Received: November 16, 2020. Revised: January 20, 2021.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Semantic Recommender System Based on Semantic Similarity Using FastText and Word Mover's Distance

Nabil Haidarrahman Pribadi[1] Riyanarto Sarno[1]* Adhatus Solichah Ahmadiyah[1]
Kelly Rossa Sungkono[1]

[1]*Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia*
* Corresponding author's Email: riyanarto@if.its.ac.id, kelly@its.ac.id

Abstract: Lecturers and students use research papers as references for their projects, such as research, final projects or theses. Obtaining suitable papers is critical for the success of their projects. Nonetheless, finding the correct papers, especially in determining the correct words to paper searching, is challenging. This paper proposes a method that combines FastText and Word Mover's Distance for a recommender system to solve the challenge. FastText and Word Mover's Distance are utilized to gather word embedding and reach semantic similarity, respectively. Data on papers from the IEEE Xplore Digital Library, ACM Digital Library, Science Direct, SpringerLink, and Wiley Digital Library are collected as case studies in this paper. Experiments for verifying the proposed method referring to seven scenarios in which additional three types of queries. The first query applies the original word from the case studies, whereas the second query operates the original word with one word or two words chosen by a user. The third query uses the original word-combining with chosen words based on the proposed method. After implementing these scenarios, most of the results indicate that results produced by the proposed method gain higher accuracy and F-1 score than user query and original query. The proposed method increases a mere 2.3 percent of accuracy and 1.7 percent of F-1.

Keywords: Semantic similarity, Natural language processing, FastText, Word Mover's distance.

1. Introduction

Lately, the development of technology has gone rapidly. The rapid development is proven by many results of research publications conducted by researchers around the world. The results of the research are usually published on the website of the paper publisher. Websites that provide the paper among others are IEEE for electronic engineering and electrical engineering, PubMed for life sciences and biomedical topics, ScienceDirect for scientific and medical research, and many other sources besides them.

Papers that are available on websites can be used by other researchers and students who need references in conducting research, assignments, or thesis. To search for papers on those websites, researchers or students usually use a query that can be done on each website to find the desired paper according to the advised search.

However, there is a problem faced by researchers and students when seeking papers on those websites, which is they cannot determine the keywords of the paper. This problem can be solved using the Natural Language Processing (NLP) approach.

NLP is a subfield of artificial intelligence, linguistics, information engineering, and artificial intelligence concerning the interaction between computers and human languages. Several studies that use the implementation of NLP include a recommender system to produce recommended items to users, semantic matching to calculate matching similarity of words in a document with other words in another document based on semantic similarity, and word embedding to provide vectors of word representations. This research proposed to build a recommender system based on semantic similarity using word embedding.

There are several implementations of the recommender system, such as content-based filtering [1], collaborative filtering [2], dan hybrid filtering [3]. Previous research has been conducted about computer science publications [4], collaborators [5], financial service [6]. Previous research on recommender system provides book recommendations to user using queries from the Neo4j graph database [7]. The research used metadata from book content as an input to provide recommendations which is the implementation of content-based filtering. This research uses the idea from the previous research to implement the content-based filtering that uses metadata from papers.

Word embedding is the type of word representation for mapping the word to vectors by capturing the context of a word in a vocabulary. There are several methods to create word embedding, some of which are GloVe [8], Word2vec [9], and FastText [10]. Previous research on word embedding uses this method for query expansion [11] and classification for sentiment analysis [12]. From those previous researches, they implement machine learning models from text used to be a vocabulary for creating word embedding. From that idea, this research uses word embedding as an input in implementing semantic similarity to provide recommendation items for users. We use data from paper as a corpus.

Semantic similarity implementation is based on word embedding or ontology [13, 14]. Previous research on semantic similarity used the method for aspect-based sentiment analysis [15]. Those previous research proposed cosine similarity to determine the relevance of texts between two different documents [16]. Cosine similarity used text as a vector and similarity between two texts obtained by finding the cosine between the term vectors of the two texts. However, this method remained unable to handle the semantic meaning of the text correctly. For example, "Jokowi had a conversation with Vladimir Putin in Jakarta" and "The President of Indonesia meets the President of Russia in Indonesia capital city" will find little results when using cosine similarity. This is because the two texts, if viewed per word character, are significantly different. From this idea, this research uses Word Mover's Distance (WMD) [17] to determine semantic similarity between two texts.

