Advances in Artificial Intelligence Research (AAIR)
ISSN 2757-7422
www.dergipark.com/aair/

Vol. 3 (No. 1), pp. 1-12, 2023
doi: 10.54569/aair.1142568
Published online: Feb 15, 2023

Semantic Similarity Comparison Between Production Line Failures for Predictive Maintenance

Hilal Tekgöz [1,*], Sevinç İlhan Omurca[1], Kadir Yunus Koç [2], Umut Topçu [3], Osman Çelik [2]

[1] Kocaeli University, Department of Computer Engineering;
[2] Ibss Consulting, Department of Research and Development;
[3] Vestel Beyaz Eşya San. ve Tic. A.Ş., Department of Research and Development

Abstract

With the introduction of Industry 4.0 into our lives and the creation of smart factories, predictive maintenance has become even more important. Predictive maintenance systems are often used in the manufacturing industry. On the other hand, text analysis and Natural Language Processing (NLP) techniques are gaining a lot of attention by both research and industry due to their ability to combine natural languages and industrial solutions. There is a great increase in the number of studies on NLP in the literature. Even though there are studies in the field of NLP in predictive maintenance systems, no studies were found on Turkish NLP for predictive maintenance. This study focuses on the similarity analysis of failure texts that can be used in the predictive maintenance system we developed for VESTEL, one of the leading consumer electronics manufacturers in Turkey. In the manufacturing industry, operators record descriptions of failure that occur on production lines as short texts. However, these descriptions are not often used in predictive maintenance work. In this study, semantic text similarities between fault definitions in the production line were compared using traditional word representations, modern word representations and Transformer models. Levenshtein, Jaccard, Pearson, and Cosine scales were used as similarity measures and the effectiveness of these measures were compared. Experimental data including failure texts were obtained from a consumer electronics manufacturer in Turkey. When the experimental results are examined, it is seen that the Jaccard similarity metric is not successful in grouping semantic similarities according to the other three similarity measures. In addition, Multilingual Universal Sentence Encoder (MUSE), Language-agnostic BERT Sentence Embedding (LAbSE), Bag of Words (BoW) and Term Frequency - Inverse Document Frequency (TF-IDF) outperform FastText and Language-Agnostic Sentence Representations (LASER) models in semantic discovery of error identification in embedding methods. Briefly to conclude, Pearson and Cosine are more effective at finding similar failure texts; MUSE, LAbSE, BoW and TF-IDF methods are more successful at representing the failure text.

Keywords: *Predictive maintenance; natural language processing; sentence similarity; word representation methods.*

1. Introduction

With the huge increase in text data generated over time, Natural Language Processing (NLP) has attracted strong attention in the field of Artificial Intelligence (AI). Today, measuring similarity between sentences, paragraphs or documents accurately enables us to develop successful implementations in a wide range of NLP tasks such as information retrieval, keyword extraction, text summarization, text clustering, question answering or predictive maintenance (PdM). In the early literature, text representation was mostly based on Bag of Words (BoW) and Term Frequency - Inverse Document Frequency (TF-IDF) techniques. As a result of these representations, two texts were considered similar as long as they contained the same terms [1]. However even though BoW and TF-IDF provide how often the words appear in a given document and how distinctive they are, they are unable to capture the word meaning. To overcome these drawbacks of classical document representations, representations based on modern word embeddings that can capture semantic features and linguistic relationships between words using deep neural networks have gained importance [2]. Document representations based on classical approaches are sparse and ignore the semantic and syntactic similarity between words [3]. In contrast, word embedding methods such as Word2Vec [4], Glove [5], FastText [6] map variable length text to dense vectors and overcome the problem of lack of syntactic and semantic information in representations. However, all these models ignore the context of the words so they are defined as context unaware representations.

*Corresponding author
E-mail address: hilaltekgoz@outlook.com.tr

Received: 19/Jul/2022; Received in revised form: 20/Sep/2022; Accepted: 02/Nov/2022.

Against this, Bidirectional Encoder Representations from Transformers (BERT) [7] provides an effective way to contextualized representation of documents due to the fact that it generates vector representations for a word depending on its context. The similarity of the document vectors can be calculated using a distance measure such as Cosine similarity which has been widely used among NLP researchers to date [8].

Although many text mining jobs and NLP studies are gaining importance in industrial applications, there are few NLP studies in the field of PdM. Most of the systems in PdM aim to predict the potential failures before they occured. And several machine learning and deep learning methods are used for these kinds of solutions in the related literature. Essentially, in PdM systems, the explanations of the failures occurring in the production lines are recorded as text information by the operators. Analyzing the failure texts can provide a deep insight for PdM. Further, they can be considered as a valuable source of getting information about human feedback. Consequently, failure descriptions give more opportunity to explore valuable insights such as estimating unplanned downtimes which cannot be achieved from quantitative data. As known, an unplanned downtime occurs when there is an unexpected shutdown or failure of equipment or process. An unplanned downtime causes delays in regular production or maintenance schedules which can be extremely costly and inconvenient for a business operation.

