Document Similarity Detection Using Indonesian Language Word2vec Model

Nahda Rosa Ramadhanti[1], Siti Mariyah[2]
STIS Polytechnic of Statistics
Jakarta, Indonesia
[1]15.8774@stis.ac.id, [2] sitimariyah@stis.ac.id

***Abstract*—Most researches on text duplication in Bahasa uses the TF-IDF method. In this method, each word will have a different weight. The more frequencies the word appears, the greater the weight. This study aims to detect the similarity of documents by calculating cosine similarity from word vectors. The corpus was built from a collection of Indonesian Wikipedia articles. This study proposes two techniques to calculate the similarity which is simultaneous and partial comparison. Simultaneous comparison is direct comparison without dividing documents into several chapters, while partial comparison divides documents into several chapters before calculating the similarity. Similarity result from partial comparison is more accurate than simultaneous comparison. This study uses Unicheck application TF-IDF method as a benchmark. Similarity result from Unicheck and this study are different, due to the different method applied. Similarity result using TF-IDF method is smaller than using Word2vec, this is because TF-IDF can't detect paraphrase. The limitation in this study is that the Unicheck application used as a benchmark does not use the same method as the method used in this study other than that the determination of expected value is still subjective.**

Keywords—word2vec, cosine similarity, word embedding, similarity detection, semantic similarity

I. INTRODUCTION

Text duplication on an article is something to avoid. The originality of an article is crucial. An article with the same data, methods, and goals with another article can be called similar. Detecting similarity on article can be done with comparing each chapter, for example comparing the introduction, methodology, results, or references from both documents. There are some examples of text duplication. The first one is to duplicate the text as it is, and the other one is using paraphrases and a paraphrase is difficult to detect. The paraphrase is a method for re-expressing a sentence using different words but having the same meaning. One of the method that can be used to detecting the duplication text is Term Frequency-Inverse Document Frequency (TF-IDF). TF-IDF is a method to calculate the frequency of the occurrence of words on a document. Some research already using TF-IDF to calculate the similarity [1-3].

TF-IDF is difficult to use to detect paraphrase because TF-IDF calculates word by word. Other than that TF-IDF cannot calculate the words semantically. Two words with the same meaning but different writing will count twice, for example, house and home. Another method that can be used to calculate the words semantic is word embedding. The word embedding can turn a word into a vector and calculate the similarity. One of the most popular methods of word embedding is Word2vec. Some research already used Word2vec to detect texts similarity [4-6]. Word2vec is a method proposed by Mikolov et al. [7-8]. Word2vec can capture the words semantically. This method can map the word vector into vector space to see the similarity between words. Word2vec can be used in many languages. Word2vechas potential to be used in machine translation between non-closely related language, e.g. English and Vietnamese [9]. For some languages with another alphabet, such as Korean or Japanese, it is necessary to do segmentation to the corpus. Nowadays many usage of Word2vec in English or other languages, but not in Indonesian language (Bahasa Indonesia). Bahasa corpus is still limited, some of it was made by Leipzig University and SEAlang Library, other than that a corpus can be build using articles from Indonesian Wikipedia.

Grammar in English and Bahasa is a little different, e.g. "red table" in English and "meja merah" in Bahasa. In English, "red" describe "table", but in Bahasa "merah" describe "meja". Furthermore, in Bahasa there is homonym. A homonym is two words having the same spelling or pronunciation but different meanings. In this study, we aim to develop Word2vec model using Bahasa corpus to measure the similarity between two documents. We conduct some simulation to compare the method of how to detect similarity. We propose two technics which are simultaneous comparison and partial comparison, we tested which method is accurate. The difference with other studies is we simulate composite similarity on the documents. Other than that we also build a desktop application facilitate users in calculating cosine similarity of two documents.

The rest of this paper will explain how we propose the Word2vec model using Bahasa corpus. In section II we describe some literature that we used as related research to help this study. Section III explains the method that we use in this study. We also describe the simulation and evaluation process. Section IV tells the result of the simulation and evaluation, and also the user interface for the application. The last is the conclusion, explain what we have done in this study.