From the previous research, we proposed a combination of FastText and WMD to build semantic similarity. FastText, as a word embedding method, is chosen because unlike Word2vec and GloVe, it can produce word-to-word vectors that are not in the corpus. WMD method is selected as a semantic similarity method in this research because compared to using cosine similarity, WMD looks for dissimilarities between words from both texts then moves to other words to find the minimum of the dissimilarity. This research compares results based on the proposed method and results based on choice words of users by measuring their accuracy and F-1 score.

Furthermore, this paper has several sections as follows; section 2 explains the theories related to the proposed method in this paper; section 3 contains an explanation of the proposed methods in this paper; section 4 describes the results and analysis of the experiments in this paper; finally, section 5 concludes this paper.

2. Related theory

This section explains the theories related to this research.

2.1 Recommender system

The recommender system is an information filtering system that seeks to predict the item for user by using user preferences or other user ratings [18]. The recommender system has several methods, such as content-based filtering, collaborative filtering, and hybrid filtering. Content-based filtering uses information from existing data. Collaborative filtering uses the preferences and ratings of other users to give recommendations. Hybrid filtering combines both content-based filtering and collaborative filtering.

2.2 Semantic Similarity

Semantic similarity is a calculation between two documents or texts based on the meaning similarity or semantic content that is different from similarity which usually uses a syntactic representation (of string format). Semantic similarity is crucial in cases, such as plagiarism, automatic technical surveys, and semantic search [19]. So, in this research, the use of semantic similarity aims to search for documents by considering the same meaning semantically in a query. This research use FastText method to generate vector of representation words and Word Mover's Distance to obtained Word Mover's distance.

2.3 Pre-processing

Pre-processing is the process of eliminating clutter found in text [20]. There is several pre-processing conducted in this research which can be seen in Table 1.

Table 1. Pre-processing description

Pre-processing	Description
Lowercase	This process converts the entire word contained in the text into a lowercase text.
Remove punctuation	This process removes all the punctuation contained in the text
Remove stopwords	This process removes all stopwords contained in the text
Lemmatization	This process changes the words contained in the text into its basic words.

2.4 Skip-gram model

Skip-gram model is one of the models that currently use in word embedding model like Word2vec [21] and FastText [10]. For obtaining word vector, skip-gram model uses the surrounding of the target word to generate the word vector representation from it [9].

The objective of training skip-gram model is to find word representations from the surrounding words for target words. For example, given the sequence of training words $w_1, w_2, w_3, \dots, w_T$, the equation to maximize the average of log probability represents in Eq. (1).

$$LP = \frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le j\le c, j\neq 0}\log p\left(w_{t+j}\middle|w_t\right) \quad (1)$$

where: LP = average log probability
w_t = sequence of training word at number t
c = size of the training context

The increase of the c in Eq. (1) affects the accuracy to be better. The basic Skip-gram equation defines $p(w_{t+j}|w_t)$ using softmax function represents in Eq. (2).

$$p(w_O|w_I) = \frac{\exp({v'_{w_O}}^{\top} v_{w_I})}{\sum_{w=1}^{W}\exp({v'_w}^{\top} v_{w_I})} \quad (2)$$

where: $p(w_O|w_I)$ = probability of w_O to w_I
v_w = input of word representation
v'_w = output of word representation of vector w
w_O = context word
w_I = center word
W = total number of words in vocabulary

2.5 Fasttext

FastText is a method for generating word vector representations to be an input for machine learning methods. FastText can implement the unsupervised learning or supervised learning algorithm in data owned by the user and produce a model that contains the results of the vector in the input data when implementing the unsupervised and supervised learning algorithm. FastText learns words from subword information. Each word w can be represented as a collection of -gram characters. Added special symbols < and > at the beginning and end of words. For example, in the word computing and $n = 3$, it can be represented as follows.

