Konya Mühendislik Bilimleri Dergisi., c. 10, Özel Sayı, 41-51, 2022
Konya Journal of Engineering Sciences v. 10, Special Issue, 41-51, 2022
ISSN: 2667-8055 (Electronic)
DOI: 10.36306/konjes.1081213

ACADEMIC TEXT CLUSTERING USING NATURAL LANGUAGE PROCESSING

[1]Salimkan Fatma TAŞKIRAN , [2]Ersin KAYA

Konya Technical University, Faculty of Engineering and Natural Sciences, Department of Computer Engineering, Konya, TÜRKİYE
[1] sftaskiran@ktun.edu.tr, [2] ekaya@ktun.edu.tr

(Geliş/Received: 01.03.2022; Kabul/Accepted in Revised Form: 16.11.2022)

ABSTRACT: Accessing data is very easy nowadays. However, to use these data in an efficient way, it is necessary to get the right information from them. Categorizing these data in order to reach the needed information in a short time provides great convenience. All the more, while doing research in the academic field, text-based data such as articles, papers, or thesis studies are generally used. Natural language processing and machine learning methods are used to get the right information we need from these text-based data. In this study, abstracts of academic papers are clustered. Text data from academic paper abstracts are preprocessed using natural language processing techniques. A vectorized word representation extracted from preprocessed data with Word2Vec and BERT word embeddings and representations are clustered with four clustering algorithms.

Keywords: *Natural Language Processing, Machine Learning, Text Representation*

Doğal Dil İşleme ile Akademik Metin Kümeleme

ÖZ: Günümüzde verilere ulaşmak çok kolaylaşmıştır. Ancak bu verileri verimli bir şekilde kullanmak için onlardan doğru bilgileri çıkarmak gerekir. İhtiyaç duyulan bilgiye kısa sürede ulaşabilmek için bu verilerin kategorilere ayrılması büyük kolaylık sağlamaktadır. Akademik alanda araştırma yapılırken genellikle makale, bildiri veya tez çalışması gibi metin tabanlı veriler kullanılmaktadır. Bu metin tabanlı verilerden ihtiyacımız olan doğru bilgiyi elde etmek için doğal dil işleme ve makine öğrenmesi yöntemleri kullanılmaktadır. Bu çalışmada akademik makalelerin özetleri kümelenmiştir. Akademik makale özetlerinden alınan metin verileri, doğal dil işleme teknikleri kullanılarak önceden işlenir. Word2Vec ve BERT ile vektörize edilen kelime temsilleri, dört farklı kümeleme algoritması ile kümelenmiştir.

Anahtar Kelimeler: *Doğal Dil İşleme, Makine Öğrenmesi, Metin Temsili*

1. INTRODUCTION

Thanks to developing technology and globalizing communication networks, it is very easy to access data. However, it is very difficult to select the desired data and perform a qualitative study on the processed data. The abundance and size of the data make it difficult to access the necessary information for the analysis or solution of problems. The processing and classification of data gain importance at this point. Extracting low-dimensional, good data representations from raw data makes it more suitable for use.

Text classification is the categorization of extracted features from texts using various methods. Access to data is extremely easy today, but it is almost impossible to use this data unless the data is in the desired order. For this reason, categorizing texts is a rather complex task, even if it is thought of as simple. Further which subjects are studied more, or which subjects are studied together is of great

importance for academic studies. In order to determine the most effective solution to the problems to be studied, it is very convenient to classify academic texts according to the subjects they contain. With the classification of academic texts, the desired results can be reached quickly in literature searches, and the most effective methods for solving the problem can be easily found.

Natural language processing can be defined as a joint field of linguistics, artificial intelligence, and computer science that deals with the interaction between the natural language used by humans and computers. In the context of this study, some of the natural language processing techniques are used for text preprocessing and representation. Different problems such as author-work matching (Amasyalı ve Diri, 2006), email classification, finding spam mails (Yang and Park, 2002), text subject determination (Bekkerman et al., 2003), sentiment analysis (Medhat et al., 2014) can be classified as text classification problems.

