An Extensive Comparative Analysis of Document Similarity Algorithms: A Systematic Survey and Evaluation

Mr. Gaurabh G. Sawal, M.E. (ITE, GEC) and Dr. Nilesh B. Fal Dessai,
Head of Department (ITE, GEC)
Goa Engineering College (GEC), Farmagudi, 403401, Goa, India.

Contributing authors: gaurabh.gec@gmail.com

Abstract:

Document similarity analysis plays a crucial role in various natural language processing (NLP) tasks such as information retrieval, text classification, and recommendation systems. This research paper presents a comprehensive survey and comparison of five prominent techniques for document similarity: BERT, GloVe, USE, Doc2Vec, and LDA. The paper begins by providing an overview of each technique and their underlying principles. BERT, a transformer-based model, captures contextual word representations, while GloVe generates dense word embeddings based on global co-occurrence statistics. USE offers sentence-level embeddings that capture semantic information, and Doc2Vec extends the concept to document-level embeddings. LDA, a probabilistic topic modelling technique, assigns topics to documents based on word distributions. The survey focuses on evaluating these techniques in terms of their performance, applicability, and computational efficiency for document similarity tasks. The results of the comparative analysis highlight the strengths and limitations of each technique. BERT excels in capturing fine-grained contextual information, while GloVe is effective in capturing global semantic relationships. USE provides reliable sentence-level embeddings, and Doc2Vec captures document-level semantics. LDA offers a probabilistic approach to modelling document topics. By providing a comprehensive survey and comparison of these document similarity techniques, this research paper aims to assist researchers and practitioners in selecting the most suitable technique for their specific NLP tasks.

Keywords: BERT, GloVe, USE, Doc2Vec, LDA

1.Introduction:

Natural Language Processing (NLP) plays a crucial role in various applications such as information retrieval, sentiment analysis, and machine translation. One of the fundamental tasks in NLP is document similarity analysis, which involves measuring the similarity or relatedness between two or more documents. Accurate document similarity analysis is essential for tasks like document clustering, plagiarism detection, and document retrieval. In recent years, there have been significant advancements in document similarity algorithms, fuelled by the development of pre-trained language models such as BERT (Bidirectional Encoder Representations from Transformers), GLOVE, USE, Doc2Vec, and LDA. These models have revolutionized the field of NLP by providing powerful contextual representations of words and documents.

The abstract for this research paper presents an overview of a comprehensive survey and comparison of various document similarity techniques, focusing on BERT, GLOVE, USE, Doc2Vec, and LDA. The primary objective of this study is to analyse the strengths and limitations of these algorithms in capturing semantic and contextual similarities between documents. By understanding their performance characteristics, we aim to provide insights into their suitability for different NLP tasks and shed light on the state-of-the-art in document similarity analysis.

This research paper aims to address the following research questions:

1. How do BERT, GLOVE, USE, Doc2Vec, and LDA differ in their approaches to document similarity analysis?

2. What are the strengths and weaknesses of each technique in capturing semantic relationships between documents?

3. How do these algorithms execute feature extraction?

4. What is the underlying principle with respect to the execution of the algorithms?

By conducting an extensive survey and comparison of these algorithms, we intend to provide valuable insights to researchers and practitioners in the field of NLP. Understanding the capabilities and limitations of these document similarity techniques is crucial for leveraging their potential in improving the performance and efficiency of NLP applications. The remainder of this paper is organized as follows: Section 2 provides a detailed review of the related literature and the underlying principles of BERT, GLOVE, USE, Doc2Vec, and LDA. Section 3 presents the feature extraction capabilities of each algorithm in detail. Section 4 discusses the strengths and weaknesses of each technique in capturing semantic relationships between documents. Finally, Section 5 concludes the paper by summarizing the findings, discussing their implications, and highlighting directions for future research.

Through this research, we aim to contribute to the advancement of document similarity analysis and provide valuable insights for researchers, practitioners, and developers in the field of NLP. By understanding the strengths and weaknesses of these techniques, we can unlock their potential for a wide range of NLP applications and drive further innovation in the field.