$$< com, omp, mpu, put, uti, tin, ing >$$

and the order becomes:

$$< computing >.$$

The vector of word embedding for computing is the sum of all n-gram that appears. Suppose that given a dictionary of n-grams size of N. In the word w, we showed that $N_w \in \{1, \dots, N\}$ is the set of n-grams appearing in word w. We represent vector representation as z_n for each n-gram in n. Scoring function represents in Eq. (3).

$$s(w, c) = \sum_{n\in N_w} {z_n}^{\top} v_c \quad (3)$$

where: $s(w, c)$ = scoring function
N_w = set of n-grams appearing in word w
z_n = vector representation for each n-gram
c = context position
v = vector representation of w
W = total number of words in vocabulary

In this research, the implementation of FastText uses the Gensim library from python programming language to produce vectors of word representation.

2.6 Word mover's distance

Word Mover's Distance (WMD) is a function of the distance between one document and another. WMD calculates the dissimilarity between two text documents as a minimum amount of distance so that words embedded in one document need to "move" to reach words embedded in another document. WMD has been applied to several case studies, including analysis of network logs and document summarization.

To calculate the semantic similarity between document, document represented as vector by using FastText method. If the word w_i appears tf_i time in a document, the weight represents in Eq. (4).

$$d_i = \frac{tf_i}{\sum_{i'=1}^{n} tf_{i'}} \quad (4)$$

where: d_i = weight of document
tf_i = appearance time of word w_i
n = number of all words in the document

The higher the results, the more the important the word. To calculate WMD between w_i and w_j, the equation represents in Eq. (5).

$$WMD(i,j) = \| x_i - x_j \|_2 \quad (5)$$

where: WMD = Word Mover's Distance
x_i = weight of w_i

This research applies WMD Similarity (WMDS). The formula of $WMDS$ can be defined in Eq. (6).

$$WMDS = \frac{1}{1+WMD} \quad (6)$$

where: $WMDS$ = Word Mover's Distance Similarity
WMD =Word Mover's Distance

Based on Eq. (6), $WMDS$ uses the results of WMD to be calculated in the equation to determine the similarity between two texts. The implementation of WMDS uses the Gensim library from python programming language.

3. Research method

The stages carried out in this section are presented in Fig. 1. Based on this figure, there are four stages comprising Data Preparation, Data Pre-processing, Training Corpus, and Semantic Similarity. These stages are carried out sequentially from Data Preparation to Semantic Similarity. Each stage results an output that is an input to the next stages.

3.1 Data preparation

At this stage, data preparation was done by crawling data from websites that provide papers, such as the ACM Digital Library, IEEE Xplore Digital Library, Science Direct, Springer Link, and Wiley Online Digital Library. The data taken from these websites were in the form of metadata consisting of Title, Abstract, and Keywords. The description of the metadata can be seen Table 2.

Upon the collection of metadata, we labelled the data. The label was used to divide the dataset into two parts. The first part was labelled positive that discussed the query which had been searched and the second part was labelled negative data that discussed the query which had been sought. For data labelling, we used the systematic literature review from query process mining [22], semantic search [23], and mixed reality [24]. For query COCOMO, we used the dataset which had been collected. The distribution of data can be seen in Table 4.

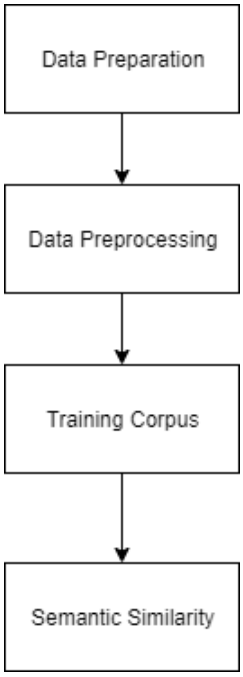

Figure. 1 Method stages

The dataset in this research was collected by query with advanced search on those websites. This research used a specific query to find research papers. These queries have a description which can be seen in Table 3. These queries were chosen based on data in the systematic literature review [23-25].