In our study, we have the basic motivation to estimate the possible downtimes in minutes through the explanation text of a failure that occurs in the washing machine production line of VESTEL which is a manufacturing company for consumer electronics in Turkey. For this purpose, it is aimed to calculate the similarities between the explanation text of a newly occurring failure and the explanation texts of the existing past failures on the production lines.

Accurate measurement of similarities between texts strongly relies on more accurate document representations. Thus, in this study, we explored various models such as BoW, TF-IDF, MUSE, LAbSE, FastText and LASER for document representation. On the other hand, to obtain a better document similarity scheme, the efficiency of Pearson Similarity Metric, Cosine Similarity Metric, Jaccard Similarity Metric, and Levenshtein Similarity Metric were also explored. As a result of experiments with the Turkish production failure texts obtained from VESTEL, it was observed that the similarity metrics cosine and pearson gave more successful results, while MUSE, LAbSE models were more effective in document representations.

To the best of our knowledge, our study is the first in the literature to use Turkish NLP solutions in predictive maintenance.

The organization of this article is as follows: Section 2 will review previous studies in related literature. Section 3 will describe the techniques used. Section 4 will cover the applied models and dataset. Section 5 will discuss the conclusion and the future work.

2. Related Studies

Any unplanned interruption of industry to devices or machinery equipment within the organization can disrupt the core operation of the organization. An unplanned downtime not only causes costly delays and urgent repairs, but can also lead to poor customer experience and satisfaction in the long run. In the literature, PdM systems and posture prediction studies are encountered.

In Akhbardeh et al.'s study, it was aimed to cluster similar fault records using the MaintNet dataset produced for PdM and the DBSCAN clustering method. In the study, Pearson used cosine similarity metrics to examine similarity between clusters [9]. In another study on PDM, the multivariate time series classification method was used. The dataset used here was obtained from 23 sensors. They created an architecture called Conv-MHSA. It was emphasized that this architecture is much more computational efficiency than RNN methods [10]. Wellsandt et al. have announced that digital smart assistant systems for PdM will play an active role with developments in NLP [11].

Semantic similarity analysis is one of the key features of the NLP approaches [11]. Semantic similarity is usually calculated with datasets that include word pairs and a human-assigned similarity score. The versatility of natural language makes it difficult to define rule-based methods for determining semantic similarity measures [12]. Various word representation methods are used when calculating semantic similarities. For example, the Word2Vec method has recently gained attention due to new developments in the use of neural networks to learn low-dimensional, dense vector representation models known as word embedding [13]. Word representation methods provide vector representations for words where the relationship between two vectors reflects the linguistic relationship between two words and aims to capture semantic and syntactic similarities between words [13]. Traditional methods in semantic similarity analysis use the number of words combined to measure the similarity of sentences. However, traditional methods do not include deep similarities in semantic similarity evaluation and problems are encountered. The application of deep learning models in natural language processing tasks greatly improves the accuracy of sentence recognition tasks [14]. Pre-trained high-dimensional word vectors can also learn the context of the sentence. However, it is observed that the complexity of the algorithm increases as the vector gets larger [14]. In their study, Saipech and Seresangtakul converted the answers students entered into the system into vectors using TF-IDF. They used cosine similarity to measure the similarity of the answers. When the results obtained are compared with the scores prepared by expert teachers, it is seen that they give similar results [15]. Ruchindramalee et al. used Word2Vec and cosine similarity techniques to find similarities between social media and news sources. In the study, 70% accuracy was obtained in news sharing on social media compared to reliable online news sources [16]. In a study by Xiaolin et al, Word2Vec and Pearson developed a Chinese word-based approach using the similarity technique. As a result of the study, they obtained a high Pearson correlation coefficient [17]. Pre-trained transformer networks can be used to discover semantic text similarities. BERT, which stands for two-way encoder representations, is used in a variety of tasks such as question answering, semantic similarity, and language extraction. A new BERT model can be created by fine-tuning BERT models with just one additional output layer.

3. Materials and Methods

3.1. Similarity Metrics

The Pearson correlation coefficient [18] is a measure of the linear dependence between two random variables (real-valued vectors). Historically, it is the first official measure of correlation. The Pearson correlation coefficient of two variables, x and y, is formally defined as the covariance obtained by dividing the product of the standard deviations of the two variables and is shown in Equation 1.

$$r_{xy} = \frac{\sum(x-\underline{x})\sum(y-\underline{y})}{\sqrt{(x-\underline{x})^2}\sqrt{(y-\underline{y})^2}} \tag{1}$$

If the r_{xy} value defined in Equation 1 is 0, then x and y are said to be uncorrelated. However, the closer the r_{xy} value is to 1, the more it is said that there is a relationship between x and y [19]. According to the general guideline suggested by Evans (1996), the correlation results are classified as in the following [20]:

- very strong: 0.80 – 1.00
- strong: 0.60 – 0.79
- medium: 0.40 – 0.59
- weak: 0.20 – 0.39
- very weak: 0.00 – 0.19