II. LITERATURE REVIEW

In 2013 Mikolov et al. proposed Word2vec method [7] [8]. Word2vec uses two models, i.e. Continuous Bag of Words and Skip-Gram. On that paper, they set 300 dimensions vector and

window size 5 and give a good performance. They also state that Word2vec can process words on a big corpus in a short time. Text as input and vector as output. The third related work by Gao et al. [4] about detecting duplicate short text using three methods, TF-IDF, Word2vec, and Word2vec weighted by TF-IDF. The fourth related work by Zhang et al. [10] concludes that semantic information from Word2vec is better than LSI (Latent Semantic Indexing) and LDA (Latent Dirichlet Allocation). That corpus used on that work is NBA competition from Chinese sports news portal.

The fifth related work by Widyastuti et al. [11] test some window size on three corpora. From that work, we know that there are several things that affect the semantic result of Word2vec, which are window size and corpus size. The sixth related work by Kenter at al. [5] set 400 dimensions and window size 5 to create word vector. Another work from Suleiman et al. [6] use Arabic corpus and 100 dimensions vector and window size 10. The study by Ryansyah and Andayani [1] apply TF-IDF to calculate document similarity. Study by Sekarwati et al. [3] develop LSA weighted with TF-IDF and calculate the similarity using Cosine similarity, Dice's similarity, and Jaccard similarity. The last three works are using Bahasa.

Based on the previous works, similarity detection in Bahasa are still using TF-IDF. Whereas, TF-IDF cannot detect paraphrase. Word2vec has been widely used in another language such as English, Chinese, Arabic, etc. So we do research about detecting similarity on Bahasa documents using Word2vec and cosine similarity.

III. METHOD

In building the Word2vec model, the corpus becomes important because the richness of words in the corpus will determine whether good or bad of the model. Currently, the Bahasa corpus is still limited, so in this study, we use a collection of Indonesian Wikipedia articles to build the corpus. The total of articles used is 353,238 articles. The articles are processed to be a corpus that contains a combination of all articles into one line.

To make a corpus, we use Gensim library on Python. Gensim provide a function, called WikiCorpus, to process an .xml file format into .text file format that can be used as an input for Word2vec model. Lemmatize we set into False to avoid the stemming process. After that we form the Word2vec model. Some things to consider in forming the Word2vec model are the determination of vector dimensions and window size. In this study, we set 300 for vector dimensions and 5 for window size.

Word2vec is a hidden layer. Word2vec consist of two models, i.e. Continuous Bag of Words (CBOW) and Skip Gram. Each model consist of 3 layers, which are input layer, projection layer, and output layer [7]. CBOW model can predict an output (target word) based on the context word as an input [7]. For example, CBOW model can predict "vegetable" as an output (target word) with "mother buy … in the market" as an input. Skip Gram model is the opposite of CBOW model, in Skip Gram model the input word predict some output, for example "vegetable" as an input can predict "mother", "buy" and "market" as an output. In this paper, we use CBOW model because the training process is faster than Skip Gram model. Training process using CBOW model takes 1225.3s and Skip Gram model takes 4773.2s. Besides that, Gensim applies CBOW as the standard model.

The stages of calculating the similarity of documents is seen in Fig 1. The first stage is document input, and then preprocessing, vectorization, and the last is the calculation of cosine similarity between two vectors.

The input document are documents that contains 4 chapters and references. Preprocessing includes two stages, case folding and stopword filtering, in this stage all of the punctuations also removed. We skip the stemming process because it will reduce the diversity of words. Word embedding turns words from the preprocessing stage into vector, word vector document A and word vector document B. Finally, calculating the cosine similarity of two vector determines whether the document is similar or not. Cosine similarity range between 0 and 1, 0 if the angle perpendicular or 90^o, and 1 if the angle is 0^o. For example there are two abstracts that are suspected to be similar.

The first process is preprocessing, in preprocessing we do case folding and stopword filtering. Case folding is lowercasing all of the letters. Stopword filtering is deleting unimportant words, such as conjunction.

After preprocessing, the next process is vectorization. On this process, not all of words has word vector. Only words that exist on corpus will turn into a word vectors.

