KnE Social Sciences

EBEEC 2019
Economies of the Balkan and Eastern European Countries
Volume 2019

Knowledge E
enriching | engaging | empowering

Conference Paper

A New Similarity Measure for Document Classification and Text Mining

Mete Eminağaoğlu[1] and Yılmaz Gökşen[2]

[1]Dept. of Computer Science, Dokuz Eylül University, Tınaztepe, Buca, İzmir, Turkey
[2]Dept. of Management Information Systems, Dokuz Eylül University, Buca, İzmir, Turkey

Abstract

Accurate, efficient and fast processing of textual data and classification of electronic documents have become an important key factor in knowledge management and related businesses in today's world. Text mining, information retrieval, and document classification systems have a strong positive impact on digital libraries and electronic content management, e-marketing, electronic archives, customer relationship management, decision support systems, copyright infringement, and plagiarism detection, which strictly affect economics, businesses, and organizations. In this study, we propose a new similarity measure that can be used with k-nearest neighbors (k-NN) and Rocchio algorithms, which are some of the well-known algorithms for document classification, information retrieval, and some other text mining purposes. We have tested our novel similarity measure with some structured textual data sets and we have compared the results with some other standard distance metrics and similarity measures such as Cosine similarity, Euclidean distance, and Pearson correlation coefficient. We have obtained some promising results, which show that this proposed similarity measure could be alternatively used within all suitable algorithms, methods, and models for text mining, document classification, and relevant knowledge management systems.

Keywords: text mining, document classification, similarity measures, k-NN, Rocchio algorithm

Corresponding Author:
Mete Eminağaoğlu
mete.eminagaoglu@deu.edu.tr

Received: 17 November 2019
Accepted: 6 January 2019
Published: 12 January 2020

Publishing services provided by Knowledge E

Selection and Peer-review under the responsibility of the EBEEC Conference Committee.

1. Introduction

Text classification or document categorization is one of the main research and application areas in text and web mining today. Text classification can simply be defined as the task of assigning predefined categories or classes to texts based on the contents of the documents [1]. The rapid increase in the amount and usage of digital text data, such as electronic news articles, digital libraries, and blogs, has enabled text classification to become a key player in the field of natural language processing or other text-based knowledge applications and management information systems. Text mining and document classification applications have a strong positive impact on

digital libraries and electronic content management, e-marketing, electronic archives, customer relationship management, decision support systems, copyright infringement, and plagiarism detection, which strictly affect economics, businesses, and organizations. Text classification problems can be solved by applying supervised learning algorithms to train classification models with a collection of previous examples by specific machine learning and classifier algorithms, for which the correct classifications (or labels) are known [1, 2].

There are some crucial aspects that make text categorization or classification different from classification cases in data mining. First, the raw textual data is much more unstructured than the data used in classical data mining. This fact brings about the requirement of some extra data pre-processing phases for text mining and document classification, such as tokenization, stemming, stop words, lemmatization, and some other computational linguistics methodologies, if necessary [2, 3]. Second, in most cases, the number of documents or instances and the number of features are significantly higher in text mining applications when compared to ordinary data mining cases. This brings about two new necessities; one of them is the need for very high data storage and fast computational capacities, and the other is the extreme criticality of feature reduction or selection [4]. Another aspect that makes text mining and document classification a more challenging task than ordinary data mining is the number of categories or classes. In most of the document categorization problems, the number of classes might vary between five up to hundreds of different categories [2].

Accurate prediction of new test documents and assigning them to the correct categories is one of the primary aims in text classification. The success of a text classification mainly relies on three major processes: appropriate text pre-processing, suitable and feasible feature selection, and the machine learning or classification algorithm's training capability [1, 4]. There are different types of machine learning algorithms and models that are used in text classification today. One of the oldest machine learning approaches for text classification, which is still in use today, is Naïve Bayes classifier [5]. It is known as one of the simplest and fastest text classification algorithms that are based on statistical modeling. However, if there are more than two categories, then different versions of Naïve Bayes are implemented and used more effectively. Multinomial Naïve Bayes algorithm is one of such approaches [2, 6]. There are also some other approaches that are based on Naïve Bayes and its adaptation to text categorization problems, such as Complement Naïve Bayes [7], and Negation Naive Bayes [8].

Some of the kernel-based functions that are used in machine learning are also implemented for text categorization and one of them is support vector machines (SVM).