Cosine similarity is created by measuring the angle between two vectors. If the cosine value is 0, it reports that the text snippets are not similar. A high cosine value indicates that the pieces of text are quite similar to each other. This gives a measurement of orientation, not magnitude; can be viewed as a comparison between documents in a normalized space [21]. Equation 1 shows the mathematical formula of cosine similarity.

$$cos\ (\theta) = \frac{A.B}{\|A\|\,\|B\|} \tag{2}$$

Where, parameters A and B represent different document vectors. It is also known as the Jaccard similarity coefficient [22], which is a statistical method used to find the similarity and diversity between two finite sets. It is defined as in Equation 3. In the equation, A and B denote the vectors of failure texts.

$$J(A,B) = \frac{\|A \cap B\|}{\|A \cup B\|} \tag{3}$$

The Levenshtein similarity is a simple edit distance. The distance between two strings is defined as the minimal number of basic operations (insert, delete, or replace) needed to convert one string to another. The ratio between the distance and length of longer strings is considered as the similarity between these strings.

3.2. Document Representation Methods

In traditional document representation methods, a document is considered as a bag of words. BoW was first proposed in text retrieval problems to create the vector representations of text documents [24]. The basic assumption is that the words selected from the documents are expressed using frequency sets in an unordered manner. Thus, categorical data is converted into numerical form [25].

TF-IDF is a commonly used method to convert words to vectors when analyzing documents. The TF-IDF determines the relative frequency of words in a selected document in inverse proportion over the entire document. Here each word is treated as a feature. In determining the value, it uses the frequency of the term in the TF document and the reverse document frequency of the term IDF [26]. In addition, using less features on texts with TF-IDF, higher performance is achieved compared to the word bag method [27].

The Word2Vec method is one of the word representation methods used to represent documents. It understands and vectorizes the meanings of words in a document based on the close distances of words with similar meanings in a given context [28]. There are two main learning algorithms. These are continuous bag-of-words and continuous skip-gram algorithms. With the continuous word bag method, it predicts the current word according to the context. It predicts the surrounding words based on the current word with Skip-gram. Unlike the standard word bag model, continuous wordbags use a distributed representation of context [29].

FastText is an open-source modern word representation method developed by the Facebook research team [6]. FastText is an effective word representation method for learning word representations and text classification. Its purpose is not to learn word representations, but to consider the internal structure of the word. It has been observed that this method gives more effective results in morphologically rich languages. FastText works by sliding a window above the introductory sentence or by using the continuous word bag method. It can be viewed as a series of updates in a neural network containing two layers of weights and three layers of neurons [30]. It is based on the skip-gram sub-method of Word2Vec method. Unlike Word2Vec, words are handled by dividing them into n-grams. Words divided into n-grams are represented vectorally as the sum of n-grams. Other methods that do not deal with n-grams ignore the morphology of the word because they convert each word into a vector. There is a limitation especially for languages with large vocabulary and very rare words. The FastText method overcomes this limitation thanks to the n-gram method.

3.3. Transformers

Transformers have been developed as a solution to the forgetting problem of long inputs. The Encoder-Decoder model developed to avoid long input problems was insufficient. It was created to reduce both long input problems and training time. Transformer is a 6-layer encoder-decoder model that creates a target sequence from the source sequence via the decoder [31]. The encoder and decoder consist of self-attention and a feed-forward layer. In the decoder, it provides a match to the markers with the encoder with an attention layer in between. It is a word representation method based on a masked language model and pre-trained using bidirectional transformers. In word vector generation methods such as Word2vec, GloVe and other neural network models, word vectors are mostly context independent. And to represent the ambiguous nature of words, it becomes harder to learn more relevant information. BERT models can successfully learn semantic features and create vectors according to the semantic differences of words.

In Transformer models, within the scope of this study, LaBSE, MUSE, LASER models were used to define Turkish sentence embeddings. It is possible to retrain models with fine tuning. To briefly explain these methods, LaBSE is a BERT based model trained on 17 billion monolingual sentences and 6 billion bilingual sentence pairs. MUSE is a sentence encoding model simultaneously trained on multiple tasks and 16 languages, including Turkish. LASER is a language model based on the BiLSTM encoder trained on 93 languages, including Turkish.

4. Experiments

4.1. Dataset Description and Preprocessing Steps

In this study, a real dataset obtained from a manufacturing company about consumer electronics in Turkey is used to explore the semantic similarities between the mechanical and electrical failure explanations in the production lines. There are 146 failure records belonging to a special production line in the dataset. The average document length of the failure texts is 30. The attributes of the dataset and some data samples are reflected in **Table 1**. In the failure description column, the abbreviation 'MK' means that the failure in the production line is a mechanical failure. Numerical values in the failure description column, are the identifying numbers of the equipment used in the production lines.

The failure texts in the dataset were constituted with the failure descriptions belonging to a specific Production line entered by the operators when a downtime in the production had occurred. While creating these records it has been considered that the production failures can only be mechanical or electrical.