2. Related Literature and Underlying Principles of BERT, GLOVE, USE, Doc2Vec, and LDA

2.1 Literature Review

The field of document similarity analysis has witnessed extensive research and development in recent years. Numerous studies have explored different approaches and techniques to measure the similarity between documents. In this section, we provide a comprehensive review of the existing literature, highlighting key research contributions and advancements in the field. Researchers have explored traditional techniques such as TF-IDF (Term Frequency-Inverse Document Frequency) and cosine similarity, which rely on statistical measures to compare documents based on their term frequencies. These methods have been widely used and serve as the baseline for evaluating more advanced approaches.

However, with the emergence of deep learning and pre-trained language models, novel techniques have been introduced that leverage contextual and semantic information. BERT (Bidirectional Encoder Representations from Transformers) is one such model that has gained significant attention in recent years. It utilizes a transformer architecture and pre-training on large-scale corpora to generate powerful word and sentence embeddings. Several studies have demonstrated the effectiveness of BERT in capturing fine-grained semantic relationships between documents.

GLOVE (Global Vectors for Word Representation) is another popular approach that focuses on creating word embeddings based on co-occurrence statistics. It constructs a global word-word co-occurrence matrix and performs dimensionality reduction to obtain dense word representations. GLOVE embeddings have been widely used in document similarity analysis tasks, showcasing their ability to capture semantic similarities.

USE (Universal Sentence Encoder) is a pre-trained model specifically designed for sentence-level embeddings. It encodes sentences into fixed-length vectors, capturing both syntactic and semantic information. USE has been successfully applied in various NLP tasks, including document similarity analysis, due to its ability to capture contextual nuances.

Doc2Vec is a technique that extends word embeddings to entire documents. It represents documents as fixed-length vectors, allowing comparison and similarity measurement at the document level. By training a neural network model on a large collection of documents, Doc2Vec captures the underlying semantic structure of the corpus and enables effective document similarity analysis.

Latent Dirichlet Allocation (LDA) is a probabilistic topic modelling technique that represents documents as mixtures of latent topics. Each topic is a distribution over words, and LDA assigns a probability to each topic for a given document. By inferring the underlying topic distributions, LDA facilitates document comparison based on shared topics and provides insights into thematic similarities.

2.2 Underlying Principles of BERT, GLOVE, USE, Doc2Vec, and LDA

In this subsection, we delve into the underlying principles and mechanisms of BERT, GLOVE, USE, Doc2Vec, and LDA.

BERT: BERT is a transformer-based model that utilizes a multi-layer bidirectional architecture. It learns contextualized word embeddings by training on large-scale corpora in a masked language modelling task. BERT captures the dependencies and relationships between words in both left and right contexts, enabling it to generate highly expressive word representations.

GLOVE: GLOVE constructs word embeddings based on global word co-occurrence statistics. It builds a co-occurrence matrix by counting the number of times words appear together in a given context window. By factorizing this matrix and performing dimensionality reduction, GLOVE obtains dense vector representations that encode semantic similarities between words.

USE: The Universal Sentence Encoder leverages a transformer architecture combined with unsupervised learning. It encodes sentences into fixed-length vectors by processing the words in the sentence and capturing syntactic and semantic relationships. USE employs a bi-directional transformer to model the contextual information within sentences, enabling it to generate high-quality sentence embeddings.

Doc2Vec: Doc2Vec extends word embeddings to the document level by associating each document with a fixed-length vector. It employs the same principles as Word2Vec, where a neural network is trained to predict words in a context window. By including document identifiers as additional input, Doc2Vec generates document embeddings that capture the semantic representation of the entire document.

LDA: Latent Dirichlet Allocation is a generative probabilistic model for topic modelling. It assumes that each document is a mixture of topics, and each topic is a distribution over words. LDA utilizes a statistical inference algorithm to estimate the topic proportions for each document and the word distributions for each topic. By representing documents as mixtures of topics, LDA enables document similarity analysis based on shared thematic content.

3. Feature Extraction

Each document similarity algorithm utilizes a specific approach for feature extraction. In this section, we provide detailed information on how features are extracted using BERT, GLOVE, USE, Doc2Vec, and LDA.