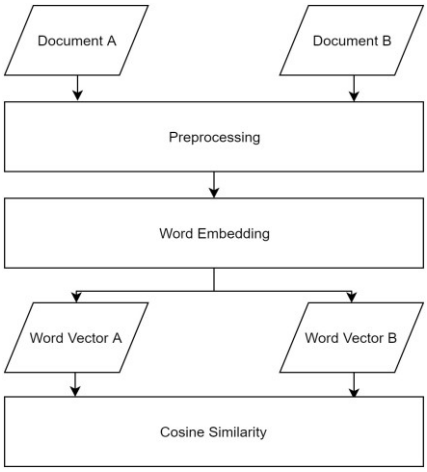

Fig. 1. Research model

TABLE I. TWO ABSTRACTS THAT ARE SUSPECTED TO BE SIMILAR

Abstract 1

Duplikasi teks sulit dideteksi hanya dengan menggunakan metode TF-IDF atau perbandingan antar kata. Pada metode tersebut, dokumen dikatakan mirip jika menggunakan struktur dan pilihan kata yang sama. Penelitian ini bertujuan untuk medeteksi kemiripan teks dengan menghitung cosine similarity dari vektor kata yang didapatkan melalui model Word2Vector Bahasa Indonesia. Korpus dibuat dari kumpulan judul artikel dari Wikipedia Bahasa Indonesia. Tahapan penelitian yang dilakukan yaitu pembangunan model Word2Vector, preprocessing, pembentukan vektor untuk dokumen, dan penghitungan nilai kemiripan menggunakan cosine similarity.
Abstract 2
Duplikasi pada dokumen akan sulit dideteksi jika hanya menggunakan metode TF-IDF atau perbandingan kata saja, hanya dokumen yang menggunakan struktur kalimat yang sama yang akan dideteksi kemiripannya. Pada penelitian ini bertujuan untuk melihat kemiripan dokumen dengan cara menghitung cosine similarity dari vektor kata yang didapatkan dari model Word2Vector Bahasa Indonesia. Korpus dibangun dari kumpulan judul artikel Wikipedia Bahasa Indonesia. Tahapan yang dilakukan pada penelitian ini antara lain, membangun model Word2Vector, preprocessing, pembentukan vektor kata untuk dokumen, dan penghitungan kemiripan menggunakan cosine similarity.

TABLE II. TWO ABSTRACTS AFTER PREPROCESSING

Abstract 1
“duplikasi”, “teks”, “sulit”, “dideteksi”, “metode”, “tfidf”, “perbandingan”, “metode”, “dokumen”, “struktur”, “pilihan”, “penelitian”, “bertujuan”, “medeteksi”, “kemiripan”, “teks”, “menghitung”, “cosine”, “similarity”, “vektor”, “didapatkan”, “model”, “word2vector”, “bahasa”, “indonesia”, “korpus”, “kumpulan”, “judul”, “artikel”, “wikipedia”, “bahasa”, “indonesia”, “tahapan”, “penelitian”, “pembangunan”, “model”, “word2vector”, “preprocessing”, “pembentukan”, “vektor”, “dokumen”, “penghitungan”, “nilai”, “kemiripan”, “cosine”, “similarity”
Abstract 2
“duplikasi”, “dokumen”, “sulit”, “dideteksi”, “metode”, “tfidf”, “perbandingan”, “dokumen”, “struktur”, “kalimat”, “dideteksi”, “kemiripannya”, “penelitian”, “bertujuan”, “kemiripan”, “dokumen”, “menghitung”, “cosine”, “similarity”, “vektor”, “didapatkan”, “model”, “word2vector”, “bahasa”, “indonesia”, “korpus”, “dibangun”, “kumpulan”, “judul”, “artikel”, “wikipedia”, “bahasa”, “indonesia”, “tahapan”, “penelitian”, “membangun”, “model”, “word2vector”, “preprocessing”, “pembentukan”, “vektor”, “angka”, “dokumen”, “penghitungan”, “kemiripan”, “cosine”, “similarity”