Preprocessing is a very crucial task in terms of managing and analyzing the text data before it is fed into the model as an input. In our experiments, no stem reduction was performed on the words in the sentence collection. Only letters in the capital were converted to lowercase letters and numerical/special characters were removed. Additionally, first we only evaluated failure descriptions excluding station information. We then extended the experiments by adding textual station information to the failure descriptions.

Table 1. *Data Samples of The "Mechanical-Electrical" Failures Occurring on a Specific ProductionLine*

Shift Information	Start of Downtime	End of Downtime	Downtime in Minutes	Station Information	Failure Description
08:00-20:00	08:00:00	08:25:00	25	3.ist.2.pres	1,2,Tank Minimum Yağ
08:00-20:00	09:50:00	10:10:00	20	4.ist.3.pres	Sıcaklık Alarmı Veriyor . 10001.MK
08:00-20:00	16:25:00	16:35:00	10	2.ist.1.pres.	Sıcaklık Alarmı Veriyor
24:00-08:00	23:00:00	01:00:00	120	15.ist.tds tox	1503 Sök Tak Yapmada
08:00-16:00	10:50:00	11:05:00	15	9.ist.boy bük.kal.	HSV901 Yukarı Pozisyonuna
08:00-16:00	09:50:00	10:05:00	15	15.ist AmorAyağıBük	15.İST 15093 Nolu Sensör
08:00-16:00	11:15:00	12:15:00	60	1.ist.kesik sac bes.	SM101 Kaprin Kırık Hareket

4.2. Algorithm and Results

Recently, semantic discovery of similarities between texts has come to the fore as an important field of natural language processing for meaningful information search and matching. In this study, BoW, FastText, TF-IDF, Word2Vec methods were used to represent the failure descriptions. Other than these, Language-agnostic BERT Sentence Encoder, Multilingual Universal Sentence Encoder, LanguageAgnostic SEntence Representations models were used to represent sentence embeddings. In order to extract semantic similarities between failure text descriptions more accurately, the effectiveness of Cosine, Jaccard, Levenshtein, Pearson similarity scales were investigated. In **Figure 1**, the semantic similarity steps are shown in the diagram. This work was done with a machine with NVIDIA GTX 1660 Ti GPU and 16 GB Memory.

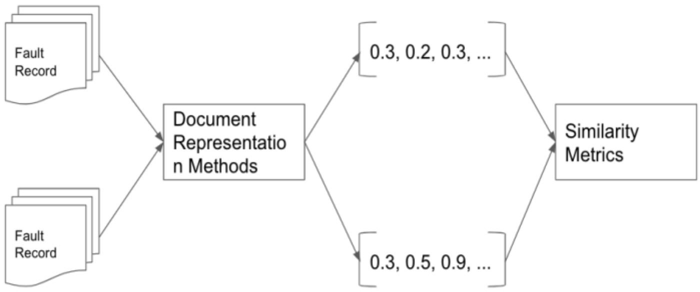

Figure 1. Steps to extract semantic similarities between failures

To investigate the effectiveness of the methods an example failure description text "Sıcaklık alarmı veriyor mk" ("Gives temperature alarm mk") has been considered as a query text. The similar failure descriptions can be extracted according to a user defined threshold value. In our experiments, the threshold value is determined as 70 to define whether the documents are semantically similar. The experimental results in **Tables 2, 3, 4, 5,** respectively, were obtained by BoW, TF-IDF, Word2Vec, FastText methods. The ID column in the tables indicates a unique identifier assigned to failure records. The Downtime in Minutes column shows how long the fault records continue.

Table 2. *Results obtained by BoW and subtracting numerical values*

Similarity Metric	Id	Failure Description	Similarity Score	Downtime in Minutes
Levenshtein	3	sicaklik alarmi veriyor mk	BASE	20
	30	minimum yağ alarmi veriyor mk	71,43	25
	97	minimum yağ alarmi veriyor mk	71,43	15
	132	sicaklik alarmi veriyor	96,00	10
Jaccard	3	sicaklik alarmi veriyor mk	BASE	20
Cosine	3	sicaklik alarmi veriyor mk	BASE	20
	132	sicaklik alarmi veriyor	75,00	10
Pearson	3	sicaklik alarmi veriyor mk	BASE	20
	132	sicaklik alarmi veriyor	75,00	10

Looking at the results in **Table 2**, similar failure descriptions obtained according to Pearson and Cosine scales are 'sicaklik alarmi veriyor mk' (gives temperature alarm-mechanical failure) and 'sicaklik alarmi veriyor'(gives temperature alarm) as the best match results. However, looking at the downtime minutes, it is seen that there is a difference of 10 minutes between them. This shows that if the failure in the production line is caused by the temperature alarm, it will be in the range of 10-20 minutes and the downtimes may differ from each other by 10 minutes. According to Jaccard, no similar failure record was found. According to Levenshtein similarity, 'sicaklik alarmi veriyor' (gives temperature alarm) and 'minimum yağ alarmi veriyor', (gives minimum oil alarm) which are semantically different, were found to be similar records.