Creating meaningful representations from texts is of great importance in terms of classification and clustering success in such problems. Commonly used text representation methods in the literature can be shown as word or phrase frequencies, hidden meaning indexing, and information gain. With the popularity of artificial neural networks and these mentioned methods, word representation methods in which words are expressed with vectors have been put forward. Word2Vec (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013), GloVe (Pennington, Socher, & Manning, 2014), FastText (Joulin, Grave, Bojanowski, & Mikolov, 2016), BERT (Joulin et al., 2016) are the most widely used ANN-based word representation methods. Also there are models available that can vectorize an entire sentence or paragraph. In addition to extracting the features that will best represent the data set, the preprocessing methods used to clean the unwanted parts of the texts are also important for classification success. There are many different preprocessing methods such as morphological analysis, correcting spelling mistakes, clearing texts from punctuation marks/numbers, removing foreign words or words that are not important for text representation from texts.

In literature text clustering is studied widely and finds its way in applications like customer segmentation, classification, collaborative filtering, visualization, document organization and indexing (Aggarwal & Zhai, 2012). Problems and the scope of the clustering may vary depending on the datasets used in studies. Clustering algorithms applied to multi-domain are various and widely studied (Premalatha & Natarajan, 2010). Also there are studies that explore multi-topic document clustering as in Romeo's thesis (Romeo, Greco, & Tagarelli, 2014). In Popova's (Popova, Danilova, & Egorov, 2014) and Pinto's (Pinto, Rosso, & Jiménez-Salazar, 2011) works clustering short documents with narrow subjects problem is studied. Short document clustering brings the sparsity problems with them. For such problems LDA (Blei, Ng, & Jordan, 2003) based solutions are proposed and used (Onan, Bulut, & Korukoglu, 2017; Tajbakhsh & Bagherzadeh, 2019). For academic texts, clustering is helpful for researchers to explore the subjects studied in the certain areas (Li et al., 2018). There are multiple works in this area for English (Alexandrov, Gelbukh, & Rosso, 2005; Makagonov, Alexandrov, & Gelbukh, 2004; Weißer, Saßmannshausen, Ohrndorf, Burggräf, & Wagner, 2020) but not as many in Turkish.

Although Turkish is not as advanced as Latin-based languages or Arabic, there are many studies in the field of natural language processing. For text classification, Amasyalı's (2006) study is the first text classification study in Turkish using n-grams (Amasyalı & Diri, 2006). In the author recognition study of Türkoğlu (2007) using various feature vectors such as author features, n-grams, and different combinations of these vectors, feature vectors were analyzed by comparing them with each other using different machine learning methods (Türkoğlu et al., 2007). Torunoğlu (Torunoğlu et al., 2011) and Uysal (Uysal & Gunal, 2014) studied how preprocessing techniques affect classification success in Turkish text classification. ITU Natural Language Processing Group, which works on Turkish natural language processing, also has essential studies outside the field of text classification. Works such as the morphological analyzer design (Erygit and Adali, 2003), the first statistical dependency parser for Turkish (Erygit and Oflazer, 2006), conditional random fields (CRF) based name entity work (Şeker and Eryiğit, 2012), and many more can be recognized as important studies by this group.

In this study, academic texts are divided into groups, and it is aimed to get academic texts sets with similar subjects easily. At the same time, it is aimed to extract the subjects studied together from the articles that contain more than one study area and fall into the same group. In this way, the subjects that need to be concentrated during the research can be determined more quickly and the time lost by examining irrelevant studies can be regained. For this purpose, a dataset of Turkish article prefaces was prepared. ANN based text representations were obtained from the created data set and clustering operations were performed using these text representations.

This paper is structured as follows. In the second chapter dataset, preprocessing steps, text representation and clustering methods used in study are described. In third section the results obtained by different clustering methods are presented. The quality of the clusters and the parameter values are discussed. In the last section, comments were made on the development of the study and the details that could be added to it were presented to the reader.