3.2 Data pre-processing

Before the data were trained using the FastText method to produce word vectors, the data had to be pre-processed first. Pre-processing was done by doing Lowercase, removing punctuation, removing stop words, and Lemmatization. The data that had been pre-processed would be applied to the training corpus. For training corpus, metadata such as Title, Abstract, and Keywords would be combined and used as an input for new column called Content. Pre-

Table 2. Papers metadata

Name	Description
Title	Title of collected research papers
Abstract	Summaries of articles, theses, reviews, conference proceedings, or in-depth analysis of a particular subject and are often used to help readers quickly understand the purpose of this paper.
Keywords	Words that represent articles, thesis, reviews, and conference processing to facilitate searching.

Table 3. Query description

Name	Description
COCOMO	Constructive Cost Model (COCOMO) is a procedural cost estimate model for software projects and currently used as a process of predicting various parameter such as size, effort, cost, time, and quality.
Process Mining	Process mining is a technique in process management for conducting analysis for business process using event logs.
Semantic Search	Semantic search is a system for search engine to capture the words that have a similar meaning by understanding the context from the searchers to generate relevant results.
Mixed Reality	Mixed reality (MR) is the combination of virtual reality (VR) and augmented reality (AR) to produce new environments and visualizations for user by combining physical and digital objects.

Table 4. Distribution of dataset

Name	Positive	Negative
COCOMO	105	23
Process Mining	42	14
Semantic Search	46	16
Mixed Reality	43	9

process Content column was done consisting of Lowercase, removing stopwords, removing punctuation, and Lemmatization.

3.3 Training corpus

We propose FastText to generate word vectors from the Content column. It is used to obtain vector results of word representations based on the collected data. The algorithm for the training FastText method can be seen in Fig. 2.

3.4 Semantic similarity

In this stage, the authors proposed the Word Mover's Distance (WMD) method. This method used input from the pre-processed data results, queries, and corpus training results using the FastText method. The algorithm of this stage can be seen in Fig. 3.

Start
1. Using fastText from the Gensim library
2. Input the Content column in FastText
3. Set iteration parameters, windows, and dimensions
4. Running fastText

End

Figure. 2 Fasttext algorithm

Start
1. Implement the Word Mover's Distance method from the Gensim library
2. Perform training methods by entering papers data and vector results
3. Enter the query to the Word Mover's Distance method
4. Generate Recommendations

End

Figure. 3 WMD algorithm

4. Results and analysis

This section explains the results of the experiments which had been done in the previous section. We evaluated the results of the experiments using Accuracy (*A*), Precision (*P*), Recall (*R*), and F-1 Score (*F*) which had an input of True Positive (*TP*), True Negative (*TN*), False Positive (*FP*), False Negative (*FN*) as formulated in Eq. (7), Eq. (8), Eq. (9) and Eq. (10), respectively.

$$A = \frac{TP+TN}{TP+FP+TN+FN} \tag{7}$$

$$P = \frac{TP}{TP+FP} \tag{8}$$

$$R = \frac{TP}{TP+FN} \tag{9}$$

$$F = 2 \cdot \frac{P \cdot R}{P+R} \tag{10}$$

We evaluated the proposed method using seven scenarios as seen in Table 5. We compared the results between original queries, user queries, and the proposed method. The original queries contain an original word for each scenario. The user queries consist of an original word which is merged with one word or two words chosen by users. Users choose words by considering the meaning of the original word. Furthermore, the proposed method implements Gensim library to obtain the combination words for the original word.