A number of different methods exist for determining a good support vector for a category and for calculating to which class a document belongs to from a set of support vectors for multiple classes in multidimensional space by selecting the optimal kernel and setting appropriate parameters for the kernels [9, 10]. However, it should be noted that all types of SVM models are designed for binary classification. Hence, some additional methods must be included and used for document categorization with three or more classes.

There are some other models that have been adapted and used for document classification, such as artificial neural networks and clustering. For instance, learning vector quantization (LVQ), which is a type of simple neural network, is used for text classification [11]. Another study that adapts artificial neural networks for text categorization is the hierarchical perceptrons [12]. It should be noted that hierarchical topic classification is one of the most difficult and challenging areas in document classification because the document is multi-labeled with different categories in a hierarchical manner and the goal is to classify each document correctly for all different categories and sub-categories that it belongs to. The researchers propose that their hierarchical perceptron model achieved better performance scores than multilayer perceptron models or Bayesian approaches [12]. Adaptation of clustering methodology for text classification can be considered as another unique approach [13].

Another type of algorithm that is implemented for text classification is k-NN (k-nearest neighbors), which is an instance-based learner that uses the vector representations of documents and tries to classify the documents by using some similarity measures [14]. Rocchio classifier is another instance-based learning algorithm that uses centroids and vector space models with similarity measures [15]. These two algorithms are the ones that are specially chosen and used in this study and they will be discussed in more detail in the following sections.

Holzinger et al. have made a detailed research and literature survey in biomedical text mining and text classification, where they have elaborated on some open problems and future challenges. In their research article, one of the open problems and future challenges was stated to be an improvement of similarity metrics that are used in vector space models of biomedical text mining and other types of document categorization [16]. The primary aim of our study is to propose a new similarity measure as well, which improves the classification performance and accuracy of classifiers based on vector space models that are used in text mining and document classification. This study proposes a novel similarity measure that can be easily implemented and used for document classification, which also provides some promising results that improve the relevant algorithms' performance in terms of classification accuracy.

2. Materials and Methods

As it was mentioned in the previous sections of this article, a new similarity measure was proposed, implemented and used among different datasets that were derived from Turkish texts. The proposed similarity measure can be used as an alternative metric in text mining and document classification where the attribute values are composed of non-negative numerical values that were derived from frequencies and other statistical measures of words in the documents. Our novel similarity measure is compared with the basic well-known similarity measured and distance metrics that are mostly used in text classification.

It should be noted that there are various text classification algorithms, but if the similarities between different texts / documents / records are to be analyzed by using each feature's value by means of vector values in a hyper-space, then k-NN (k-nearest neighbors) and Rocchio algorithms are mostly used. Hence, these two algorithms with relevant similarity measures were used and comparatively analyzed with our novel method in our experiments.

2.1. Similarity measures and relevant classifiers

In this section, the basic similarity measures and the relevant classification algorithms used in text mining and document categorization problems, which are specified as ``instance-based learners'', are explained. k-NN is a type of instance-based machine learning algorithm where ``a set of labeled training examples is used to predict the new example's label within the range of its nearest k neighbor(s) that are identified using a similarity or distance measure'' [17]. This machine learning algorithm can be used for regression, binary or multi-classification problems with any type of data. It is known that if k-NN is to be used for text categorization or document classification, Cosine similarity, Euclidean distance, and Pearson correlation coefficient are mostly preferred as the similarity measure. It is usually recommended not to use the closest single neighbor *(k=1)* only, but also should be experimented and observed for *k=2, 3,* and so on. However, in this study, when k was set to a higher value than one, the performance values significantly degraded for all of the data sets with any of the alternative similarity metrics. Thus, only the experimental results for *k=1* is included in this article.

Rocchio classifier is another instance-based learning algorithm that is mostly used for document classification. Although this algorithm was originally designed for information retrieval systems [18], it was converted to a text classifier algorithm later on. Rocchio

algorithm for classification can be simply described as ``a centroid-based classifier that calculates representative vectors of each class to define the class boundaries, where the representative vectors (centroids) are given by the average vector of the training documents assigned to the represented class" [15]. Since the similarity measures in Rocchio algorithm are mostly preferred as Euclidean distance and Cosine similarity, these two were also used and comparatively analyzed with our novel similarity measure. Rocchio classification algorithm is shortly described in equation (1).