2. MATERIAL AND METHOD

The Turkish prefaces of the articles published in the Konya Journal of Engineering Sciences (KONJES) between 2011-2020 were used for this study's data set. Preface texts were taken directly from PDF files and converted into text files with txt extension.

Table 1. Total statistic of dataset

Total Data	213
Total Word Count (before preprocessing)	21794
Total Word Count	10350
Total Unique Word Count	3562

Statistical information of the data set created is given in Table 1 and Table 2. The unlabeled dataset is labeled by considering the title of the articles, their keywords, the subjects of the referenced articles, and the fields in which the article authors work. The labels of the data set and the number of documents belonging to each label are given in Table 2.

Table 2. Class labels

Class	Document Number
Bilgisayar	29
Elektronik	18
Endüstri	15
Harita	15
Jeoloji	12
Kimya	32
Maden	18
Makine	17
Malzeme	13
Ziraat	4
Çevre	14
İnşaat	26

2.1. Data Preprocessing

Preprocessing texts is an important part of natural language processing problems because the characters, words, and sentences defined at this stage are the basic units transferred to all subsequent

work stages, such as morphological analysis or word type tagging (Kannan and Gurusamy, 2014). Text data often includes numbers, dates, special characters, and commonly used words such as prepositions, conjunctions, and pronouns. These are units that have no importance or low importance in text representations. For this reason, it is appropriate to remove the data from the texts in the preprocessing stage in order to avoid problems in the later stages.

Table 3. Preprocessing stages

Original Text	Bu çalışmada, 9m çaplı ve 900m derinliğe ulaşan düşey bir kuyunun beton tahkimat kalınlıkları, iki boyutlu sayısal analizler ile belirlenmiştir.
1. Stage	bu çalışmada, 9m çaplı ve 900m derinliğe ulaşan düşey bir kuyunun beton tahkimat kalınlıkları, iki boyutlu sayısal analizler ile belirlenmiştir.
2. Stage	bu çalışmada m çaplı ve m derinliğe ulaşan düşey bir kuyunun beton tahkimat kalınlıkları iki boyutlu sayısal analizler ile belirlenmiştir
3. Stage	bu çalışmada çaplı ve derinliğe ulaşan düşey bir kuyunun beton tahkimat kalınlıkları iki boyutlu sayısal analizler ile belirlenmiştir
5. Stage	çalışmada çaplı derinliğe ulaşan düşey kuyunun beton tahkimat kalınlıkları boyutlu sayısal analizler belirlenmiştir

In this study, various preprocessing has been done to make the raw text data suitable for the problem. The preprocessing methods used in this work consist of the following stages:

1. Converting uppercase letters to lowercase letters,

2. Removal of numbers and punctuation marks from the text,

3. Removal of single-letter units that will not make sense for the text,

4. Separating texts into words,

5. Removal of Turkish stop words from texts,

6. Body and root analysis on words.

After the text cleaning stages, the last preprocessing stage is the stemming and lemmatizing stage. This stage is extremely critical for text representation. Some of the text's suffixed words mean the same, but they are perceived as different words by the machines due to the suffixes they take. This also means that machines will perceive the meanings of words differently. Since Turkish is an agglutinative language, there are many different rules and inconsistencies in adding and removing suffixes. The language structure of Turkish makes stemming and lemmatizing very difficult. As a result, automated algorithms do not always reach the same root from words with different suffixes with the same meaning.

In this study, morphological analysis is done by using stemming and lemmatization; two different methods for stemming and one method for lemmatizing. For casing, Snowball's Turkish casing and Turkish Stemmer casing developed by Osman Tuncelli and Burak Özdemir as open source for Python were used (Tuncelli and Özdemir, 2019). Although these two different algorithms give similar results, there may be differences in the reduction of some words to the root. For root analysis, Turkish Lemmatizer, which was developed by Abdullatif Köksal as open-source, was used (Abdüllatif Köksal, 2018). Table 4 shows a sample of lemmatization and stemming results from the dataset.

