

Semantic Coupling of Scientific Literature using sBERT: An Enhanced Model for Systematic Literature Review

Aditi Roy, *SRF, DLIS, University of North Bengal, rs_aditi@nbu.ac.in*

Saptarshi Ghosh, *Professor, DLIS, University of North Bengal, sghosh@nbu.ac.in*

Abstract:

Semantic coupling refers to betweenness among documents having the same textual context. Various AI tools are available for identifying semantically correlated texts for literary warrant grouping. Semantic similarity measures the quantitative distance between two documents in account of likeness. This paper is broadly divided into three segments: firstly, individual document similarity using sentence embedding; second, to identify document pair similarity; and lastly, using Doc2Vec algorithm relevant document identification and retrieval of top k documents matching the query from a document corpus. The highest similarity between sentence pairs from every document is nearly 1, the highest matching between a document pair is 0.798823833, and the least matching is 0.003227258. Lastly, most of the document's similarity ranges between 0.2 to 0. The paper analysed the semantic coupling of documents and their granular components for validation in identifying related documents required for Systematic Literature Review (SLR). This model will help identify correlated texts that may extend possibilities of systematic literature review (SLR) more broadly if the model is implemented through a web interface like AsReview, etc.

Keywords: Systematic Literature Review, Semantic Coupling, sBERT, Cosine Similarity, Semantic Textual Similarity, Doc2Vec

1. Introduction: A literature review critically analyses available literature on a specific topic to provide substantial knowledge. Systematic Literature Review (SLR) includes logical, scientific, and systematic evaluation and methodical representation of published literature to produce high-quality research (Linnenluecke et al., 2020). In order to respond to an articulated question, a systematic literature review (SLR) identifies, picks, and critically evaluates research (Dewey et al., 2016). Thus, satisfying a research query and producing high-yield research SLR is the most crucial task. This paper helps to identify the semantic coupling between scientific

and scholarly documents using machine learning algorithms. Semantic coupling refers to betweenness among documents having the same textual context. Various AI tools are available for identifying semantically correlated texts for literary warrant grouping. Semantic similarity measures the quantitative distance between two documents in account of likeness. Semantic Textual Similarity (STS) is a crucial evaluative indicator in Natural Language Processing (NLP) to assess the meaning of two texts and determine their similarity even when the words used within each text are different. The similarity is not just considered from a lexical perspective that considers character sequences but also has to include the semantic meaning (Prakoso et al., 2021). In this study, we used the Sentence-BERT (Bidirectional Encoder Representations from Transformers) to generate semantically significant sentence embeddings that can be compared using cosine similarity. We also used the Doc2Vec algorithm to retrieve contextually similar documents. This model will help identify correlated texts that extend the possibilities of systematic literature review (SLR) more greatly if the model is implemented through a web interface like AsReview, etc.

This study aims to identify the following:

- Semantic similarity between sentences of a document.
- To measure the STS of a document with another document that belongs to the collection of a repository.
- To trace the importance of a pre-trained model with embedding features for its suitability in identifying related literature for systematic literature review.

2. Literature Review: Scientific Procedures and Rationales for Systematic Literature Reviews (SPAR-4-SLR) protocol developed by expert editors help researchers identify logical methods for Systematic Literature Review and produce high-quality transparent research (Paul et al., 2021). Semantic similarity between pairs of sentences is measured using sentence embedding representation by cosine similarity matrix. A framework was proposed to train and test large-scale datasets in a supervised and unsupervised manner to generate high-quality and contextual similarity scores between two sentences (Sun et al., 2022). Semantically Structured Sentence BERT(S³BERT) provides Semantic Similarity to "learn a decomposition of the sentence embeddings into semantic features, through approximation of a suite of interpretable AMR graph metrics" as well as preserving Neural embeddings (Opitz & Frank, 2022). A more refined form of sentence BERT named Refined SBERT was proposed by Chu et al., which represents

semantic similarity from manifold learning by re-embedding sentence vectors in the ambient space (Chu et al., 2023). Using the Doc2Vec and Cosine Similarity method, semantically similar Philippine Supreme Court case decisions were retrieved with 80% accuracy (Barco et al., 2019). With document embedding method, i.e., Doc2Vec PV DBOW architecture to find similar documents in PubMed database along with comparing the PubMed Related articles (pmra) statistic model that reflected pmra need prior indexing of documents. However, it is only needed in Doc2Vec. Even with knowledge about the documents, it can be similar to PV DBOW architecture (Dynomant et al., 2019). The similarity or dissimilarity between a document pair depends on the presence and absence of terms; a new property added by Oghbaie, M., & Mohammadi Zanjireh, M was that "the similarity degree should increase when the number of present terms increases" (Oghbaie & Mohammadi, 2018).