$$Let\ D_c\ be\ the\ set\ of\ all\ documents\ in\ class\ c\ and\ v\ \ d\ \ be\ the\ vector\ space\ of\ d.$$
$$The\ centroid\ of\ class\ c\ is\ calculated\ as\ follows:$$
$$\mu(c) = \frac{1}{D_c} \sum_{d \in D_c} v(d) \tag{1}$$
$$A\ new\ document\ or\ record\ will\ be\ assigned\ to\ the\ closest\ centroid's\ class.$$
$$Cosine\ similarity\ or\ Euclidean\ distance\ is\ used\ to\ calculate\ the\ closeness.$$

Euclidean distance, Cosine similarity, and Pearson correlation coefficient measures are shortly given in equations (2), (3), and (4). It should be noted that the coordinates of vectors are in fact the records, instances, or documents and consequently, vector elements or values are in fact the values of the features or attributes of that instance.

$$Let\ p = \ p_1, p_2, \dots\ p_n\ \ and\ q = \ q_1, q_2, \dots\ q_n$$

$$Euclidean\ distance\ between\ p\ and\ q\ is;$$
$$d\ (p,q) = \sqrt{\sum_{i=1}^{n} p_i - q_i\ ^2} \tag{2}$$

$$Cosine\ similarity\ between\ p\ and\ q\ is;$$
$$Cos\_sim\ (p,q) = \frac{\sum_{i=1}^{n} p_i q_i}{\sqrt{\sum_{i=1}^{n} p_i^2}\ \sqrt{\sum_{i=1}^{n} q_i^2}} \tag{3}$$

$$Pearson\ cc\ (correlation\ coefficient)\ between\ p\ and\ q\ is;$$
$$cc\ \ p,q\ \ = \frac{\sum_{i=1}^{n} p_i - p\ \ q_i - q}{\sqrt{\sum_{i=1}^{n} p_i - p\ ^2}\ \sqrt{\sum_{i=1}^{n} q_i - q\ ^2}} \tag{4}$$

$$where\ p\ \ and\ q\ \ are\ means\ of\ p\ and\ q$$

2.2. Proposed similarity measure

The novel similarity measure proposed in this study is shown in equation (5). The relative difference between the two instances is calculated first, and then the similarity value is achieved. The similarity values that can be obtained by this new method can be any continuous value between zero and one. If the similarity is 1, this shows that both records are identical, which is also true for Cosine similarity and Pearson cc. On the other hand, for the most dissimilar vector or records, our method produces a very small value close to zero. If both records' all attributes are equal to zero, then our method does not make any calculation and directly sets the similarity between those two instances as zero.

It should be noted that this novel measure could only be used for data sets having non-negative numerical values. This is due to the fact that this similarity measure is especially aimed, designed, and implemented for document classification where the representative vector values of each document can only be 0 or positive continuous numbers.

$$
\begin{aligned}
&for\ two\ instances\ \ texts\ or\ records\ \ a_i\ and\ a_j, and\ n\ attributes\ \ words\ ;\\
&Rd\ \ Relative\ difference\ =\ \sum_{k=1}^{n} d_k = \sum_{k=1}^{n} \frac{a_{ik} - a_{jk}}{0.5\ \ a_{ik} + a_{jk}}\\
&where\ d_k = 0\ if\ a_{ik} = a_{jk} = 0 \qquad (5)\\
&Similarity = \frac{2}{1 + e^{\frac{Rd}{s}}}\\
&where\ \ s =\ \ number\ of\ attributes\ with\ d_k\ \neq 0\ \ \ and\ s > 0
\end{aligned}
$$

The similarity measure proposed in this study can also be demonstrated with a simple example. For instance, suppose that there are three different documents (texts) *d1, d2*, and *d3* with four attributes, where each attribute is a specific word and its numerical value stands for its normalized term frequency value, which is given in Table 1.

Table 1: A sample data for similarity measure.

	a1	a2	a3	a4
d1	3.55	4.23	0.00	0.00
d2	0.86	0.00	0.00	0.00
d3	0.00	4.23	1.00	1.12

The relative difference between *d1* and *d2* will be calculated as 3.22, and the relative difference between *d1* and *d3* will be 6. Consequently, the similarity between *d1* and *d2* will be measured as 0.333 and the similarity between *d1* and *d3* will be found out as 0.365 according to our method. Hence, document *d1* is observed to be slightly more similar to *d3* than *d2*, according to our similarity measure. The proposed similarity measure can be easily adapted and used in k-NN and Rocchio classifier algorithms for text classification instead of other known measures such as Cosine similarity, Euclidean distance, and so on.