Table 3. *Results obtained by TF-IDF and subtracting numerical values*

Similarity Metric	Id	Failure Description	Similarity Score	Downtime in Minutes
Levenshtein	3	sicaklik alarmi veriyor mk	BASE	20
	30	minimum yağ alarmi veriyor mk	71,4	25
	97	minimum yağ alarmi veriyor mk	71,43	15
	132	sicaklik alarmi veriyor	96,00	10
Jaccard	3	sicaklik alarmi veriyor mk	BASE	20
Cosine	3	sicaklik alarmi veriyor mk	BASE	20
	132	sicaklik alarmi veriyor	84,4	10
Pearson	3	sicaklik alarmi veriyor mk	BASE	20
	132	sicaklik alarmi veriyor	84,0	10

When the results in **Table 3** are examined, it is observed that the same failure descriptions are captured with a similarity rate of 0.84 in the Pearson and Cosine similarity scales. When the results reflected in **Table 2** and **Table 3** are examined together, it is observed that the same document similarities are captured with the TF-IDF and BoW representations.

Table 4. *Results obtained obtained by Word2Vec and subtracting numerical values*

Similarity Metric	Id	Failure Description	Similarity Score	Downtime in Minutes
Levenshtein	3	sicaklik alarmi veriyor mk	BASE	20
	30	minimum yağ alarmi veriyor mk	71,43	25
	97	minimum yağ alarmi veriyor mk	71,43	15
	132	sicaklik alarmi veriyor	96,00	10
Jaccard	3	sicaklik alarmi veriyor mk	BASE	20
Cosine	3	sicaklik alarmi veriyor mk	BASE	20
Pearson	3	sicaklik alarmi veriyor mk	BASE	20

Looking at the results of **Table 4**, according to Pearson, Cosine and Jaccard similarity scales, no record matching the failure description of 'sicaklik alarmi veriyor mk' (gives temperature alarm-mechanical failure) was found. It was also found that the failure description 'sicaklik alarmi veriyor' matched with a similarity ratio of 0.96 on the Levenshtein scale. It is understood from the similarity results of Jaccard, Cosine and Pearson that the abbreviations in the fault descriptions (as in the 'mk-mechanical failure' abbreviation here) are not very effective in the similarity calculation when the documents are represented in word2vec.

Table 5. *Results obtained by FastText and subtracting numerical values*

Similarity Metric	Id	Failure Description	Similarity Score	Downtime in Minutes
Levenshtein	3	sicaklik alarmi veriyor mk	BASE	20
	30	minimum yağ alarmi veriyor mk	71,4	25
	97	minimum yağ alarmi veriyor mk	71,43	15
	132	sicaklik alarmi veriyor	96,00	10
Jaccard	3	sicaklik alarmi veriyor mk	BASE	20
Cosine	3	sicaklik alarmi veriyor mk	BASE	20

	28	minimum yağ alarmi veriyor	76,41	20
	30	minimum yağ alarmi veriyor mk	77,43	25
	39	tank minimum yağ alarmi veriyor	72,96	25
	43	pompa minimum yağ alarmi veriyor	73,51	30
	56	tank minimum yağ alarmi veriyor mk	72,03	20
	58	tank minimum yağ alarmi veriyor mk	72,96	15
	59	minimum yağ alarmi veriyor mk	72,96	15
	97	minimum yağ alarmi veriyor mk	77,43	15
	122	minimum yağ alarmi veriyor pompa açilmiyor	75,3	25
	124	pres minimum yağ alarmi veriyor pompa açilmiyor	72,33	20
	132	sicaklik alarmi veriyor	99,82	10
	146	tank minimum yağ alarmi veriyor mk	72,96	25
Pearson	3	sicaklik alarmi veriyor mk	BASE	20
	28	minimum yağ alarmi veriyor	76,47	20
	30	minimum yağ alarmi veriyor mk	77,50	25
	39	tank minimum yağ alarmi veriyor	73,01	25
	43	pompa minimum yağ alarmi veriyor	73,00	30
	56	tank minimum yağ alarmi veriyor mk	72,55	20
	58	tank minimum yağ alarmi veriyor mk	73,96	15
	59	minimum yağ alarmi veriyor mk	72,96	15
	97	minimum yağ alarmi veriyor mk	77,43	15
	122	minimum yağ alarmi veriyor pompa açilmiyor	75,3	25
	124	pres minimum yağ alarmi veriyor pompa açilmiyor	72,33	20
	132	sicaklik alarmi veriyor	84,0	10

According to the results in **Table 5**, more failure descriptions were discovered as similar descriptions by using FastText. However, it was observed that the temperature alarm and the oil alarm cannot be distinguished semantically.

In our experiments, we also investigate the efficiency of transformer models by considering the sentence "Sıcaklık alarmı veriyor mk" as a query sentence. The obtained results for LaBSe, MUSE and LASER are reflected in **Tables 6, 7** and **8**, respectively.