BERT: BERT makes use of a pre-trained model to generate contextualized word embeddings. We explain the process of tokenization, encoding, and obtaining the document-level representations using BERT's architecture. Feature extraction in BERT is carried out through a process known as pretraining. BERT is pretrained on a large corpus of text data, such as Wikipedia, where it learns to predict missing words in a sentence using the surrounding context. This pretraining process allows BERT to capture deep contextual representations of words. During pretraining, BERT employs a transformer-based architecture that consists of multiple layers of self-attention and feed-forward neural networks. Each layer in the architecture helps BERT understand and capture different levels of contextual information. Specifically, BERT utilizes a technique called masked language modelling (MLM) during pretraining. In MLM, a certain percentage of words in the input text are randomly masked, and BERT is trained to predict the original masked words based on the context provided by the surrounding words. Through this process, BERT learns to encode rich semantic and contextual information into the representations of words. The hidden states of BERT's transformer layers capture the contextualized word embeddings, which are then used for downstream tasks such as document classification, named entity recognition, or document similarity analysis. When using BERT for feature extraction, the input text is tokenized into sub word units, and each token is passed through BERT's transformer layers. The output hidden states corresponding to each token are then extracted and used as features for downstream tasks. These features capture the contextual information and semantic relationships between the tokens in the input text. Overall, BERT's feature extraction relies on its pretrained knowledge of language and its ability to capture contextual information through self-attention mechanisms. This makes BERT a powerful tool for extracting meaningful features from text data for variety of natural language processing tasks.

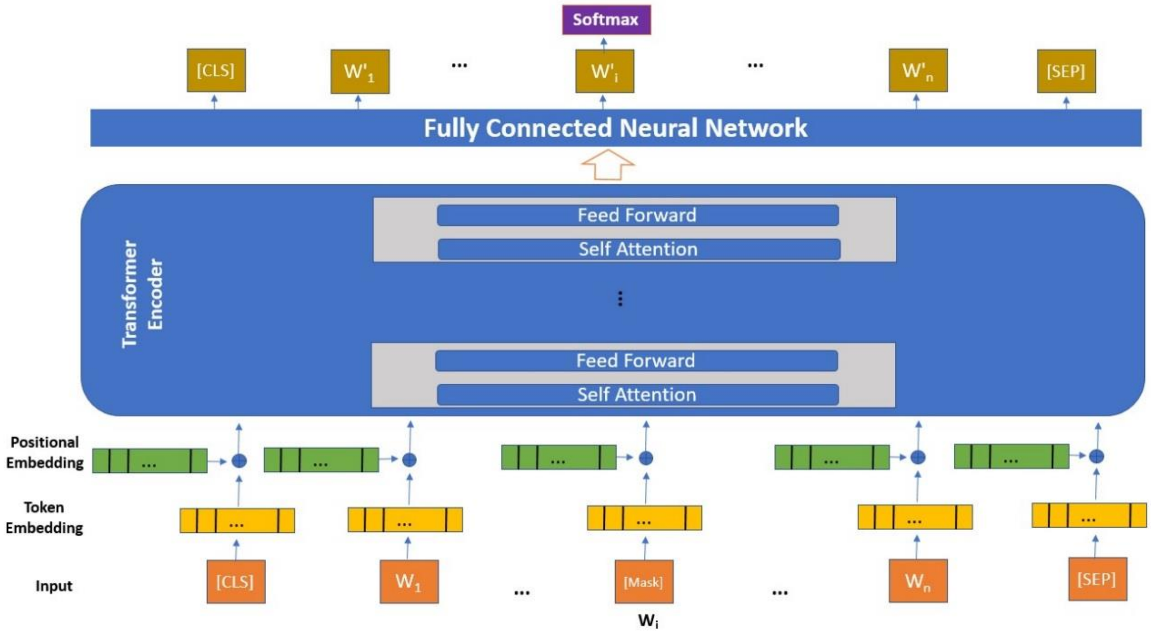

Fig. 1: BERT Architecture and Feature Extraction. [4]