2.3. Datasets and performance evaluation methods

Several datasets were used in this research in order to evaluate and compare the performance of classifiers with standard similarity measures versus our novel similarity measure. One of these datasets is TTC-3600, which is a universal benchmark Turkish dataset for document classification that can be publicly accessed via Internet. TTC-3600 is a collection of Turkish news and articles including categorized 3600 text documents from well-known news portals in Turkey [19]. There are six classes / categories in TTC-3600, economy, culture-arts, health, politics, sports and technology, namely, each category with 600 instances. Kılınç et al. derived four different alternative structured datasets from the raw text data, which all of them were also tested in this study. One of these alternatives is the original dataset with 3600 instances and 7507 features (words). Another dataset with 5692 features was constructed by using Zemberek stemmer [20]. Stemmer algorithms are used for stemming the words into their root forms for that specific language according to its grammatical and linguistic rules; however, these rules are not as complex and specific as the ones in lemmatization [15]. Zemberek is one of the well-known and most preferred stemmer algorithm and tool for stemming Turkish lexicons. Two additional datasets were also constructed in the same manner by using different stemmers, F5 and F7, namely [19]. The dataset constructed by F5 stemmer had 3208 features and the one constructed by F7 stemmer had 4812 features. All of the datasets had exactly the same number of records, which is 3600.

A different dataset was also constructed and developed by the authors of this study. This proprietary data was obtained from a private software company in İzmir, Turkey. The unstructured Turkish data was composed of daily business messages and live reports produced by software engineers, help-desk staff, and company managers. Each of the message or report belongs to a specific client, so the dataset had four different categories / classes. The dataset that was obtained from the software company had

originally 17869 records (documents) and 1223 features (words) after tokenization and stemming processes by using Zemberek stemmer. We also used ``Incremental Wrapper Subset Selection (IWSS) with Naive Bayes'' algorithm for feature reduction [21]. Thus, after the feature reduction process, and the elimination of records with all zero values, the proprietary dataset finally was composed of 12494 instances, 138 features, and 4 classes. It should be mentioned that some of the texts in this dataset contained out-of-vocabulary words such as technical jargon, or English terms, and the sizes of the messages varied significantly. The class imbalance problem can be considered as another difficult issue in this dataset. The number of documents for each of the four categories is 5793, 3344, 2320, and 1037 that make up 12494 instances in total.

As it was mentioned in the previous paragraphs, the data sets used in this study are composed of *tf-idf (term frequency-inverse document frequency)* values of words in the documents, which is one of the most common frequency-based metrics among text and document classification tasks [2]. The *tf-idf* calculation is described in equation (6).

$$\begin{aligned}&Let\ freq\ \ i,j\ \ \ be\ the\ number\ of\ occurrences\ of\ keyword\ i\ in\ document\ j\\&Let\ N\ denote\ the\ total\ number\ of\ documents\ (instances)\\&Let\ n\ \ i\ \ \ denote\ the\ number\ of\ documents\ in\ which\ word\ i\ is\ observed\\&tf(i,j)=\frac{freq(i,j)}{\sum_{i'\in j} freq(i',j)}\\&idf(i)=log_{10}\frac{N}{n(i)}\\&tf\_idf\ \ i,j\ \ =tf\ \ i,j\ \ *idf(i,j)\end{aligned}\tag{6}$$

On the other hand, if the length of the documents in a data set varies significantly, then augmented (double-normalized) *tf-idf* is usually used in order to prevent the bias towards longer documents [3]. The augmented *tf-idf* is given in equation (7).

$$\begin{aligned}&Let\ maxOthers\ \ i,j\ \ denote\ the\ highest\ number\ of\\&occurrences\ of\ another\ keyword\ of\ j\\&tf\_idf(i,j)=0.5+\ \ 0.5\frac{freq(i,j)}{maxOthers(i,j)}\ \ log_{10}\frac{N}{n(i)}\end{aligned}\tag{7}$$

The prediction performance and the accuracy of classifier algorithms in text mining and document classification are mostly evaluated by calculating Precision, Recall, and F-measure [2, 22]. Precision can be simply defined as ``the percentage of retrieved

documents that are in fact relevant to a query", and Recall can be described as ``the percentage of documents that are relevant to the query and were in fact retrieved" [22]. F-measure (F1 Score) is the harmonic mean of Precision and Recall. These performance measures are also given in equation (8).