TABLE III. TWO ABSTRACTS AFTER PREPROCESSING

Abstract 1
“duplikasi”, “teks”, “sulit”, “dideteksi”, “metode”, “perbandingan”, “metode”, “dokumen”, “struktur”, “pilihan”, “penelitian”, “bertujuan”, “medeteksi”, “kemiripan”, “teks”, “menghitung”, “cosine”, “similarity”, “vektor”, “didapatkan”, “model”, “bahasa”, “indonesia”, “korpus”, “kumpulan”, “judul”, “artikel”, “wikipedia”, “bahasa”, “indonesia”, “tahapan”, “penelitian”, “pembangunan”, “model”, “pembentukan”, “vektor”, “dokumen”, “penghitungan”, “nilai”, “kemiripan”, “cosine”, “similarity”
Abstract 2
“duplikasi”, “dokumen”, “sulit”, “dideteksi”, “metode”, “perbandingan”, “dokumen”, “struktur”, “kalimat”, “dideteksi”, “kemiripannya”, “penelitian”, “bertujuan”, “kemiripan”, “dokumen”, “menghitung”, “cosine”, “similarity”, “vektor”, “didapatkan”, “model”, “bahasa”, “indonesia”, “korpus”, “dibangun”, “kumpulan”, “judul”, “artikel”, “wikipedia”, “bahasa”, “indonesia”, “tahapan”, “penelitian”, “membangun”, “model”, “pembentukan”, “vektor”, “angka”, “dokumen”, “penghitungan”, “kemiripan”, “cosine”, “similarity”

If Table II is compared to Table III then there are some words which do not appear after the vectorization process, such as “tfidf”, “word2vector”, and “preprocessing”. Those words do not exist on Indonesian Wikipedia articles. After that we calculate the similarity using cosine similarity, the result is 0.984. It means the two abstract has a high similarity, we can conclude that two abstract is similar.

The documents used in this paper are 20 articles written in Bahasa, can be seen in Table IV. There are two techniques to calculate the similarity which are simultaneous comparison and partial comparison. Simultaneous comparison calculates word vectors directly then calculates cosine similarity between two documents. While in partial comparison, every documents will be divided based on the chapters on the document, chapter 1 compared with chapter 1, chapter 2 compared with chapter 2, etc. Every chapter has a different weight.

1. Formula 1 = (0.35 * chapter1) + (0.15 * chapter2) + (0.20 * chapter3) + (0.25 * chapter4) + (0.05 * references)
2. Formula 2 = (0.20 * chapter1) + (0.20 * chapter2) + (0.20 * chapter3) + (0.20 * chapter4) + (0.20 * references)

Environment set:

1. Doc1 vs. Doc2. The topic is full different. Doc1 about document classification and Doc2 about economy
2. Doc3 vs. Doc4. Has the same topic, about sentiment analysis
3. Doc5 vs. Doc6. Has the same topic, about sentiment analysis
4. Doc7 vs. Doc8. Has the same topic, about sentiment analysis
5. Doc9 vs. Doc10. The topic is full different. Doc9 about agribusiness and Doc10 about law
6. Doc11 vs. Doc12. Doc11 about phrase detection and Doc12 about implementation of vector space
7. Doc13 vs. Doc14. Doc13 about comparative analysis and Doc14 about intrusion detection system
8. Doc15 vs. Doc16. Doc15 about sentiment analysis and Doc16 about algorithm classification
9. Doc17 vs. Doc18. Has the same topic, about sentiment analysis
10. Doc19 vs. Doc20. Has the same topic, about sentiment analysis

For the documents with different topic (1 and 5), the expected values of the cosine similarity are between 0.55-0.7. The expected values of documents with same topic (6, 7, and 8) are between 0.7-0.85. For the document with the same topic about sentiment analysis (2, 3, 4, 9, and 10), the cosine similarity will be estimated between 0.85-0.95.