Depend on Table 6, results of evaluations for each scenario verify proposed combination methods obtain higher accuracy and F-1 than both original queries and user queries in mostly scenarios. Apart from Mixed Reality, proposed combination methods increases an average of 2.5% accuracy and 1.9% F-1 compared with user queries. In addition, the results of proposed combination methods rise an average of 2.1% accuracy and 1.6% F-1 compared with those by original queries. The accuracy and F-1 of the proposed method are as high as user queries. In conclusion, the proposed method increases a mere 2.3

Table 5. Query for scenarios

Scenarios	Query	Query Examples			
		COCOMO	**Process Mining**	**Semantic Search**	**Mixed Reality**
1	Original word	COCOMO	Process Mining	Semantic Search	Mixed Reality
2	Original word + 1 word chosen by user	COCOMO + cost	Process Mining + business	Semantic Search + query	Mixed Reality + virtual
3	Original word + the 1st most similar word chosen by Proposed Combination Methods	COCOMO + computing	Process Mining + processing	Semantic Search + searching	Mixed Reality + interaction
4	Original word + the 10th most similar word chosen by Proposed Combination Methods	COCOMO + measurement	Process Mining + representation	Semantic Search + relation	Mixed Reality + visualization
5	Original word + the 20th most similar word chosen by Proposed Combination Methods	COCOMO + modifiability	Process Mining + organizational	Semantic Search + searchable	Mixed Reality + reconstruction
6	Original word + 2 words chosen by user	COCOMO + cost + constructive	Process Mining + business + techniques	Semantic Search + query + engine	Mixed Reality + virtual + realization
7	Original word + two words of the 1st and 10th most similar word by Proposed Combination Methods	COCOMO + computing + measurement	Process Mining + processing + representation	Semantic Search + searching + relation	Mixed Reality + interaction + visualization

percent of accuracy and 1.7 percent of F-1.

On the other hand, all results, including by original queries, user queries and the proposed method, has 1 of precision. The same precision number is obtained because the negative labelled data did not mainly discuss the query. For example, in the COCOMO negative labelled data, there is a paper which focused on the virology method to proviral DNA named BLV-CoCoMo-qPCR-2 [25]. This paper did not mainly discuss the majority paper focusing on COCOMO in the field of information technology, not virology. This case also occurred in another query.

The main reason for results obtained by users are lower accuracy than those by the proposed method is the words selected by users are not found in the data. For example, in scenario 6 of COCOMO, recall, and F-1 score results are worse than scenario 7 because the word 'constructive' is not in the vocabulary.

5. Conclusion

This research proposes the combination of FastText and Word Mover's Distance for building semantic similarity method. Fast Text method produces vectors of word representations and Word Mover's Distance method to determine the semantic similarities between query and paper.

The results of the proposed method indicate high accuracy and F-1 score. The proposed method increases an average of 2.5% accuracy and 1.9% F-1 compared with user queries. Then, the results of proposed method rise an average of 2.1% accuracy and 1.6% F-1 compared with those by original queries. The proposed method increases a mere 2.3 percent of accuracy and 1.7 percent of F-1.

The comparison results of this research are based on choices of users; hence this research should be

Table 6. Results from each scenarios

Query	Results	Query Examples						
		1	**2**	**3**	**4**	**5**	**6**	**7**
COCOMO	**TP**	96	96	97	97	97	93	97
	FP	0	0	0	0	0	0	0
	TN	23	23	23	23	23	23	23
	FN	9	9	8	8	8	12	8
	A	0.930	0.930	**0.938**	**0.938**	**0.938**	0.906	**0.938**
	P	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	R	0.914	0.914	**0.924**	**0.924**	**0.924**	0.886	**0.924**
	F	0.955	0.955	**0.960**	**0.960**	**0.960**	0.939	**0.960**
Process Mining	**TP**	36	36	38	38	38	36	38
	FP	0	0	0	0	0	0	0
	TN	16	16	14	14	14	16	14
	FN	6	6	4	4	4	6	4
	A	0.897	0.897	**0.929**	**0.929**	**0.929**	0.897	**0.929**
	P	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	R	0.857	0.857	**0.905**	**0.905**	**0.905**	0.857	**0.905**
	F	0.923	0.923	**0.950**	**0.950**	**0.950**	0.923	**0.950**
Semantic Search	**TP**	41	41	42	42	42	41	42
	FP	0	0	0	0	0	0	0
	TN	16	16	16	16	16	16	16
	FN	5	5	4	4	4	5	4
	A	0.919	0.919	**0.935**	**0.935**	**0.935**	0.919	**0.935**
	P	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	R	0.891	0.891	**0.913**	**0.913**	**0.913**	0.891	**0.913**
	F	0.943	0.943	**0.955**	**0.955**	**0.955**	0.943	**0.955**
Mixed Reality	**TP**	38	39	39	39	39	39	39
	FP	0	0	0	0	0	0	0
	TN	9	9	9	9	9	9	9
	FN	5	4	4	4	4	4	4
	A	0.904	**0.923**	**0.923**	**0.923**	**0.923**	**0.923**	**0.923**
	P	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	R	0.884	**0.907**	**0.907**	**0.907**	**0.907**	**0.907**	**0.907**
	F	0.94	**0.95**	**0.95**	**0.95**	**0.95**	**0.95**	**0.95**