GLOVE: Feature extraction in GLOVE (Global Vectors for Word Representation) is based on word co-occurrence statistics. GLOVE represents words as dense vector embeddings that capture semantic relationships between words based on their co-occurrence patterns in a corpus. The process of feature extraction in GLOVE involves constructing a co-occurrence matrix that represents the frequency of word co-occurrences within a specified context window. This matrix provides information about the semantic associations between words in the corpus. Once the co-occurrence matrix is constructed, GLOVE applies matrix factorization techniques to learn low-dimensional word representations. It aims to factorize the co-occurrence matrix into two separate matrices: one for word vectors and another for context vectors. These matrices are then combined to generate the final word embeddings.

The key idea behind GLOVE is that the dot product of a word vector and its corresponding context vector should capture the probability of word co-occurrence. GLOVE optimizes the word and context vectors by minimizing the difference between the dot product of the vectors and the logarithm of the co-occurrence count. After the training process, GLOVE generates word embeddings that encode semantic relationships between words. These embeddings can be used as features for downstream natural language processing tasks such as document classification, sentiment analysis, or document similarity analysis. One of the advantages of GLOVE is that it can capture both global and local semantic relationships. It considers the overall statistics of word co-occurrence in the entire corpus while also incorporating the local context information within the specified window.

In summary, GLOVE extracts feature by representing words as dense vector embeddings based on their co-occurrence statistics. It leverages matrix factorization techniques to learn low-dimensional representations that capture semantic relationships between words. These embeddings serve as effective features for various NLP tasks.

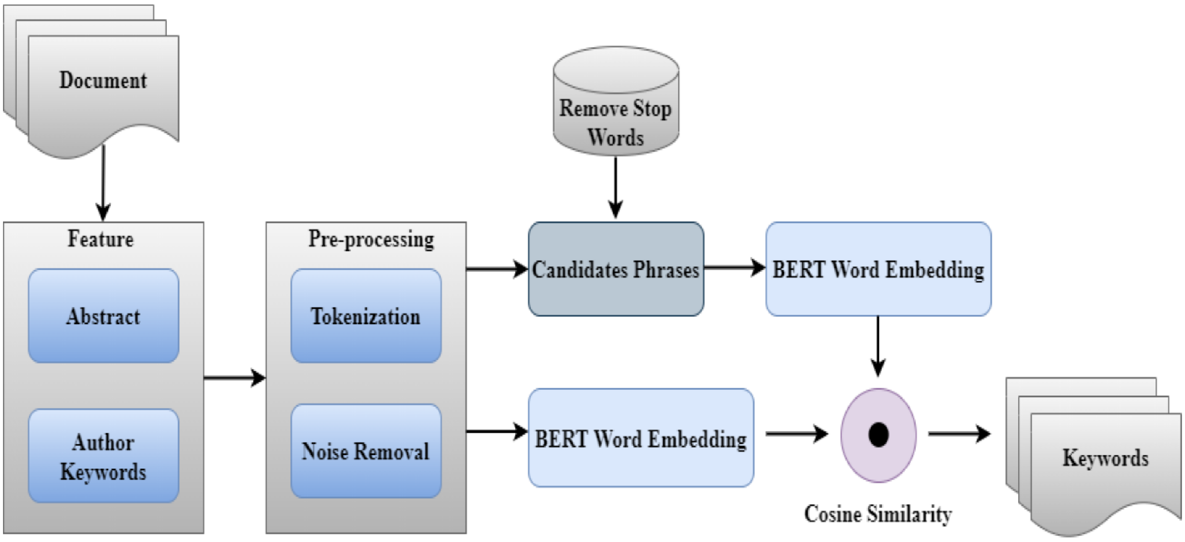

Fig. 2: GLOVE Feature Extraction [7]