Table 4. A sample of stemming and lemmatization result

Original Text	Hafif ve yüksek dayanımlı malzemelerden olan magnezyum alaşımları, yetersiz korozyon direnci ve düşük yüzey kalitesi nedeniyle bazı sınırlamalara sahiptir.
After Preprocessing	hafif yüksek dayanımlı malzemelerden olan magnezyum alaşımları yetersiz korozyon direnci düşük yüzey kalitesi nedeniyle sınırlamalara sahiptir
Snowball Stemmer	hafif yük dayanımlı malzeme ola magnezy alaşım yetersiz korozyo direnci düşük yüzey kalites neden sınırlama sahip
Turkish Stemmer	hafif yük dayanım malzeme olan magnezyum alaşım yeters korozyon direnç düşük yüzey kalite neden sınırlama sahip
Turkish Lemmatizer	hafif yüksek dayanım malzeme magnezyum alaşımla yetersiz korozyon direnç düşük yüzey kalite neden sınırla sahip

2.2. Text Representation (Feature Extraction)

After the text preprocessing, the stage of extracting the features that will represent this data from the data comes. Generally, there exists two basic methods for feature extraction:

- Traditional bag of words approach
- Neural network-based approach

In this study, neural network-based approaches known as 'word embedding' are chosen for text representation. ANN-based approaches are newer and generally more successful methods in text classification than the traditional bag of words approaches. Word embedding methods, which are based on representing words as a fixed-size vector, are widely used in natural language processing problems.

The most well-known and widely used word embedding method is the Word2Vec method developed by Mikolov (Mikolov, Chen, et al., 2013) in 2013. In this method, vectors are continuously updated with gradient descent and backpropagation methods for the texts given as input by using a single hidden layer ANN model. The method takes the word meanings into account while vectorizing the words; its power to represent texts insufficient context is higher than frequency-based methods.

To find word vector representations with Word2Vec, a model is first created using textual data. Word vectors are accessed using this model. Here, the data set used to create the model is of great importance. Because as the context increases, models that produce word vectors with better representation power will be created. Once the model is created, the vector representation of each word in the model's dictionary can be easily extracted.

By using the model created after these processes, the vector version of the preprocessed data set was obtained. For each document, the vector forms of the words in the document were found one by one. After this stage, we have documents in which each word is 400-dimensional vectors. To make a multidimensional dataset, each element of which is a list of vectors suitable for classification, a text representation must be extracted using these word vectors. There are various methods to do this. In this study, text representations created by summing and averaging vectors were used.

BERT (Bidirectional Encoder Representations from Transformers) (Devlin, Chang, Lee, & Toutanova, 2018) is a transformer-based model for various NLP tasks. It is a method to designed to pre-train deep bidirectional representations from unlabeled text and these models can be either used to extract language features or can be fine-tuned for a specific task like classification, entity recognition, question answering, etc. It uses attention mechanism (Vaswani et al., 2017) and encoder-decoder architecture which is known as transformer model.

Both Word2Vec and BERT produce a vectorized form of words. The main difference between these word vectors is that the Word2Vec representations have fixed-length feature embeddings while BERT word representations are dynamically informed by the words around them. Which means when there

are synonymous word in different sentences, Word2Vec will produce same vector for all them. But BERT will produce different vectors depending on their meaning in that particular sentence.

BERT models are based on encoder-decoder pairs (transformer block) and attention mechanism. Depending on the number of transformer blocks and hidden layers the models can categorized as Basic BERT or Large BERT. In base models there are 12 transformer blocks and output from each of them can be used to get word embeddings. There are multiple ways to obtain these embeddings from transformer outputs (Devlin et al., 2018). Most used methods are:

- Sum of each blocks outputs,
- Sum from 2nd layer to last layer,
- Sum last four layer,
- Concatenate last four layer.