3. Methodology:

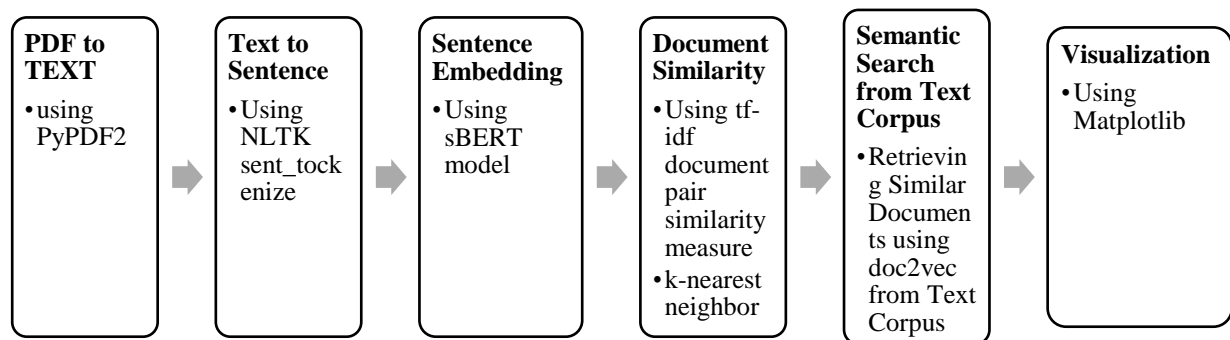


Figure 1: Methodology Followed for the Study

The above process depicts the workflow of activities to extract full text from pdf for preparing sentence tokeniser. NLTK python library is used to determine sentence vectors (tokens). The customised Sentence-BERT model (sBERT) is used for sentence embedding with the extracted text from PDF files. Cosine similarity is measured for identifying document vectors having proximity with the sentence-embedded result. Then, the similarity between document pairs is measured using the tf-idf similarity matrix and cosine similarity measure. Similar documents against each document are also identified by applying the k-nearest Neighbour algorithm. Next, semantically similar documents are retrieved from a corpus of documents using the doc2vec

library in Python. Finally, the visualisation of document similarity is portrayed using the matplotlib library.

As the sample of this study, we have selected the Indian National Science Academy (INSA) collections with 2095 items available at the Institutional Digital Repository (IDR) Hosting Service of the National Digital Library of India, which were available under two collections.

We selected the sample size using the following formula for finite population,

$$n' = \frac{n}{1 + \frac{z^2 \times p(1-p)}{\epsilon^2 N}}$$

Where, z = Z Score, ϵ = Margin of Error, p = Population Proportion, N = Population Size

The sample size calculated is 92 with a confidence level of 95%, margin of error of 10%, population proportion of 50% and population size of 2095. These samples are chosen by a simple random sampling method.

4. Results:

4.1. Sentence Similarity using sBERT:

Sentence embedding represents sentences as vectors to measure textual similarity using metrics like cosine similarity or Euclidean distance. To create high-quality vectors, sentence embedding fused with transformers, i.e., a type of neural network, is used in sBERT to measure the semantic similarity between sentences. The sentence transformer model we used in this study is '*paraphrase-MiniLM-L6-v2*', which maps sentences in 384-dimensional dense vector space (<https://huggingface.co/sentence-transformers/paraphrase-MiniLM-L6-v2>).

The matrix created by sentence transformers represents the matching between each sentence; the highest matrix created is of array size 5026X5026 (File name- BM7_8310), and the lowest matrix is of array size 19X19 (File name- Vol42_3_13_HistoricalNoteGJuleff)

File Name	Highest Cosine Similarity	Cell Numbers
Vol01_1_1_PRay.csv	0.99999994	(0,9)
Vol01_1_2_JRRavetz.csv	0.99999994	(0,19)
Vol01_1_3_MHoskin.csv	0.99999994	(0,15)

Vol01_1_4_DJDSPrice.csv	0.99999994	(0,4)
Vol01_1_5_SNSen.csv	0.99999994	(0,9)
Vol01_1_6_VRonchi.csv	0.99999994	(0,4)
Vol01_1_7_VSubbarayappa.csv	0.99999994	(0,25)
Vol01_1_8_AKBag.csv	0.99999994	(0,14)
Vol01_1_9_BRensch.csv	0.99999994	(0,9)
Vol01_2_1_WPetri.csv	0.99999994	(0,26)

Table 1: Highest Similarity Value between Sentence Pairs

Although Table 1 shows only ten documents with the highest similarity value for each document, the highest similarity is the same. The column cell numbers that represent the sentence pairs with the most semantic similarity.

File Name	Lowest Value	Cell Number
BM7_8310.csv	-0.3414742	(1258,55)
BM7_8313.csv	-0.3323907	(229,39)
BM8_8401.csv	-0.3196548	(569,27)
BM8_8404.csv	-0.31555206	(0,394)
BM8_8404.csv	-0.31555206	(393,1)
Vol05_2_4_DPAgrawal.csv	-0.29516277	(274,17)
Vol39_3_4_ABasu.csv	-0.29511765	(422,58)

Table 2: Lowest Similarity Value between Sentence Pairs

Table 2 represents the lowest similarity score between sentence pairs as in cell numbers. The similarity scores -1 represents an opposite sentence, and the negative values in Table 2 show these sentences are pretty dissimilar but not opposite. This result is justified because all the sentences in a particular article need not be homogenous. However, the highest similarity between sentence pairs from every document is nearly 1.