Looking at the results in **Table 6** with Jaccard and Cosine similarity metrics, the two sentences "Sıcaklık alarmı veriyor mk" (gives temperature alarm-mechanical failure) and "Sıcaklık alarmı veriyor" (gives temperature alarm) were discovered to be similar. The LaBSe model succeeded in capturing the sentences that are semantically the most similar.

Table 6. *Results obtained by LaBSe and subtracting numerical values*

Similarity Metric	Id	Failure Description	Similarity Score	Downtime in Minutes
Levenshtein	3	sicaklik alarmi veriyor mk	BASE	20
	30	minimum yağ alarmi veriyor mk	73,77	25
	97	minimum yağ alarmi veriyor mk	73,77	15
	132	sicaklik alarmi veriyor	96,67	10
Jaccard	3	sicaklik alarmi veriyor mk	BASE	20
Cosine	3	sicaklik alarmi veriyor mk	BASE	20
	132	sicaklik alarmi veriyor	96,67	10
Pearson	3	sicaklik alarmi veriyor mk	BASE	20
	132	sicaklik alarmi veriyor	96,67	10

Table 7. *Results obtained by MUSE and subtracting numerical values*

Similarity Metric	Id	Failure Description	Similarity Score	Downtime in Minutes
Levenshtein	3	sicaklik alarmi veriyor mk	BASE	20
	132	sicaklik alarmi veriyor	92,00	10
Jaccard	3	sicaklik alarmi veriyor mk	BASE	20
Cosine	3	sicaklik alarmi veriyor mk	BASE	20
	132	sicaklik alarmi veriyor	90,88	10
Pearson	3	sicaklik alarmi veriyor mk	BASE	20
	132	sicaklik alarmi veriyor	91,00	10

In the results of **Table 7**, it is observed that the Pearson, Cosine, and Levenshtein scales give a higher similarity rate for the failures such as 'sicaklik alarmi veriyor mk' and 'sicaklik alarmi veriyor'. The same results are obtained for LaBSe and MUSE models by using Pearson and Cosine metrics.

Table 8. *Results obtained by LASER and subtracting numeric values*

Similarity Metric	Id	Failure Description	Similarity Score	Downtime in Minutes
Levenshtein	3	sicaklik alarmi veriyor mk	BASE	20
	132	sicaklik alarmi veriyor	92,00	10
Jaccard	3	sicaklik alarmi veriyor mk	BASE	20
Cosine	3	sicaklik alarmi veriyor mk	BASE	20
	30	minimum yağ alarmi veriyor mk	86,43	25
	58	tank minimum yağ alarmi veriyor mk	82,96	15
	59	minimum yağ alarmi veriyor mk	82,21	15
	97	minimum yağ alarmi veriyor mk	86,43	15
	132	sicaklik alarmi veriyor	86,88	10
	146	tank minimum yağ alarmi veriyor mk	86,96	25
Pearson	3	sicaklik alarmi veriyor mk	BASE	20
	30	minimum yağ alarmi veriyor mk	86,00	25
	58	tank minimum yağ alarmi veriyor mk	82,46	15
	59	minimum yağ alarmi veriyor mk	82,11	15
	97	minimum yağ alarmi veriyor mk	86,83	15
	146	sicaklik alarmi veriyor	82,00	25

When the results in **Table 8** are examined, it is realized that the LASER model was not successful in capturing semantic similarities compared to the LaBSe and MUSE models. However, considering the Levenshtein scale, similar failure descriptions seem to be matched. As a result of the LASER model, it was concluded that the downtimes of the faults discovered similarly could be more distant from each other.

The following experimental results were obtained by combining station information and failure descriptions. In the remaining experiments, only Pearson and Cosine similarities, which gave more significant results in previous experiments, were taken into consideration as similarity metrics. Similar to the results obtained above, it was observed that the BoW and TF-IDF models were more successful than the Word2Vec model. As in **Table 5**, the new results obtained with the FastText model are not reflected in **Table 9**, as they fail to capture the semantic relationships between sentences.

Table 9. *BoW, TF-IDF and Word2Vec results obtained by including station information*

Model	Similarity Metric	Id	Failure Description	Similarity Score	Downtime in Minutes
BoW	Cosine	3	sıcaklık alarmi veriyor 10001 mk4.ist3.pres	BASE	20
		132	sıcaklık alarmi veriyor 2.ist.1.pres	79,86	10
	Pearson	3	sıcaklık alarmi veriyor 10001 mk4.ist3.pres	BASE	20
		132	sıcaklık alarmi veriyor 2.ist.1.pres	79.54	10
TF-IDF	Cosine	3	sıcaklık alarmi veriyor 10001 mk4.ist3.pres	BASE	20
		132	sıcaklık alarmi veriyor 2.ist.1.pres	79.86	10
	Pearson	3	sıcaklık alarmi veriyor 10001 mk4.ist3.pres	BASE	20
		132	sıcaklık alarmi veriyor 2.ist.1.pres	79.54	10
Word2Vec	Cosine	3	sıcaklık alarmi veriyor 10001 mk4.ist3.pres	BASE	20
	Pearson	3	sıcaklık alarmi veriyor 10001 mk4.ist3.pres	BASE	20

In **Table 10**, results obtained with LaBSe, MUSE and LASER are summarized. As in the experiments reflected in **Table 9**; failure descriptions were also combined with station information to achieve these results.