Simulation for the evaluation,

1. Doc1 vs. Test1. Test1 is the same as Doc1 with a reduction in some paragraphs. We set the expected value for this simulation about 0.95-0.99
2. Doc1 vs. Test2. Chapter 1 and 2 on Test2 were replaced with chapter 1 and 2 from Doc9. Meanwhile, chapter 3 and 4 is the same as chapter 3 and 4 from Doc1. Doc1 and Doc9 have a different topic. We set the expected value for this simulation between 0.7-0.85

TABLE IV. LIST OF DOCUMENT USED

Document	Title
Doc1	*Penerapan Algoritma* Cosine Similarity *dan Pembobotan* TF-IDF *pada Sistem Klasifikasi Dokumen Skripsi* Application of Cosine Similarity Algorithm and TF-IDF Weighting in Thesis Document Classification System
Doc2	*Peran* Gender *Perempuan terhadap Pertumbuhan Ekonomi di Provinsi Jawa Tengah tahun 2008-2012* The Role of Women's Gender on Economic Growth in Central Java Province in 2008-2012
Doc3	*Analisis Sentimen terhadap Tayangan Televisi berdasarkan Opini Masyarakat pada Media Sosial* Twitter *menggunakan Metode* K-Nearest Neighbor *dan Pembobotan Jumlah* Retweet Sentiment Analysis of Television Program based on Public Opinion on Twitter using K-Nearest Neighbor Method and Weighting the Number of Retweets
Doc4	*Analisis Sentimen Impor Beras 2018 pada* Twitter *menggunakan Metode* Support Vector Machine *dan Pembobotan Jumlah* Retweet Sentiment Analysis of 2018 Rice Import on Twitter using Support Vector Machine and Weighting the Number of Retweets
Doc5	*Analisis Sentimen pada* Review *Konsumen menggunakan Metode* Naive Bayes *dengan Seleksi Fitur* Chi Square *untuk Rekomendasi Lokasi Makanan Tradisional* Sentiment Analysis on Consumer Reviews using Naïve Bayes Method with Chi Square Feature Selection for Recommended Traditional Food Locations
Doc6	*Analisis Sentimen tentang Opini Pilkada Dki 2017 pada Dokumen* Twitter *Berbahasa* Indonesia *menggunakan* Näive Bayes *dan Pembobotan* Emoji Sentiment Analysis of the 2017 DKI Election Opinion on Indonesian Language Twitter Documents using Naïve Bayes and Emoji Weighting
Doc7	*Analisis Sentimen Kurikulum 2013 pada Sosial Media* Twitter *menggunakan Metode* K-Nearest Neighbor *dan* Feature Selection Query Expansion Ranking Sentiment Analysis of 2013 Curriculum on Twitter using K-Nearest Neighbor Method and Feature Selection Query Expansion Ranking
Doc8	*Analisis Sentimen Tingkat Kepuasan Pengguna Penyedia Layanan Telekomunikasi Seluler Indonesia pada* Twitter *dengan Metode* Support Vector Machine *dan* Lexicon Based Features Sentiment Analysis of User Satisfaction Level of Indonesian Cellular Telecommunications Providers on Twitter with Support Vector Machine and Lexicon Based Features Method
Doc9	*Analisis Faktor Internal dan Eksternal yang Mempengaruhi Pengembangan Agribisnis Tembakau di Kabupaten Jember* Analysis of Internal and External Factors Affecting Tobacco Agribusiness Development in Jember Regency
Doc10	*Tinjauan Yuridis Kewenangan Komisi Pemberantasan Korupsi Melakukan Penyidikan Penggabungan Perkara Tindak Pidana Korupsi dan Pencucian Uang* Juridical Review the Authority of the Corruption Eradication Commision Investigates the Merger of Corruption and Money Laundering Cases
Doc11	*Pengaruh* Phrase Detection *dengan* POS-Tagger *terhadap Akurasi Klasifikasi Sentimen menggunakan* SVM The Effect of Phrase Detection with POS-Tagger on the Accuracy of Sentiment Cllasification using SVM
Doc12	*Implementasi* Vector Space Model *dalam Pembangkitan* Frequently Asked Questions *Otomatis dan Solusi yang Relevan untuk Keluhan Pelanggan* Implementation of Vector Space Model in Generating Automatic Frequently Asked Questions and Relevant Solutions for Customer Complaints
Doc13	*Analisis Perbandingan* Detection Traffic Anomaly *dengan Metode* Naive Bayes *dan* Support Vector Machine (SVM) Comparative Analysis of Traffic Anomaly Detection with Naïve Bayes Method and Support Vector Machine (SVM)
Doc14	*Penerapan* Naive Bayes *pada* Intrusion Detection System *dengan Diskritisasi Variabel* Application of Naïve Bayes in Intrusion Detection System with Variable Discretization
Doc15	*Analisis Sentimen untuk Komentar pada Sistem Pencarian Kost Menggunakan Metode* Support Vector Machine (SVM) Sentiment Analysis for Comments on Boarding Search Systems using Support Vector Machine (SVM) Method
Doc16	*Klasifikasi Algoritma* TF *dan* Neural Network *dalam Sentimen Analisis* Classification of TF and Neural Network Algorithms in Sentiment Analysis
Doc17	*Klasifikasi Dokumen* Twitter *untuk Mengetahui Karakter Calon Karyawan Menggunakan Algoritme* K-Nearest Neighbor (KNN) Twitter Document Classification to Know the Character of Prospective Employees using K-Nearest Neighbor (KNN) Algorithm
Doc18	*Implementasi Metode* K-Nearest Neighbor *dengan* Decision Rule *untuk Klasifikasi Subtopik Berita* Implementation of K-Nearest Neighbor Method with Decision Rule for Classification of News Subtopics
Doc19	*Sentimen Analisis Berinternet pada Media Sosial dengan Menggunakan Algoritma* Bayes Sentiment Analysis of Internet on Social Media using Bayes Algorithm
Doc20	*Analisis Sentimen* Review *Restoran menggunakan Algoritma* Naive Bayes *berbasis* Particle Swarm Optimization Sentiment Analysis of Restaurant Review using Naïve Bayes Algorithm based on Particle Swarm Optimization

