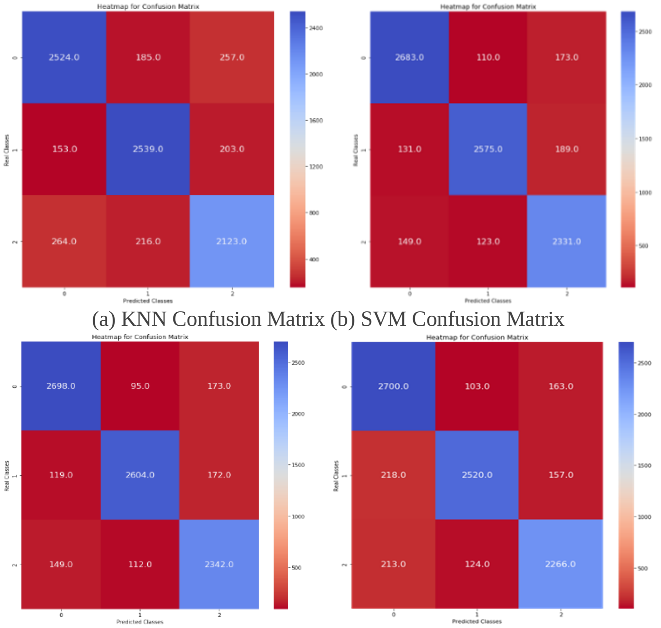
To optimize the results in each classification model, pipeline and grid search were used. In the pipeline, different parameters were selected, the grid search uses this pipeline and creates a combination from these parameters to train the model with cross validation value and then select the best results from these parameters.

VI. Result and Discussion

This section describes the results produced from our model. Based on the grid search, the best K parameter for the KNN classifier is 15 with 84.57% training accuracy. The testing accuracy of the 15-KNN model is 84.73%. Moreover, the confusion matrix is shown in Fig. 4(a), the highest false classification is in the third class 'travel' with 289 classified as 'style & beauty' and 177 as 'parenting'. The 15-KNN model recall score is 84.6%. While SVM best parameters are for gamma is 15 with C equals to 0.01 based on the grid search results, which gave a 90% training accuracy and 89.19 % testing accuracy. Additionally, the result of the confusion matrix is shown in Fig. 4. From the grid search the best parameters found was 600 for the number of estimators, 0.25 for learning rate, 7 for the tree maximum depth, 6 for minimum split sampling and 0.1 for minimum split leaf.

SVM increased the accuracy and the recall it also minimized the misclassified classes, as shown in the confusion matrix in Fig. 4(b). Finally, Naïve Bayes showed a high accuracy with 90.16% in training while the testing got a bit lower with 90.12% while the recall is 90.14%. Similarly Fig. 4(c, d) represent the confusion matrix for Multi-Nominal Naïve Bayes and Gradient Boosting. Table V shows each classifier result with precision, recall, F1 and accuracy metric for each category of news headlines.

Table VI shows the overall outcome of all the classifiers. As shown in the table most of the classifiers have low variance and low bias which indicate the proposed model doesn't have underfitting and overfitting. Also, most of the classifiers has a high recall, as recall is one of the important measures in the text classification problem. The Gradient Boosting classifier outperforms the other classifier in terms of precision and F1 score. While, the naïve Bayes gives the best accuracy and recall between them all.



(a) KNN Confusion Matrix (b) SVM Confusion Matrix

(c) MNB Confusion Matrix. (d) GB Confusion Matrix

Fig. 4. Confusion Matrix for Each Classifier.

TABLE. V. Results Comparison

Classifier	Class	Precision	Recall	F1-Score
KNN	Travel	0.82	0.82	0.82
	Style & beauty	0.86	0.87	0.86
	Parenting	0.86	0.85	0.85
SVM	Travel	0.87	0.90	0.88
	Style & beauty	0.92	0.89	0.90
	Parenting	0.91	0.90	0.90
Multinomial Naive Bayes	Travel	0.87	0.91	0.89
	Style & beauty	0.89	0.89	89
	Parenting	0.91	0.90	0.90
Gradient Boosting	Travel	0.88	0.87	0.87
	Style & beauty	0.92	0.87	0.89
	Parenting	0.86	0.91	0.88

TABLE. VI. OVERALL RESULT

Classifier	Precision	Recall	F1-Score	Testing Accuracy	Training Accuracy
KNN	84.66 %	84.66%	84.66%	84.73%	84.57%
SVM	86.33%	89.66%	86.82%	89.19%	90%
MNB	88.33%	90%	88.14%	90.12%	90.16%
GB	90%	88.33%	89.81%	88.58%	87.66%

VII. CONCLUSION

In this paper, we produced a text classification model that maintains the semantics of text to gain more accuracy and recall score. The semantics of the text was preserved by using word2vec word embeddings with TF-IDF vectorizer. We conducted a comparison between different classifiers the KNN, SVM, Naïve Bayes and the Gradient Boosting classifiers. The training was done with parameter tuning and optimization to give a better result. The best classifier out of these are Multinomial Naïve Bayes which has higher accuracy and recall compared to other classifiers. Also, compared to the other reviewed studies with the same classifier our classifier has better accuracy. The limitation of our study is it covers only three categories of the news. In the future, a possible enhancement to our work is to apply the classification on more than three targets in the dataset and improve the model using methodologies like the neural network.