$TP: True\ Positives\ (intersection\ of\ relevant\ and\ retrieved\ documents)$
$FP: False\ Positives\ (incorrectly\ retrieved\ documents\ that\ were\ not\ relevant)$
$FN: False\ Negatives\ (incorrectly\ missed\ documents\ that\ were\ relevant)$

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN} \tag{8}$$

$$F\_measure = 2 * \frac{Precision * Recall}{Precision + Recall}$$

In most of the document classification problems, there are more than two classes or categories. Hence, besides observing the performance measures of records for each class in a dataset, the average of Precision, Recall, and F-measure scores should also be calculated. There are two measurement methods for this purpose, macro average and micro average, namely. The micro average is also known as ``the weighted average of all classes where the overall value is dominated by the more frequent class" [2]. On the other hand, in the macro average, the performance score for each class is calculated independently first, and then the arithmetic mean of these are obtained as the macro average value. Macro average ``measures the classification effectiveness for each class, considered independently and equally important" [23]. In other words, if the number of records among several classes is imbalanced and each category has the same importance, the macro average must be taken into consideration first. If the number of instances for each class is equal, then macro and micro average values will also be the same. In this study, since each of the six categories had the same number of records in TTC-3600 dataset, only micro average was observed. However, for the proprietary dataset obtained from the software company, macro and micro averages were separately observed because there was a class imbalance problem.

3. Results

The results within the experiments conducted on structured Turkish texts were obtained by stratified 10-folds cross-validation. Cross-validation is one of the most reliable and accurate techniques when the data has to be separated into train and test data sets [24].

In ``k-fold cross-validation'', the initial data set is partitioned into *k* ``mutually exclusive subsets'' or ``folds'' [25]. In each round, *k-1* folds are used for training, the other fold is used for testing, and this procedure is repeated *k* times so that every instance in the data set will have been used exactly once for testing [25]. All of the algorithms and similarity measures were developed in Java programming language using Java NetBeans IDE v8.0.1 platform. The comparative results obtained from the experiments conducted with four different datasets of TTC-3600 are given in Tables 2, 3, 4, and 5. It should be noted that the macro averages are not included in these tables because both micro and macro averages exactly give the same results due to the fact that the datasets are composed of an equivalent number of records for each class or category.

TABLE 2: Comparative results from the first dataset in TTC-3600. This dataset has 3600 instances and 7507 features.

Algorithm name and parameters	**Average F-Score (micro average)**	**Average Precision (micro average)**	**Average Recall (micro average)**
Rocchio classifier (Cosine similarity)	0.714	0.713	0.715
Rocchio classifier (Euclidean distance)	0.709	0.708	0.711
k-NN (k=1, Pearson cc)	0.574	0.571	0.577
k-NN (k=1, Euclidean distance)	0.579	0.589	0.570
k-NN (k=1, Cosine similarity)	0.593	0.594	0.592
Rocchio classifier (proposed similarity method)	**0.753**	**0.755**	**0.752**
k-NN (k=1, proposed similarity method)	**0.662**	**0.654**	**0.671**

TABLE 3: Comparative results from the second dataset in TTC-3600. The words in this set were constructed by Zemberek stemmer. This dataset has 3600 instances and 5692 features.

Algorithm name and parameters	**Average F-Score (micro average)**	**Average Precision (micro average)**	**Average Recall (micro average)**
Rocchio classifier (Cosine similarity)	0.715	0.721	0.710
Rocchio classifier (Euclidean distance)	0.712	0.711	0.714
k-NN (k=1, Pearson cc)	0.567	0.565	0.570
k-NN (k=1, Euclidean distance)	0.561	0.554	0.568
k-NN (k=1, Cosine similarity)	0.587	0.587	0.588
Rocchio classifier (proposed similarity method)	**0.801**	**0.792**	**0.811**
k-NN (k=1, proposed similarity method)	**0.677**	**0.671**	**0.684**

The comparative results obtained from the experiments conducted with the proprietary dataset is given in Table 6. It can be seen from all of the results and corresponding tables that our novel similarity measure improves all of the performance scores within both classifier algorithms, k-NN and Rocchio, namely. It can also be observed that

Table 4: Comparative results from the third dataset in TTC-3600. The words in this set were constructed by F5 stemmer. This dataset has 3600 instances and 3208 features.