3. Doc1 vs. Test3. Chapter 3 and 4 on Test3 were replaced with chapter 3 and 4 from Doc9. Meanwhile, chapter 1 and 2 is the same as chapter 1 and 2 from Doc1. We set the expected value for this simulation between 0.7-0.85
4. Doc1 vs. Test4. Chapter 1 and 2 on Test4 were replaced with chapter 1 and 2 from Doc17, meanwhile, chapter 3 and 4 are the same as chapter 3 and 4 from Doc1. Doc1 and Doc17 have the same topic. We set the expected value for this simulation about 0.85-0.95
5. Doc1 vs. Test5. Chapter 3 and 4 on Test5 were replaced with chapter 3 and 4 from Doc17. Meanwhile, chapter 1 and 2 is the same as chapter 1 and 2 from Doc1. We set the expected value for this simulation about 0.85-0.95

IV. Result and Analysis

A. Experiment Simulation

This simulation is aimed to see the cosine similarity from the two techniques of comparison. The simulation process and experimental parameters are follows: First, a corpus is made by articles from Indonesian Wikipedia. Then Word2vec is used to find the word vector for each words, the vector dimension are 300 dimensions with window size 5. Therefore, each word in the corpus will turn into word vector with 300 dimensions. Vector dimension and window size used in the literature review are different for each research. In this study, we used 300 for vector dimensions and 5 for window size based on the research by Mikolov et.al [7]. Finally the similarity is calculate cosine similarity formula.

Based on Table V, the cosine similarity is higher with simultaneous comparison rather than partial comparison. This is because on simultaneous comparison all of the elements are calculated, such as title and abstract, while on partial comparison, the contents are calculated only. The simultaneous result of the second, third, fourth, fifth, and sixth simulations are not in accordance with the expected value, the simultaneous result gives a higher result than the expected value. For the partial comparison, on seventh simulation the result higher than the expected value, and for ninth simulation the result lower than the expected value.

The result of the fifth simulation is the smallest among all simulations because the two documents have a different topic. Doc 9 about agribusiness and Doc 10 about the law. Some of the simulation results are not accordance with the expected value because all of the word vector from the text is directly averaged, that average is too big to represent all of the words from a text, so the character of a document tends to decrease.