Here in this study, we used a fine-tuned pretrained base Turkish BERT model from Hugging Face repositories. Word embeddings are obtained as sum of last four layers of transformer blocks.

3. RESULT AND DISCUSSION

In this study K-Means, K-Medoids, Affinity Propagation, and OPTICS algorithms were used for clustering. They are compared using different preprocessing and text representation methods. The stemming and root analysis methods used in the preprocessing stage were changed, and their effects on the results were analyzed. For text-based clustering, Word2Vec and BERT text representations were extracted from word vectors by getting the average value of word vectors. In Table 5, clustering results for Word2Vec representations as the Silhouette index (SL), Davies-Bouldin index (DB), Calinski-Harabaz index (CH), and precision values are given for each algorithm.

Table 5. Experimental results for Word2Vec

Preprocessing Method	Clustering Method	Cluster Number	SL	DB	CH	Precision
Snowball Stemmer Preprocessing	K-Means	12	0.216	1.404	34.464	0.345
	K-Medoids	12	0.150	2.018	22.653	0.390
	OPTICS	**23**	**-0.024**	**1.863**	**3.756**	**0.525**
	Affinity Propagation	40	-0.021	4.15	3.752	0.544
Turkish Stemmer Preprocessing	K-Means	12	0.109	1.618	29.746	0.396
	K-Medoids	12	0.07	1.971	17.382	0.360
	OPTICS	**30**	**-0.19**	**1.713**	**3.801**	**0.651**
	Affinity Propagation	36	0.19	4.287	4.048	0.544
Turkish Lemmatizer Preprocessing	K-Means	12	0.119	1.613	27.716	0.381
	K-Medoids	12	0.108	1.925	17.382	0.366
	OPTICS	**32**	**-0.16**	**1.673**	**4.329**	**0.602**
	Affinity Propagation	43	-0.02	4.258	3.343	0.549

Table 5 shows the clustering results of Word2Vec text representation for the data set preprocessed with three different stemming and root analysis methods. Here, the data set given as input to the algorithms is the document vectors created by averaging the word vectors. Looking at table 5, it can be seen that the K-Means and K-Medoids algorithms give below-average precision and SL index values among the three preprocessing results. In the OPTICS and Affinity Propagation algorithms, which give higher precision, the SL score gives an average value close to zero, while the number of clusters is well above the real value. If a general comment is to be made for the whole table, it is expected that the preprocessing methods will have little effect on the results since Word2Vec creates the vectors by taking into account the meaning of the word.

Table 6. Experimental results for BERT

Clustering Method	Cluster Number	SL	DB	CH	Precision
K-Means	12	0.306	2.061	34.451	0.437
K-Medoids	12	0.300	2.050	35.120	0.440
OPTICS	15	0.260	1.507	48.756	0.730
Affinity Propagation	26	0.056	4.018	6.752	0.560

In table 6 experimental results for BERT representations are given. BERT vectors are obtained from raw data so none of the above morphologic processes are realized at this stage of the study. Here, it is seen that the OPTICS clustering algorithm gives better results than other algorithms. It has higher precision score and better cluster quality since its DB score is lower and CH score is higher than its other peers. K-Means and K-Medoids nearly gave identical results and Affinity Propagation method gave higher precision score but also has higher cluster number. If we compare Table 5 and Table 6 results, it can be seen that the BERT representations have better results than Word2Vec results. This situation can be interpreted as BERT models tokenize the words within themselves and create vectors by considering their different meanings.