Algorithm name and parameters	Average F-Score (micro average)	Average Precision (micro average)	Average Recall (micro average)
Rocchio classifier (Cosine similarity)	0.736	0.732	0.741
Rocchio classifier (Euclidean distance)	0.722	0.725	0.720
k-NN (k=1, Pearson cc)	0.566	0.573	0.559
k-NN (k=1, Euclidean distance)	0.562	0.558	0.566
k-NN (k=1, Cosine similarity)	0.581	0.581	0.581
Rocchio classifier (proposed similarity method)	**0.812**	**0.806**	**0.819**
k-NN (k=1, proposed similarity method)	**0.668**	**0.659**	**0.677**

Table 5: Comparative results from the fourth dataset in TTC-3600. The words in this set were constructed by F7 stemmer. This dataset has 3600 instances and 4813 features.

Algorithm name and parameters	Average F-Score (micro average)	Average Precision (micro average)	Average Recall (micro average)
Rocchio classifier (Cosine similarity)	0.708	0.714	0.702
Rocchio classifier (Euclidean distance)	0.703	0.705	0.701
k-NN (k=1, Pearson cc)	0.565	0.561	0.570
k-NN (k=1, Euclidean distance)	0.546	0.554	0.538
k-NN (k=1, Cosine similarity)	0.588	0.581	0.596
Rocchio classifier (proposed similarity method)	**0.789**	**0.783**	**0.795**
k-NN (k=1, proposed similarity method)	**0.688**	**0.684**	**0.692**

the highest F-Score, Precision, and Recall values were always obtained by using the proposed similarity measure with Rocchio classifier among all of the datasets.

The micro averages for all of the metrics amongst all of the classifiers in Table 6 are observed to be significantly higher than the corresponding macro averages. This is an expected and reasonable outcome because as it was mentioned before, there is a class imbalance problem in this proprietary dataset, which produces biased micro average results regarding the domination of some of the categories. Hence, it would be a more reliable and realistic approach to focus on the macro average results rather than the micro averages in Table 6.

Table 6: Comparative results from the proprietary dataset. The words in this set were constructed by Zemberek stemmer and Incremental Wrapper with Naive Bayes was used for feature reduction. This dataset has 12494 instances, 138 features, and 4 classes.

Algorithm name and parameters	Average F-Score (micro average)	Average F-Score (macro average)	Average Precision (micro average)	Average Precision (macro average)	Average Recall (micro average)	Average Recall (macro average)
Rocchio classifier (Cosine similarity)	0.752	0.685	0.750	0.681	0.754	0.689
Rocchio classifier (Euclidean distance)	0.706	0.652	0.706	0.652	0.706	0.652
k-NN (k=1, Pearson cc)	0.704	0.649	0.703	0.648	0.706	0.650
k-NN (k=1, Euclidean distance)	0.713	0.654	0.705	0.653	0.721	0.655
k-NN (k=1, Cosine similarity)	0.722	0.656	0.711	0.651	0.733	0.662
Rocchio classifier (proposed similarity method)	**0.800**	**0.698**	**0.798**	**0.698**	**0.802**	**0.698**
k-NN (k=1, proposed similarity method)	**0.754**	**0.696**	**0.754**	**0.697**	**0.754**	**0.695**

4. Conclusion

The results obtained in this study show that the proposed new similarity measure could be considered and used as an alternative similarity measure for document categorization. It can be concluded that the new similarity measure improves the classification accuracy (in terms of Precision, Recall, and F-Score) of instance-based algorithms and vector space models such as k-NN and Rocchio. In addition, this proposed similarity measure might be adapted flexibly and effectively used within all suitable algorithms, methods, and models for text mining, document classification, and relevant knowledge management systems. It should be noted that the datasets used to evaluate the performance of our novel similarity measure were Turkish texts, which might be considered as the only limitation of this study. This is due to the fact that the method proposed in this study was developed during a research for a Turkish company and the primary objective was to obtain a feasible model for textual data in Turkish language. Thus, one of our further studies will be test and observation of our similarity measure's performance among universal benchmark datasets in English, such as Reuters, BBC news, and 20 newsgroups. Thus, one of our further studies will be test and observation of our similarity measure's performance among universal benchmark datasets in English, such as Reuters and 20 newsgroups, as well as some other alternative Turkish textual data. It should also be noted that the data types are limited to non-negative numerical values for the proposed similarity measure. However, new alternatives to this measure could be implemented if it is to be used for data mining purposes with categorical attributes or negative numbers.