TABLE V. SIMULATION RESULT

Simulation	Similarity		
	Simultaneous	Partial	
		Formula 1	Formula 2
Doc1 vs. Doc2	0.692	0.638	0.657
Doc3 vs. Doc4	0.983	0.900	0.916
Doc5 vs. Doc6	0.955	0.881	0.898
Doc7 vs. Doc8	0.956	0.901	0.914
Doc9 vs. Doc10	0.580	0.571	0.601
Doc11 vs. Doc12	0.942	0.805	0.843
Doc13 vs. Doc14	0.960	0.887	0.884
Doc15 vs. Doc16	0.788	0.732	0.711
Doc17 vs. Doc18	0.898	0.802	0.806
Doc19 vs. Doc20	0.915	0.858	0.875

TABLE VI. EVALUTION WITH DOC1

Document	Doc 1		
	Simultaneous	Partial	
		Formula 1	Formula 2
Test 1	0.996	0.973	0.972
Test 2	0.978	0.808	0.847
Test 3	0.951	0.858	0.874
Test 4	0.985	0.923	0.938
Test 5	0.981	0.937	0.944

TABLE VII. SIMILARITY RESULT FOR EACH CHAPTER

Document	Similarity				
	Doc 1				
	Chap. 1	Chap. 2	Chap. 3	Chap. 4	References
Test 1	0.998	0.931	0.991	0.943	0.999
Test 2	0.613	0.624	1	0.999	0.999
Test 3	0.999	0.990	0.714	0.666	0.999
Test 4	0.843	0.851	1	0.999	0.999
Test 5	0.999	0.990	0.883	0.848	0.999

TABLE VIII. UNICHECK RESULT

Document	Similarity
	Doc 1
Test 1	0.909
Test 2	0.713
Test 3	0.647
Test 4	0.562
Test 5	0.674

B. Evaluation

There are two evaluations, the first one is using a simulation of Doc 1 and some test documents, and the second one is benchmarking with Unicheck application and TF-IDF method. The test documents contents are based on Doc 1 with some replacement on several chapters.

From the Table VI, the result of simultaneous comparison is higher than partial comparison. The first simulation is higher because not many differences of Doc 1 and Test 1. All of the simultaneous comparisons are not accordance with expected value.

From Table VII, the similarity for reference does not reach 1.0. This is because there is a difference when parsing the documents. In Doc 1 and Test 1 there is the word “Al-Qur’an” which is not detected in Doc 1 but detected in Test 1. We assume that there is influence from the construction of test documents which are done manually by changing the pdf format into doc format and saved back into pdf format.

For the second simulation, the contents from chapter 3 and chapter 4 of test document are as same as the contents from chapter 3 and chapter 4 of Doc 1, but there is a small reduction in some parts, such as pictures and the description. The result for chapter 1 and chapter 2 is small because the topics are different, the topic of chapter 1 and chapter 2 of the test document is about agribusiness and for the Doc 1 is about document classification.

We also perform an evaluation using another application, Unicheck, even though the method not apple to apple, we chose this application as a benchmark because it has been used as a plagiarism checker by the STIS Polytechnic of Statistics. Table VI is the result using Word2vec and cosine similarity and table VIII is the result using Unicheck application. The result is different because the method to calculate the similarity between this paper and Unicheck is different. One use Word2vec and Unicheck do not apply Word2vec.

Beside compared the result with Unicheck application, we also compared it with TF-IDF method. Table IX is the cosine similarity result using TF-IDF method. If we compared Table IX and Table V, the similarity result using TF-IDF is smaller than using Word2vec, this is because TF-IDF can’t detect paraphrase. Based on the result, we conclude that partial comparison is more accurate than simultaneous comparison, the result of partial comparison is more in accordance with expected value than the result of the simultaneous comparison.

TABLE IX. TF-IDF RESULT

Simulation	Similarity
Doc1 vs. Doc2	0.363
Doc3 vs. Doc4	0.751
Doc5 vs. Doc6	0.534
Doc7 vs. Doc8	0.557
Doc9 vs. Doc10	0.236
Doc11 vs. Doc12	0.445
Doc13 vs. Doc14	0.589
Doc15 vs. Doc16	0.438
Doc17 vs. Doc18	0.507
Doc19 vs. Doc20	0.695

C. *Application Display*

As stated in the introduction, we not only build the Word2vec model, but also we develop an application completed with a graphical user interface (GUI) in order to facilitate the users. The application was built using Python GUI with Pyside framework. This application is used to calculating the cosine similarity of two documents. The features are document input, chapters input, and result. The application shows the similarity between two documents generally and the similarity between each chapter on those documents which applying both simultaneous comparison and partial comparison.