Table 7. Word2Vec representations parameter results for K-Means and K-Medoids

Clustering Algorithm	Cluster Number	SL	DB	CH
K-Means	5	0.232	1.270	159.368
	8	**0.246**	**1.133**	**196.894**
	12	0.211	1.281	158.906
	15	0.178	1.336	132.749
	20	0.154	1.341	117.986
K - Medoids	5	0.189	1.459	156.110
	8	**0.210**	**1.323**	**190.667**
	12	0.163	1.498	77.444
	15	0.092	1.571	54.04
	20	0.089	1.481	48.229

In Table 6, 7 and 8, parameter analyzes and comparisons made for clustering algorithms used in the study are given. For K-Means and K-Medoids algorithms, the results obtained by changing the k parameter were compared. For the OPTICS algorithm, effects of the parameter changes within the algorithm were analyzed. Lemmatization with Word2Vec and BERT text representation were used for parameter analysis. While creating text representations from word vectors, the method of adding vectors was used.

Table 8. BERT representations parameter results for K-Means and K-Medoids

Clustering Algorithm	Cluster Number	SL	DB	CH
K-Means	5	0.232	2.454	30.440
	8	**0.357**	**1.600**	**40.678**
	12	0.306	2.061	34.451
	15	0.280	2.365	32.009
	20	0.198	3.084	28.505
K - Medoids	5	0.200	2.187	28.308
	8	**0.333**	**1.802**	**38.570**
	12	0.300	1.950	35.120
	15	0.205	2.185	28.78
	20	0.190	3.100	23.600

Clustering results obtained with different k parameters for the K-Means clustering algorithm are given in Table 6 and Table 7. In Table 6, for the SL clustering index, it can be seen that the parameter k = 8, which has the closest result to 1, gives the best result. It is seen that the highest in-cluster density, that is, the lowest DB index, is achieved when k takes the value of 5. In addition, it was observed that the highest value for the CH index, which is another index used in the measurement of clustering quality, was obtained when the k value was 8. Since higher CH values indicate higher in-cluster density and better cluster quality, it can be said that the best results for K-Means clustering are obtained when the cluster number is eight.

For the K-Medoids algorithm, it can be seen that the SL index has the highest value when k is 8. If we look at the DB index, it can be seen that the lowest value is obtained when k is 8, and the closest results are obtained for k = 5 and k = 12 values. As a result of the CH index, the best value was taken for k = 8.

Table 8 shows results of BERT representations for different K parameters. Here the best SL score is obtained when parameter k = 8 for both K-Means and K-Medoids algorithm. If we look at DB and CH indices both have their best value when k is chosen 8. For other k values, k = 12 gave close results to k = 8 but it has lower cluster quality scores. By looking at the table we can say that the clusters have poorer quality when k value increases too much.

Table 9. Parameter experiment results for OPTICS

Clustering Algorithm	Epsilon	Cluster Number	SL	DB	CH
OPTICS	30	18	-0.337	1.962	3.165
	35	22	-0.333	2.001	2.929
	40	27	-0.332	2.332	2.698
	45	30	-0.296	2.087	2.553

For the OPTICS algorithm, the epsilon (the maximum distance between the data to be evaluated as adjacent to each other) parameter analysis results are given in Table 9. While getting results for the parameter, the default value of the other parameters is used. The default value of the Epsilon parameter in the tools used is given as infinite (inf). Looking table 9, it can be seen that with the increase in the epsilon coefficient, the SL values approach to 0, which means the clustering quality increases, albeit slightly. However, even though the quality increased according to the SL parameter, the number of clusters are also increased. Looking at the DB index, it can be seen that more dense clusters are obtained in cases where the epsilon value is low. Finally, when we look at the CH index, it can be said that a low epsilon value indicates higher results, that is, better clustering results. When a general examination is

made for the OPTICS algorithm, it has been observed that the cluster verification indices give more or less close results, but the number of clusters increases as the epsilon parameter increases.

4. CONCLUSION

In this study, a basic clustering problem was handled using natural language processing and machine learning methods. The data were passed through text cleaning using tokenization, lemmatization, stemming, and preprocessing methods, and Word2Vec word vector representations were extracted from the processed data. In addition to Word2Vec representations, BERT word vectors are obtained from raw data. These text representations were compared using K-Means, K-Medoids, OPTICS, and Affinity Propagation clustering methods. With these word representation methods, the classic preprocessing and feature extraction methods are tested against more recent transformer based pre trained models.