REFERENCES

[1] M. P. Naik, H. B. Prajapati, and V. K. Dabhi, “A survey on semantic document clustering,” 2015 IEEE Int. Conf. Electr. Comput. Commun. Technol., pp. 1–10.

[2] R. A. Sinoara, “Text mining and semantics : a systematic mapping study,” 2017.

[3] X. Yang, C. Macdonald, and I. Ounis, “Using word embeddings in Twitter election classification,” Inf. Retr. J., vol. 21, no. 2–3, pp. 183–207, 2018.

[4] S. M. Haewoon Kwak, Changhyun Lee, Hosung Park, “What is Twitter, a Social Network or a News Media?,” WWW 2010, April 26–30, 2010, Raleigh, North Carolina, USA. ACM 978-1-60558-799-8/10/04, 2010.

[5] Y. Wang et al., “A comparison of word embeddings for the biomedical natural language processing,” J. Biomed. Inform., vol. 87, pp. 12–20, 2018.

[6] T. Mikolov, G. Corrado, K. Chen, and J. Dean, “Vector Space,” pp. 1–12.

[7] J. Pennington, R. Socher, and C. D. Manning, “GloVe : Global Vectors for Word Representation,” pp. 1532–1543, 2014.

[8] B. Jang, I. Kim, and J. W. Kim, “Word2vec convolutional neural networks for classification of news articles and tweets,” PLoS One, vol. 14, no. 8, pp. 1–21, 2019.

[9] B. Jang and J. Yoon, “Characteristics Analysis of Data from News and Social Network Services,” IEEE Access, vol. 6, pp. 18061–18073, 2018.

[10] C. Zhang, L. Zhang, C. Wang, and J. Xie, “Text Summarization Based on Sentence Selection with Semantic Representation,” 2014 IEEE 26th Int. Conf. Tools with Artif. Intell., pp. 584–590, 2014.

[11] “Graph Based Text Representation for Document,” Vol. 76, No. 1, 2015.

[12] K.- Means, “C Ontext - Aware C Lustering using G Lo V E and,” no. July, 2017.

[13] T. Wei, Y. Lu, H. Chang, Q. Zhou, and X. Bao, “A Semantic Approach for Text Clustering using WordNet and Lexical Chains Expert Systems with Applications A semantic approach for text clustering using WordNet and lexical chains,” Expert Syst. Appl., vol. 42, no. 4, pp. 2264–2275, 2014.

[14] S. Baccianella, A. Esuli, and F. Sebastiani, “S ENTI W ORD N ET 3. 0 : An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining,” vol. 0, pp. 2200–2204, 2008.

[15] A. Khadjeh Nassirtoussi, S. Aghabozorgi, T. Ying Wah, and D. C. L. Ngo, “Text mining of news-headlines for FOREX market prediction: A Multi-layer Dimension Reduction Algorithm with semantics and sentiment,” Expert Syst. Appl., vol. 42, no. 1, pp. 306–324, 2015.

[16] A. Khadjeh, S. Aghabozorgi, T. Ying, D. Chek, and L. Ngo, “Expert Systems with Applications Text mining of news-headlines for FOREX market prediction : A Multi-layer Dimension Reduction Algorithm with semantics and sentiment,” Expert Syst. Appl., vol. 42, no. 1, pp. 306–324, 2015.

[17] S. Seifollahi and M. Shajari, “Word sense disambiguation application in sentiment analysis of news headlines : an applied approach to FOREX market prediction Word sense disambiguation application in sentiment analysis of news headlines : an applied approach to FOREX market prediction,” no. April 2018, 2019.

[18] S. S. Jadhav and S. D. Thepade, “Fake News Identification and Classification Using DSSM and Improved Recurrent Neural Network Classifier,” Appl. Artif. Intell., vol. 33, no. 12, pp. 1058–1068, 2019.

[19] R. A. Pambudi, Adiwijaya, and M. S. Mubarok, “Multi-label classification of Indonesian news topics using Pseudo Nearest Neighbor Rule,” J. Phys. Conf. Ser., vol. 1192, no. 1, 2019.

[20] V. V. Corinna Cortes, “Support-Vector Networks,” Kluwer Acad. Publ., 1995.