here: TP = True Positive
FP = False Positive
TN = True Negative
FN = False Negative
A = Accuracy
P = Precision
R = Recall
F = F-1 Score

developed by other combination methods, such as Glove-WDM or FastText-Cosine Similarity, to reinforce the greatness of proposed combination methods.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, R. Sarno and N.H. Pribadi; supervision, R. Sarno; Methodology, R. Sarno, N.H. Pribadi, and A.S. Ahmadiyah; Experiment, N.H. Pribadi; Validation, A.S. Ahmadiyah; Writing — original draft preparation, N.H. Pribadi and K.R. Sungkono; Writing — review and editing, K.R. Sungkono.

Acknowledgments

This research was funded by the Ministry of Research and Technology/National Research and Innovation Agency Republic of Indonesia (Ristek-BRIN) and the Indonesian Ministry of Education and Culture under Newton Institutional Links - Kerjasama Luar Negeri (KLN) Program, and under Riset Inovatif-Produktif (RISPRO) Invitation Program managed by Lembaga Pengelola Dana Pendidikan (LPDP) also under Penelitian Terapan Unggulan Perguruan Tinggi (PTUPT) Program managed by Institut Teknologi Sepuluh Nopember (ITS).

References

[1] C. C. Aggarwal, Recommender Systems, New York: Springer, 2016.

[2] B. Schafer, D. Frankowski, and S. Sen, "Collaborative Filtering Recommender Systems", *The Adaptive Web*, pp. 291-324, 2007.

[3] G. Geetha, M. Safa, C. Fany, and D. Saranya, "A Hybrid Approach using Collaborative filtering and Content based Filtering for Recommender System", *Journal of Physics Conf. Series*, pp. 1-7, 2018.

[4] D. Wang, Y. Liang, D. Xu, X. Feng, and R. Guan, "A content-based recommender system for computer science publications", *Knowledge-Based Systems*, Vol. 157, No. 18, pp. 1-9, 2018.

[5] H.-H. Chen, L. Gou, X. Zhang, and C. L. Giles, "CollabSeer: A Search Engine for Collaboration Discovery", In: *Proc. of the 2011 Joint International Conf. on Digital Libraries*, pp. 231-240, 2011.

[6] A. Felfernig, K. Isak, K. Szabo, and P. Zachar, "The VITA Financial Services Sales Support Environment", In: *Proc. of the 19th National Conf. on Innovative Applications of Artificial Intelligence*, pp. 1692-1699, 2007.

[7] I. N. P. W. Dharmawan and R. Sarno, "Book Recommendation Using Neo4j Graph Database in BibTeX Book Metadata", In: *Proc. of 2017 3rd International Conf. on Science in Information Technology (ICSITech)*, pp. 47-52, 2017.

[8] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global Vectors for Word Representation", In: *Proc. of the 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532-1543, 2014.

[9] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space", In: *Proc. of the International Conf. on Learning Representations (ICLR 2013)*, pp. 1-12, 2013.

[10] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information", *Transactions of the Association for Computational Linguistics*, Vol. 5, pp. 135-146, 2017.