Considering all results BERT text representations produced better results than Word2Vec representations. Word2Vec text representation methods produced results below the expectations regardless of the preprocessing methods used. If the reasons for this situation are to be commented on, Word2Vec model was trained from scratch while BERT model was fine-tuned on a pretrained model. Also the writing style of the article prefaces can be another reason for results. Writing very specific prefaces to the article can make it difficult to find words to express the topics in general. In addition, working on more than one subject or solving problems using hybrid fields makes it difficult to classify articles on a single subject.

Another reason may be the inadequacy of the data set and the imbalance between the number of topics and articles. While there are more than twenty articles from the fields of Computer Engineering and Chemical Engineering in the data set, there are four articles related to the fields that have been little studied, such as Agricultural Engineering. The scarcity of data used also affected the forms of representation created. Although there have been successful text classification studies in Turkish before, natural language processing problems for newly created datasets still maintain their difficulty. How the data set is processed and represented is of great importance for the classification of texts. The agglutinative language structure of Turkish poses a challenge for frequency-based text representation methods. However, when there is sufficient context, there is no problem in producing robust text representations with artificial neural network-based methods.

To carry the study to the next level, developments should be carried out to increase and stabilize the data set, as well as to develop vector representations of documents. Because abstracts are relatively short texts the words have low frequency, and they are more sparse than normal documents. These reasons are also make clustering results unstable. Different solutions should be suggested for the classification of texts that may belong to more than one subject.

If productive results are obtained in the clustering of the texts, the study can continue as the next step, which is to find the subjects studied together and to shape the text classification accordingly. In addition, academic journals can be classified according to the year-based and the subjects they work on, and they can shed light on the subjects they want to work on for those who will do academic studies.

5. ACKNOWLEDGE

This study has been presented in 2nd International Symposium on Implementations of Digital Industry and Management of Digital Transformation (ISIDIMDT'21), 10-11 November 2021, Konya/Turkey. It is an extended version of the work presented at the symposium, in line with the e-mail that states all submitted papers will be included in the evaluation process for publication in the Special Issue of Konya Journal of Engineering Sciences (KONJES).

6. REFERENCES

Adalı, E. (2012). Doğal Dil İşleme. *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, *5*(2).

Aggarwal, C. C., & Zhai, C. (2012). A survey of text clustering algorithms. In *Mining text data* (pp. 77-128): Springer.

Alexandrov, M., Gelbukh, A., & Rosso, P. (2005). *An approach to clustering abstracts.* Paper presented at the International Conference on Application of Natural Language to Information Systems.

Amasyali, M. F., Balc1, S., Mete, E., & Varl1, E. N. (2012). Türkçe Metinlerin Sınıflandırılmasında Metin Temsil Yöntemlerinin Performans Karşılaştırılması / A Comparison of Text Representation Methods for Turkish Text Classification.

Amasyalı, M. F., & Diri, B. (2006). Automatic Turkish text categorization in terms of author, genre and gender. International Conference on Application of Natural Language to Information Systems,

Ankerst, M., Breunig, M. M., Kriegel, H.-P., & Sander, J. (1999). OPTICS: Ordering points to identify the clustering structure. *ACM Sigmod record*, *28*(2), 49-60.

Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J. M., & Perona, I. (2013). An extensive comparative study of cluster validity indices. *Pattern Recognition*, *46*(1), 243-256.

Bekkerman, R., El-Yaniv, R., Tishby, N., & Winter, Y. (2003). Distributional Word Clusters vs. Words for Text Categorization. *J. Mach. Learn. Res.*, *3*, 1183-1208.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, *3*(Jan), 993-1022.

Conneau, A., Schwenk, H., Barrault, L., & Lecun, Y. (2016). Very deep convolutional networks for natural language processing. *arXiv preprint arXiv:1606.01781*, *2*, 1.