Table 10. *LaBSe, MUSE and LASER results obtained by including station information*

Model	Similarity Metric	Id	Failure Description	Similarity Score	Downtime in Minutes
LaBSe	Cosine	3	sıcaklık alarmi veriyor 10001 mk4.ist3.pres	BASE	20
		132	sıcaklık alarmi veriyor 2.ist.1.pres	75.33	10
	Pearson	3	sıcaklık alarmi veriyor 10001 mk4.ist3.pres	BASE	20
		132	sıcaklık alarmi veriyor 2.ist.1.pres	75.00	10
MUSE	Cosine	3	sıcaklık alarmi veriyor 10001 mk4.ist3.pres	BASE	20
		132	sıcaklık alarmi veriyor 2.ist.1.pres	75.33	10
	Pearson	3	sıcaklık alarmi veriyor 10001 mk4.ist3.pres	BASE	20
		132	sıcaklık alarmi veriyor 2.ist.1.pres	75.00	10
LASER	Cosine	3	sıcaklık alarmi veriyor 10001 mk4.ist3.pres	BASE	20
		30	minimum yağ alarmi veriyor mk	86.43	25
		58	tank minimum yağ alarmi veriyor mk	82.96	15
		59	minimum yağ alarmi veriyor mk	82.21	15
		97	minimum yağ alarmi veriyor mk	86.43	15
		132	sicaklik alarmi veriyor	86.88	10
		146	tank minimum yağ alarmi veriyor mk	86.96	25
	Pearson	3	sıcaklık alarmi veriyor 10001 mk4.ist3.pres	BASE	20
		30	minimum yağ alarmi veriyor mk	86.00	25
		58	tank minimum yağ alarmi veriyor mk	82.46	15
		59	minimum yağ alarmi veriyor mk	82.11	15
		97	minimum yağ alarmi veriyor mk	86.83	15
		146	tank minimum yağ alarmi veriyor mk	82.00	25

When the **Table 10** is examined, it has been observed that, LaBSe and MUSE models were more successful than the LASER model in finding the most semantically similar record with query sentence "sıcaklık alarmi veriyor 10001 mk 4.ist3.pres" among 146 failure records.

As a result of extensive experimental studies conducted for different query sentences, it has been observed that LaBSe and MUSE models are superior to LASER, BoW, TF-IDF, Word2vec and FastText models in discovering semantic similarities between failure explanations of production lines. In general, models and methods are good at finding semantic similarities, but sometimes similar fault records have different downtimes. Here, the fault downtime can be specified by taking the average duration of similar matching records.

5. Results and Future Studies

In this study, the semantic text similarities between the failure descriptions in the production lines have been explored for later use in PdM work. A real-life dataset obtained from a manufacturing company about consumer electronics in Turkey has been used for the experiments. The textual explanations of the failures in the production lines were created by the operators at the time of the production downtimes. In this study, while past failure explanations similar to a real time occurred failure explanation are discovered, it is aimed to estimate the downtime of the equipment based on text similarity analysis in future studies.

For this aim, BoW, TF-IDF, MUSE, LAbSE, FastText and LASER models were used to represent the failure descriptions. Besides, Pearson, Cosine, Jaccard, and Levenshtein Similarity Metrics were used to calculate text similarity between failure descriptions. When we analyze the experimental results, it has been observed that Pearson and Cosine are more effective at finding similar failure explanations. Additionally, it is concluded that, MUSE, LAbSE and traditional word representation methods such as BOW and TF-IDF outperform FastText and LASER models in the semantic discovery of failure descriptions. At the same time, different downtimes were found in similar fault records in the estimation of the downtime by taking advantage of the similarities of the fault records. Here, after the similarities are found, the downtime minutes should be determined by taking their averages.

Declaration of interest

It was presented as a summary at the ICAIAME 2022 conference.

Acknowledgements

This work has been supported by TUBİTAK in Turkey under project number 3215073.

References

[1] Chandrasekaran D, and Vijay M. "Evolution of semantic similarity—a survey." ACM Computing Surveys (CSUR) 54.2, 1-37, 2021.

[2] Wang Y, et al. "A comparison of word embeddings for the biomedical natural language processing." Journal of biomedical informatics 87,12-20, 2018.

[3] Liu J, Tianqi L, and Cong Y. “Newsembed: Modeling news through pre-trained document representations”, arXiv preprint arXiv:2106.00590, 2021.

[4] Mikolov T, et al. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781, 2013.

[5] Pennington J, Richard S, and Christopher D.M. “Glove: Global vectors for word representation”. Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014.