References

[1] Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, vol. 34, pp. 1-47.

[2] Jurafksy, D. and Martin, J. H. (2017). *Speech and Language Processing*. USA: Prentice Hall.

[3] Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. USA: The MIT Press.

[4] Pant, G. and Srinivasan, P. (2005). Learning to crawl: Comparing classification schemes. *ACM Transactions on Information Systems*, vol. 23, pp. 430-462.

[5] Zhang, L. et al. (2004). An evaluation of statistical spam filtering techniques. *ACM Transactions on Asian Language Information Processing*, vol. 3, pp. 243-269.

[6] McCallum, A. and Nigam, K. (1998). A comparison of event models for Naive Bayes text classification, in *Proceedings of AAAI/ICML-98 Workshop on Learning for Text Categorization*, Madison, Wisconsin, USA: The AAAI Press.

[7] Rennie, J. D. M. et al. (2003). Tackling the poor assumptions of Naive Bayes text classification, in *Proceedings of the Twentieth International Conference on Machine Learning*, Washington D.C., USA: The AAAI Press.

[8] Komiya, K. et al. (2011). Negation Naive Bayes for Categorization of Product Pages on the Web, in *Proceedings of Recent Advances in Natural Language Processing*, Hissar, Bulgaria.

[9] Lodhi, H. et al. (2002). Text Classification using String Kernels. *Journal of Machine Learning Research*, vol. 2, pp. 419-444.

[10] Yu, H. et al. (2002). PEBL: positive example based learning for web page classification using SVM, in *Proceedings of the Eighth International Conference on Knowledge discovery and Data Mining*, Edmonton, Canada.

[11] Martin-Valdivia, M. T. et al. (2007). The learning vector quantization algorithm applied to automatic text classification tasks. *Neural Networks*, vol. 20, no. 6, pp. 748-756.

[12] Chen, C. et al. (2005). A Hierarchical Neural Network Document Classifier with Linguistic Feature Selection. *Applied Intelligence*, vol. 23, pp. 277-294.

[13] Liu, L. and Peng, T. (2014). Clustering-based Method for Positive and Unlabeled Text Categorization Enhanced by Improved TFIDF. *Journal of Information Science and Engineering,* vol. 30, pp. 1463-1481.

[14] Kwon, O. and Lee, J. (2003). Text categorization based on k-nearest neighbor approach for web site classification. *Information Processing and Management*, vol. 39, pp. 25-44.

[15] Manning, C. D. et al. (2009). *Introduction to Information Retrieval*. UK: Cambridge University Press.

[16] Holzinger, A. et al. (2014). Biomedical Text Mining: State-of-the-Art, Open Problems and Future Challenges. *Knowledge Discovery and Data Mining*, pp. 271--300.

[17] Aha, D. W. et al. (1991). Instance-Based Learning Algorithms. *Machine Learning*, vol. 6, no. 1, pp. 37-66.

[18] Rocchio, J. (1971). *The SMART Retrieval System: Experiments in Automatic Document Processing*. Prentice Hall, USA.

[19] Kılınç, D. et al. (2015). TTC-3600: A new benchmark dataset for Turkish text categorization. *Journal of Information Science*, vol. 43, no. 2, pp. 174-185.

[20] Akın, A. A. and Akın, M. D. (2007). Zemberek, an open source NLP framework for Turkic Languages, in *Proceedings of 45th Annual Meeting of the Association for Computational Linguistics*, Prague, Czech Republic: ACL.

[21] Bermejo, P. et al. (2014). Speeding up incremental wrapper feature subset selection with Naive Bayes classifier. *Knowledge-Based Systems,* vol. 55, pp. 140-147.

[22] Morariu, D. et al. (2013). Feature Selection in Document Classification, in *Proceedings of the Fourth International Conference in Romania of Information Science and Information Literacy*, Sibiu, Romania: IFLA.

[23] Salles, T. et al. (2010). Automatic Document Classification Temporally Robust. *Journal of Information and Data Management,* vol. 1, no. 2, pp.199-211.

[24] Han, J. and Kamber, M. (2006). *Data Mining: Concepts and Techniques, 2nd ed.*, San Francisco, USA: Morgan Kaufmann Publishers,

[25] Witten, I. H. et al. (2011). *Data Mining: Practical Machine Learning Tools and Techniques, 3rd ed.*, USA: The Morgan Kaufmann Series in Data Management Systems.