[11] M. Maryamah, A. Z. Arifin, R. Sarno, and Y. Morimoto, "Query Expansion Based on Wikipedia Word Embedding and BabelNet Method for Searching Arabic Documents", *International Journal of Intelligent Engineering and Systems*, Vol. 12, No. 5, pp. 202-213, 2019.

[12] R. A. Priyantina and R. Sarno, "Sentiment Analysis of Hotel Reviews Using Latent Dirichlet Allocation, Semantic Similarity and LSTM", *International Journal of Intelligent Engineering and Systems*, Vol. 12, No. 4, pp. 142-155, 2019.

[13] M. Gan, X. Dou, and R. Jiang, "From Ontology to Semantic Similarity: Calculation of Ontology-Based Semantic Similarity", *The Scientific World Journal*, Vol. 2013, No. 10, pp. 1-11, 2013.

[14] A. Arwan, B. Priyambadha, R. Sarno, M. Sidiq, and H. Kristianto, "Ontology and Semantic Matching for Diabetic Food Recommendations", In: *Proc. of International Conf. on Information Technology and Electrical Engineering (ICITEE)*, pp. 170-175, 2013.

[15] F. Nurifan, R. Sarno, and K. R. Sungkono, "Aspect Based Sentiment Analysis for Restaurant Reviews Using Hybrid ELMo-Wikipedia and Hybrid Expanded Opinion Lexicon-SentiCircle", *International Journal of*

Intelligent Engineering and Systems, Vol. 12, No. 6, pp. 47-58, 2019.

[16] D. Gunawan, C. A. Sembiring, and M. A. Budiman, "The Implementation of Cosine Similarity to Calculate Text Relevance between Two Documents", In: *Proc. of 2nd International Conf. on Computing and Applied Informatics 2017,* pp. 1-6, 2017.

[17] M. J. Kusner, Y. Sun, N. I. Kolkin, and K. Q. Weinberger, "From Word Embeddings To Document Distances", In: *ICML'15 Proc. of the 32nd International Conf. on International Conf. on Machine Learning,* pp. 957-966, 2015.

[18] F. Ricci, L. Rokach, and B. Shapira, *Introduction to Recommender Systems Handbook*, Boston, MA: Springer, 2010.

[19] W. H. N. Putra, Sugiyanto, R. Sarno, and M. Sidiq, "Weighted Ontology and Weighted Tree Similarity Algorithm for Diagnosing Diabetes Mellitus", *In: Proc. of International Conf. on Computer, Control, Informatics and its Applications (IC3INA),* pp. 267-272, 2013.

[20] W. S. Bhaya, "Review of Data Preprocessing Techniques in Data Mining", *Journal of Engineering and Applied Sciences,* Vol. 12, No. 16, pp. 4107-4107, 2017.

[21] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality", In: *Proc. of the 26th International Conf. on Neural Information Processing Systems,* pp. 3111-3119, 2013.

[22] R. Kelemen, "Systematic Review on Process Mining and Security", *Central and Eastern European e|Dem and e|Gov Days 2017,* pp. 1-14, 2017.

[23] J. M. Vidal and A. Melgar, "Research on Proposals and Trends in the Architectures of Semantic Search Engines: A Systematic Literature Review", In: *Prof. of 2017 Federated Conf. on Computer Science and Information Systems,* pp. 271-280, 2017.

[24] C. M. Y. Rasimah, M. Nurazean, M. D. Salwani, M. Z. Norziha, and I. Roslina, "A Systematic Literature Review of Factors Influencing Acceptance on Mixed Reality Technology", *ARPN Journal of Engineering and Applied Sciences,* Vol. 10, No. 23, pp. 18239-18246, 2015.

[25] H. Sato, S. Watanuki, H. Murakami, R. Sato, H. Ishizaki, and Y. Aida, "Development of a luminescence syncytium induction assay (LuSIA) for easily detecting and quantitatively measuring bovine leukemia virus infection", *Archives of Virology,* Vol. 163, No. 6, pp. 1519-1530, 2018.