[6] Bojanowski P, et al. “Enriching word vectors with subword information”, Transactions of the association for computational linguistics 5, 135-146, 2017.

[7] Devlin J, et al. “Bert: Pre-training of deep bidirectional transformers for language understanding”, arXiv preprint arXiv:1810.04805, 2018.

[8] Mohammad S.M, and Graeme H. “Distributional measures of semantic distance: A survey”, arXiv preprint arXiv:1203.1858, 2012.

[9] Akhbardeh F, Travis D, and Marcos Z. “NLP tools for predictive maintenance records in MaintNet”. Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: System Demonstrations. 2020.

[10] Yang H, Aidan L, and Travis D. “Predictive maintenance for general aviation using convolutional transformers”. Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 36. No. 11. 2022.

[11] Wellsandt S. et al. “Hybrid-augmented intelligence in predictive maintenance with digital intelligent assistants”. Annual Reviews in Control, 2022.

[12] Maulud D.H., et al. “State of art for semantic analysis of natural language processing”, Qubahan Academic Journal 1.2, 21-28, 2021.

[13] Netisopakul P, et al. “Improving the state-of-the-art in Thai semantic similarity using distributional semantics and ontological information”, Plos one 16.2, 2021.

[14] Zhang P, et al. “Semantic similarity computing model based on multi model fine-grained nonlinear fusion”, IEEE Access 9, 8433-8443, 2021.

[15] Saipech P, and Pusadee S. “Automatic Thai subjective examination using cosine similarity”. 2018 5th international

conference on advanced informatics: concept theory and applications (ICAICTA). IEEE, 2018.

[16] Chandrathlake R, et al. “A semantic similarity measure-based news posts validation on social media”. 2018 3rd International Conference on Information Technology Research (ICITR). IEEE, 2018.

[17] Jin X, Shuwu Z, and Jie L. “Word semantic similarity calculation based on word2vec”. 2018 International Conference on Control, Automation and Information Sciences (ICCAIS). IEEE, 2018. Pearson, K. Notes on Regression and Inheritance in the Case of Two Parents Proceedings of the Royal Society of London, 58, 240-242, 1895.

[18] Zhou H, et al. “A new sampling method in particle filter based on Pearson correlation coefficient”. Neurocomputing 216 , 208-215, 2016.

[19] Evans, J.D. “Straightforward statistics for the behavioral sciences”. Thomson Brooks/Cole Publishing Co, 1996.

[20] Jayakodi K., Bandara M., and Meedeniya D. “An automatic classifier for exam questions with WordNet and Cosine similarity”. 2016 Moratuwa engineering research conference (MERCon). IEEE, 2016.

[21] Jaccard P. “Nouvelles recherches sur la distribution florale”. Bull. Soc. Vaud. Sci. Nat., 44, 223-270, 1908.

[22] Levenshtein, V.I. “Binary codes capable of correcting deletions, insertions, and reversals”. Soviet physics doklady. Vol. 10. No. 8. 1966.

[23] Bosch A, Xavier M, and Robert M. “Which is the best way to organize/classify images by content?”. Image and vision computing 25.6, 778-791, 2007.

[24] Aksu, M Ç, and Karaman E. “FastText ve Kelime Çantası Kelime Temsil Yöntemlerinin Turistik Mekanlar İçin Yapılan Türkçe İncelemeler Kullanılarak Karşılaştırılması”. Avrupa Bilim ve Teknoloji Dergisi 20, 311-320, 2020.

[25] Trstenjak B, Sasa M, and Dzenana D. “KNN with TF-IDF based framework for text categorization”. Procedia Engineering 69, 1356-1364, 2014.

[26] Uçar, M.K, Bozkurt M.R, and Bilgin C. “Signal Processing and Communications Applications Conference”. IEEE, 2017.

[27] Jang, B.C, Inhwan K, and Jong W.K. “Word2vec convolutional neural networks for classification of news articles and tweets”. PloS one 14.8, 2019.

[28] Lilleberg J, Yun Z, and Yanqing Z. “Support vector machines and word2vec for text classification with semantic features”. 2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC). IEEE, 2015.

[29] Santos I., Nadia N, and Luiza de Macedo M. “Sentiment analysis using convolutional neural network with fastText embeddings”. 2017 IEEE Latin American conference on computational intelligence (LA-CCI). IEEE, 2017.

[30] Tekgöz H, Çelenli H.İ, and Omurca S.İ. “Semantic Similarity Comparison of Word Representation Methods in the Field of Health”. 2021 6th International Conference on Computer Science and Engineering (UBMK). IEEE, 2021.

[31] Singh S, and Ausif M. “The NLP cookbook: modern recipes for transformer based deep learning architectures”. IEEE Access 9, 68675-68702, 2021.

[32] Feng F, et al. “Language-agnostic bert sentence embedding”. arXiv preprint arXiv:2007.01852, 2020.

[33] Yang Y, et al. "Multilingual universal sentence encoder for semantic retrieval." arXiv preprint arXiv:1907.04307, 2019.