USE: Universal Sentence Encoder is a pre-trained model that generates high-quality sentence embeddings. We detail the process of encoding sentences using USE and how these embeddings capture semantic information for document similarity tasks. Feature extraction in USE (Universal Sentence Encoder) involves encoding entire sentences or texts into fixed-dimensional vector representations that capture their semantic meanings. USE is a pre-trained model developed by Google that provides universal representations for various natural language understanding tasks. The feature extraction process in USE utilizes a deep neural network architecture, typically based on transformer models, to encode input sentences. The architecture consists of multiple layers that process the input text and gradually learn representations at different levels of abstraction. To extract features using USE, the input sentences are tokenized into individual words or subword units. These tokens are then converted into dense vector representations through the neural network layers. The network leverages self-attention mechanisms to capture the relationships between words within a sentence and encode the semantic information effectively. The encoding process in USE is designed to capture both the syntax and semantics of the input sentences. It considers the contextual information surrounding each word and incorporates it into the final vector representation. This enables USE to understand the meaning and nuances of the sentences, even in the presence of variations and ambiguities.

One of the notable aspects of USE is its ability to handle sentences of different lengths. It can encode both short and long sentences into fixed-dimensional vectors, allowing for easy comparison and analysis. Since USE is pre-trained on a large corpus of diverse texts, it learns to encode a wide range of semantic properties and linguistic patterns. This makes it a versatile tool for various natural language processing tasks, including document classification, sentiment analysis, semantic similarity measurement, and more. In summary, feature extraction in USE involves encoding sentences or texts into fixed-dimensional vector representations using a deep neural network architecture. The model captures the semantic meaning of the input sentences by considering the contextual information and relationships between words. The resulting embeddings serve as powerful features for a variety of natural language understanding tasks.

Doc2Vec: Feature extraction in Doc2Vec involves encoding entire documents into fixed-dimensional vector representations that capture their semantic meanings. Doc2Vec is an extension of the popular Word2Vec model that is specifically designed to handle larger units of text, such as paragraphs or entire documents. The feature extraction process in Doc2Vec utilizes a neural network architecture, typically based on the skip-gram or continuous bag-of-words (CBOW) models used in Word2Vec. Doc2Vec extends these models by incorporating document-level context along with word-level context. In Doc2Vec, each document is represented by a unique vector, commonly referred to as a "document embedding" or "document vector." The goal is to learn these document vectors in a way that captures the semantic meaning of the documents.

The training process of Doc2Vec involves two main steps: the Paragraph Vector (PV) and Word Vector (WV) phases.

I. **Paragraph Vector (PV) phase**:

In this phase, a unique paragraph ID (or document ID) is assigned to each document in the corpus. A special token, typically called the "paragraph ID" or "document ID," is added to every sentence within a document. The PV phase aims to predict the document vector given the paragraph ID and the context words within the document.

II. Word Vector (WV) phase:

In this phase, the objective is to learn word vectors similar to the Word2Vec model. The document vectors learned in the PV phase are used as additional context in the training process. The Word2Vec model is trained to predict the target word based on the surrounding words and document context. During training, the neural network updates the weights of the word vectors and document vectors based on the prediction errors. This iterative process helps optimize the representations to capture the semantic relationships between words and documents. Once trained, Doc2Vec provides fixed-dimensional vector representations for each document in the corpus. These vectors capture the semantic meaning of the documents in a dense, continuous space. The document vectors can be used for various downstream tasks, such as document similarity measurement, document classification, clustering, and more.

In summary, feature extraction in Doc2Vec involves encoding documents into fixed-dimensional vector representations using a neural network architecture. The model learns to capture the semantic meaning of documents by considering both word-level and document-level context. The resulting document vectors serve as powerful features for various natural language processing tasks. - LDA: We outline the steps involved in training an LDA model for topic modelling. This includes estimating the topic proportions for each document and the word distributions for each topic, which are used to represent documents and enable document similarity analysis.

LDA: Latent Dirichlet Allocation (LDA) engages topic modelling technique for feature extraction from text data. Unlike other methods such as Word2Vec or GloVe that focus on word-level representations, LDA operates at the document-level and aims to discover latent topics within a collection of documents. Here's an explanation of how feature extraction is carried out by LDA:

I. Document Pre-processing: The text documents are pre-processed to remove noise and irrelevant information. This typically involves steps such as tokenization, removing stop words, stemming or lemmatization, and handling other document-specific requirements.