Çilden, E. K. (2006). Stemming Turkish Words Using Snowball. https://snowballstem.org/

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dhar, A., Mukherjee, H., Dash, N. S., & Roy, K. (2021). Text categorization: past and present. *Artificial Intelligence Review*, *54*(4), 3007-3054.

Eryigit, G., & Adali, E. (2003). AN AFFIX STRIPPING MORPHOLOGICAL ANALYZER FOR TURKISH.

Eryigit, G., & Oflazer, K. (2006). Statistical Dependency Parsing for Turkish. EACL

Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Kilinç, D., Özçift, A., Bozyigit, F., Yildirim, P., Yücalar, F., & Borandag, E. (2017). TTC-3600: A new benchmark dataset for Turkish text categorization. *Journal of Information Science*, *43*, 174 - 185.

Köksal, A. (2018). *Turkish Pre-trained Word2Vec Model*. https://github.com/akoksal/Turkish-Word2Vec

Li, C., Lu, Y., Wu, J., Zhang, Y., Xia, Z., Wang, T., . . . Guo, J. (2018). *LDA meets Word2Vec: a novel model for academic abstract clustering.* Paper presented at the Companion proceedings of the the web conference 2018.

Makagonov, P., Alexandrov, M., & Gelbukh, A. (2004). *Clustering abstracts instead of full texts.* Paper presented at the International Conference on Text, Speech and Dialogue.

Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, *5*(4), 1093-1113.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv:1310.4546*.

Onan, A., Bulut, H., & Korukoglu, S. (2017). An improved ant algorithm with LDA-based representation for text document clustering. *Journal of Information Science*, *43*(2), 275-292.

Pennington, J., Socher, R., & Manning, C. D. (2014). *Glove: Global vectors for word representation.* Paper presented at the Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP).

Pinto, D., Rosso, P., & Jiménez-Salazar, H. (2011). A self-enriching methodology for clustering narrow domain short texts. *The Computer Journal, 54*(7), 1148-1165.

Popova, S., Danilova, V., & Egorov, A. (2014). *Clustering narrow-domain short texts using k-means, linguistic patterns and lsi.* Paper presented at the International Conference on Analysis of Images, Social Networks and Texts.

Premalatha, K., & Natarajan, A. (2010). A literature review on document clustering. *Information Technology Journal, 9*(5), 993-1002.

Rodriguez, M. Z., Comin, C. H., Casanova, D., Bruno, O. M., Amancio, D. R., Costa, L. d. F., & Rodrigues, F. A. (2019). Clustering algorithms: A comparative approach. *PloS one, 14*(1), e0210236.

Romeo, S., Greco, S., & Tagarelli, A. (2014). *Multi-topic and multilingual document clustering via tensor modeling.*

Tajbakhsh, M. S., & Bagherzadeh, J. (2019). Semantic knowledge LDA with topic vector for recommending hashtags: Twitter use case. *Intelligent Data Analysis, 23*(3), 609-622.

Torunoğlu, D., Çakirman, E., Ganiz, M. C., Akyokuş, S., & Gürbüz, M. Z. (2011). Analysis of preprocessing methods on classification of Turkish texts. 2011 International Symposium on Innovations in Intelligent Systems and Applications

Tuncelli, O., & Özdemir, B. (2019). *Turkish Stemmer for Python*. https://github.com/otuncelli/turkish-stemmer-python

Uysal, A. K., & Gunal, S. (2014). The impact of preprocessing on text classification. *Information Processing & Management, 50*(1), 104-112.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., . . . Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems, 30*.

Weißer, T., Saßmannshausen, T., Ohrndorf, D., Burggräf, P., & Wagner, J. (2020). A clustering approach for topic filtering within systematic literature reviews. *MethodsX, 7*, 100831.

Yang, J., & Park, S.-Y. (2002). Email categorization using fast machine learning algorithms. International Conference on Discovery Science