V. CONCLUSION

We develop Word2vec model in Bahasa to detect similarity between two documents. We utilize Wikipedia in Bahasa as our corpus to construct word vector. We also perform the simulations to compare two techniques in calculating the cosine similarity of two documents. The two techniques are simultaneous comparison and partial comparison. Our simulation result shows that partial comparison is more accurate on measuring the similarity between two documents. This is because in partial comparison, the technique compare and calculate the contents only, such as introduction, method, result, conclusion, and references. We conduct benchmarking with Unicheck application even though the application method not apple to apple with our method. We also build an application to facilitate users in using our proposed model.

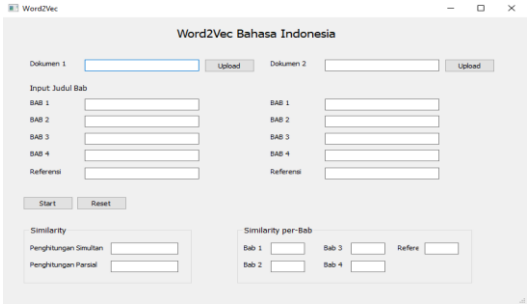

Fig. 2. Application display

Fig. 3. Application display

REFERENCES

[1] S. Andayani and A. Ryansyah, "Implementasi Algoritma TF-IDF Pada Pengukuran Kesamaan Dokumen," *JuSiTik: Jurnal Sistem dan Teknologi Informasi Komunikasi,* vol. 1, no. 1, pp. 53-62, 2017.

[2] N. Khairunnisa, D. Sihabudin, and A. Wibowo, "Aplikasi Pendeteksi Plagiat dengan Menggunakan Metode Latent Semantic Analysis (Studi Kasus: Laporan TA PCR," *Jurnal Aksara Komputer Terapan*, vol. 1, no. 2, 2012.

[3] K.A. Sekarwati, L.Y. Banowosari, I.M. Wiryana, and D. Kerami, "Pengukuran Kemiripan Dokumen dengan Menggunakan Tools Gensim," *Prosiding SNST Fakultas Teknik*, vol. 1, no. 1, 2015.

[4] J. Gao, Y. He, X. Zhang, and Y. Xia, "Duplicate short text detection based on Word2vec," 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), pp. 33-37, 2017.

[5] T. Kenter and M. De Rijke, "Short text similarity with word embeddings," In Proc. 24th ACM International on Conference on Information and Knowledge Management, 2015, pp. 1411-1420.

[6] D. Suleiman, A. Awajan, and N. Al-Madi, "Deep Learning Based Technique for Plagiarism Detection in Arabic Texts," presented at International Conference on New Trends in Computing Sciences (ICTCS), 2017, pp. 216-222.

[7] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," International Conference on Learning Representations, 2013.

[8] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Word and Phrases and Their Compositionality," presented at 26th International Conference on Neural Information Processing System, vol. 2, 2013, pp. 3111-3119.

[9] T. Mikolov, Q.V. Le, and I. Sutskever, "Exploiting similarities among languages for machine translation," arXiv:1309.4168, 2013.

[10] C. Zhang, X. Wang, S. Yu, and Y. Wang, "Research on Keyword Extraction of Word2vec Model in Chinese Corpus," presented at 2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS), 2018, pp. 339-343.

[11] N.N. Widyastuti, A.A. Bijaksan, and I.L. Sardi, "Analisis Word2vec untuk Perhitungan Kesamaan Semantik antar Kata," *eProceedings of Engineering*, vol. 5, no. 3, 2018.

[12] Wikipedia, "Index of /idwiki/latest," Jan., 2019. [Online]. Available: https://dumps.wikimedia.org/idwiki/latest/. [Accessed Aug. 14, 2019].