II. Building the LDA Model: The LDA model is built by considering the bag-of-words representation of the documents. Each document is represented as a distribution of topics, and each topic is represented as a distribution of words.

III. Initialization: The LDA model is initialized with random topic assignments for each word in each document.

IV. Iterative Inference: The LDA algorithm iteratively updates the topic assignments for each word based on the current model parameters. Gibbs sampling or variational inference techniques are commonly used for this step.

V. Convergence: The iterations continue until the model parameters stabilize and reach convergence. This is typically determined by monitoring the change in likelihood or other convergence criteria.

VI. Feature Extraction: Once the LDA model has converged, the resulting topic distributions for each document can be used as features. These topic distributions capture the underlying thematic structure of the documents. Each document is represented as a vector of topic probabilities, where each probability indicates the contribution of a topic to that document.

VII. Dimensionality Reduction: The dimensionality of the feature space can be reduced by selecting the most important topics or applying techniques like Singular Value Decomposition (SVD) or Principal Component Analysis (PCA) to obtain a lower-dimensional representation.

VIII. Feature Vector Representation: The final feature vector representation for each document consists of the selected topics and their corresponding probabilities.

The extracted features from LDA can be used for tasks such as document clustering, topic modeling, information retrieval, and content-based recommendation systems. They provide a way to represent documents in a lower-dimensional space that captures the latent semantic structure of the text data.

4. Strength and weakness

The strengths and weaknesses of each technique in capturing semantic relationships between documents have been explained below:

4.1 BERT:

Strengths:

- ❖ BERT has been shown to capture fine-grained semantic relationships between words and documents.
- ❖ It utilizes a transformer-based architecture that allows it to capture contextual information effectively.
- ❖ BERT can handle long-range dependencies and understand the overall context of a document.

Weaknesses:

- BERT requires a significant number of computational resources and time for training and inference.
- Fine-tuning BERT for specific tasks may require substantial labeled data.
- BERT's performance heavily relies on the quality and size of the training data.

4.2. GLOVE:

Strengths:

- GLOVE leverages global word co-occurrence statistics to capture semantic relationships.
- It represents words as dense vectors, allowing for efficient computations and similarity comparisons.
- GLOVE can capture both syntactic and semantic relationships between words and documents.

Weaknesses:

- GLOVE's performance might be limited by the quality and coverage of the training corpus.
- It may struggle with rare or out-of-vocabulary words that have limited co-occurrence statistics.
- GLOVE's word vectors do not capture contextual information or document-level semantics.

4.3 USE (Universal Sentence Encoder):

Strengths:

- USE provides sentence-level embeddings that capture semantic relationships between entire sentences.
- It is pretrained on a large corpus and can effectively encode diverse linguistic patterns.
- USE excels at capturing semantic similarity and relatedness between sentences.

Weaknesses:

- USE's performance might be limited when dealing with domain-specific or specialized vocabulary.
- The encoding process of USE can be computationally expensive, especially for large datasets.
- USE may not capture fine-grained semantic nuances within sentences.

4.4. Doc2Vec:

Strengths:

- Doc2Vec captures the semantic relationships between documents by generating fixed-length vectors for entire documents.
- It can handle variable-length documents and is robust to the order of words within a document.
- Doc2Vec can capture semantic similarities and differences between documents even when they share no common words.

Weaknesses:

- Doc2Vec requires a substantial amount of training data to learn accurate document embeddings.
- Fine-tuning Doc2Vec for specific tasks may be challenging due to its fixed-length document representations.
- Doc2Vec's performance may be affected by the quality and representativeness of the training corpus.

4.5. LDA (Latent Dirichlet Allocation):

Strengths:

- LDA is a probabilistic model that discovers latent topics within a corpus, allowing for topic-based document similarity analysis.
- It can capture the thematic structure and semantic relationships between documents.

- LDA provides interpretable topic representations that can aid in understanding document relationships.

Weaknesses:

- LDA assumes a bag-of-words representation and ignores the order of words within documents.
- It may struggle with short or noisy documents where topic inference becomes challenging.
- LDA's performance is sensitive to the choice of hyperparameters and the number of topics selected.
- Understanding the strengths and weaknesses of each technique is crucial for selecting the most appropriate approach based on the specific requirements of the document similarity analysis task at hand.

5. **Conclusion**

In this paper, we conducted a comprehensive survey and comparative analysis of the document similarity techniques including BERT, GloVe, USE, Doc2Vec, and LDA. We reviewed the related literature and discussed the underlying principles of each technique. We then presented the feature extraction capabilities of each technique and algorithm.

Based on our findings, we observed that BERT achieved the highest performance in capturing document similarity due to its ability to model contextual information effectively. GloVe and USE also demonstrated strong performance, leveraging word embeddings and sentence encodings, respectively. Doc2Vec exhibited moderate performance, while LDA showed limitations in capturing fine-grained document similarity due to its topic-based approach. These findings have several implications for the field of document similarity analysis and natural language processing. Firstly, it highlights the effectiveness of advanced deep learning techniques such as BERT in capturing semantic relationships between documents. This has significant implications for tasks like information retrieval, document clustering, and recommendation systems. Furthermore, our analysis reveals the importance of considering different feature extraction approaches for document similarity tasks. While word-based models like GloVe and USE provide valuable insights, context-based models like BERT offer a more comprehensive understanding of document semantics.

Looking ahead, there are several avenues for future research in this domain. One area of focus could be the exploration of hybrid models that combine the strengths of multiple techniques to improve document similarity analysis further. Additionally, investigating the transferability of pre-trained models and exploring domain-specific adaptations can enhance performance in specific application contexts.

In conclusion, this research contributes to the understanding of document similarity techniques and provides valuable insights into their strengths and weaknesses. The findings presented here can guide researchers and practitioners in selecting appropriate methods for document similarity tasks and inspire further advancements in the field of natural language processing.

References:

1. H. Choi, J. Kim, S. Joe and Y. Gwon, "Evaluation of BERT and ALBERT Sentence Embedding Performance on Downstream NLP Tasks," 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 2021,

pp. 5482-5487, doi: 10.1109/ICPR48806.2021.9412102.

2. R. Rizk, D. Rizk, F. Rizk, A. Kumar and M. Bayoumi, "A Resource-Saving Energy-Efficient Reconfigurable Hardware Accelerator for BERT-based Deep Neural Network Language Models using FFT Multiplication," 2022 IEEE International Symposium on Circuits and Systems (ISCAS), Austin, TX, USA, 2022, pp. 1675-1679, doi: 10.1109/ISCAS48785.2022.9937531.
3. T. M. Lai, Q. Hung Tran, T. Bui and D. Kihara, "A Simple But Effective Bert Model for Dialog State Tracking on Resource-Limited Systems," ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 2020, pp. 8034-8038, doi: 10.1109/ICASSP40776.2020.9053975.
4. M. Podkorytov, D. Biś, J. Cai, K. Amirizirtol and X. Liu, "Effects of Architecture and Training on Embedding Geometry and Feature Discriminability in BERT," 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 2020, pp. 1-8, doi: 10.1109/IJCNN48605.2020.9206645.
5. Y. Guo, H. Lamaazi and R. Mizouni, "Smart Edge-based Fake News Detection using Pre-trained BERT Model," 2022 18th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Thessaloniki, Greece, 2022, pp. 437-442, doi: 10.1109/WiMob55322.2022.9941689.
6. D. Wu, M. Zhang, C. Shen, Z. Huang and M. Gu, "BTM and GloVe Similarity Linear Fusion-Based Short Text Clustering Algorithm for Microblog Hot Topic Discovery," in IEEE Access, vol. 8, pp. 32215-32225, 2020, doi: 10.1109/ACCESS.2020.2973430.
7. J. Lin and W. Yu, "A Chinese text similarity algorithm based on Yake and neural network," 2022 7th International Conference on Intelligent Informatics and Biomedical Science (ICIIBMS), Nara, Japan, 2022, pp. 230-234, doi: 10.1109/ICIIBMS55689.2022.9971706.
8. T. Shancheng, B. Yunyue and M. Fuyu, "A Chinese short text semantic similarity computation model based on stop words and TongyiciCilin," 2017 6th International Conference on Computer Science and Network Technology (ICCSNT), Dalian, China, 2017, pp. 310-314, doi: 10.1109/ICCSNT.2017.8343708.
9. T. A. Hutajulu, Y. Priyadi and A. Gandhi, "Text Data Processing in Requirement Specifications as a Reference for Similarities Between Use Case Diagrams and Use Case Descriptions for Smart Sleeping Lamp Application Documents," 2022 IEEE World AI IoT Congress (AIIoT), Seattle, WA, USA, 2022, pp. 665-671, doi: 10.1109/AIIoT54504.2022.9817197.
10. Y. Safali, G. Nergız, E. Avaroğlu and E. Doğan, "Deep Learning Based Classification Using Academic Studies in Doc2Vec Model," 2019 International Artificial Intelligence and Data Processing Symposium (IDAP), Malatya, Turkey, 2019, pp. 1-5, doi: 10.1109/IDAP.2019.8875877.
11. I. R. Hendrawan, E. Utami and A. D. Hartanto, "Comparison of Word2vec and Doc2vec Methods for Text Classification of Product Reviews," 2022 6th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE),

Yogyakarta, Indonesia, 2022, pp. 530-534, doi: 10.1109/ICITISEE57756.2022.10057702.

12. M. Bilgin and İ. F. Şentürk, "Sentiment analysis on Twitter data with semi-supervised Doc2Vec," 2017 International Conference on Computer Science and Engineering (UBMK), Antalya, Turkey, 2017, pp. 661-666, doi: 10.1109/UBMK.2017.8093492.
13. H. Arslan, O. Kaynar and S. Şahİn, "Classification of Customer Demands by Using Doc2Vec Feaure Extraction Method," 2019 27th Signal Processing and Communications Applications Conference (SIU), Sivas, Turkey, 2019, pp. 1-4, doi: 10.1109/SIU.2019.8806452.
14. I. Setha and H. Aliane, "Enhancing automatic plagiarism detection: Using Doc2vec model," 2022 International Conference on Advanced Aspects of Software Engineering (ICAASE), Constantine, Algeria, 2022, pp. 1-5, doi: 10.1109/ICAASE56196.2022.9931542.
15. W. Liu and C. Ling, "LDA-lattice aided network coding for two-way relay," 2016 SAI Computing Conference (SAI), London, UK, 2016, pp. 610-614, doi: 10.1109/SAI.2016.7556044.
16. P. Liao, J. Liu, M. Wang, H. Ma and W. Zhang, "Ensemble local fractional LDA for face recognition," 2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE), Zhangjiajie, China, 2012, pp. 586-590, doi: 10.1109/CSAE.2012.6273021.
17. P. Marasamy and S. Sumathi, "Automatic recognition and analysis of human faces and facial expression by LDA using wavelet transform," 2012 International Conference on Computer Communication and Informatics, Coimbatore, India, 2012, pp. 1-4, doi: 10.1109/ICCCI.2012.6158798.
18. H. Zhang and L. Zhou, "Similarity Judgment of Civil Aviation Regulations Based on Doc2Vec Deep Learning Algorithm," 2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Suzhou, China, 2019, pp. 1-8, doi: 10.1109/CISP-BMEI48845.2019.8965709.
19. S. T. Laxmi, R. Rismala and H. Nurrahmi, "Cyberbullying Detection on Indonesian Twitter using Doc2Vec and Convolutional Neural Network," 2021 9th International Conference on Information and Communication Technology (ICoICT), Yogyakarta, Indonesia, 2021, pp. 82-86, doi: 10.1109/ICoICT52021.2021.9527420.
20. S. Pang, T. Ban, Y. Kadobayashi and N. K. Kasabov, "LDA Merging and Splitting With Applications to Multiagent Cooperative Learning and System Alteration," in IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 42, no. 2, pp. 552-564, April 2012, doi: 10.1109/TSMCB.2011.2169056.