4.2.Document Similarity between each pair of documents:

Document similarity is one of the crucial tasks in Natural Language Processing (NLP) for the representation of semantic or contextual similarity between documents in a collection. In this study, we measured the similarity between each document pair by using cosine similarity and tf-idf matrix.

Document 1	Document 2	Similarity Score
Vol09_2_1_RCGupta.txt	Vol09_2_4_RCGupta.txt	0.798823833
Vol3_2005_08_AZDAHA PAIKAR THE COMPOSITE IRON BROZE CANNON AT MUSA BURJ OF GOLCONDA FORT.txt	Vol4_2005_03_CANNONS OF EASTERN INDIA.txt	0.652362415
Vol02_2_1_VDMarza.txt	Vol09_2_6_VDMarza.txt	0.537527657
Vol02_2_3_RCGupta.txt	Vol09_2_1_RCGupta.txt	0.483285038
Vol02_2_3_RCGupta.txt	Vol09_2_4_RCGupta.txt	0.461237328
BM7_8310.txt	BM9_8403.txt	0.428625678
Vol36_1and2_1_RBalasubramanian.txt	Vol39_1_3_MIDass.txt	0.408614255

Table 3: Document Pair Similarity Score ≥ 0.4

Table 3 represents the top document pair similarity, and Table shows the least similarity, reflecting that the highest matching between a document pair is 0.798823833 and the least matching is 0.003227258 (Table 4).

Document 1	Document 2	Similarity Score
BM8_8407.txt	Vol09_2_1_RCGupta.txt	0.007716248
Vol09_2_4_RCGupta.txt	Vol42_3_13_HistoricalNoteGJuleff.txt	0.007445999
Vol45_3_11_News.txt	Vol45_3_14_EnglishtextCha.txt	0.007092925
BM7_8308.txt	Vol09_2_1_RCGupta.txt	0.006907021
BM8_8407.txt	Vol45_3_14_EnglishtextCha.txt	0.006722215
BM5_7913.txt	Vol09_2_1_RCGupta.txt	0.006220515
Vol42_3_13_HistoricalNoteGJuleff.txt	Vol45_3_14_EnglishtextCha.txt	0.005513658
Vol45_3_14_EnglishtextCha.txt	Vol4_2005_10_HISTORICAL NOTES_3.txt	0.005445721
Vol09_2_1_RCGupta.txt	Vol42_3_13_HistoricalNoteGJuleff.txt	0.004104872
BM7_8308.txt	Vol45_3_14_EnglishtextCha.txt	0.003227258

Table 4: Document Pair Similarity Score of least matched ten documents

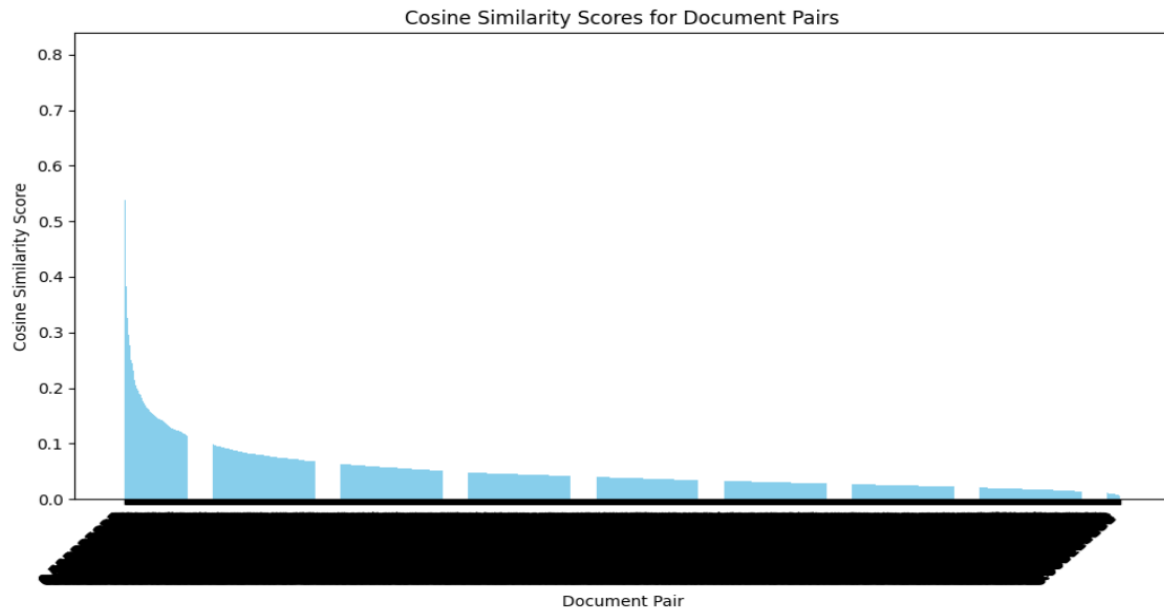


Figure 2: Document Pair and Similarity Score

This figure represents the Cosine Similarity between each document pair and shows that most of the document's similarity ranges between 0.2 and 0. Therefore, the collection of documents is mostly dissimilar, supporting the collection's heterogeneity.



Figure 3: Document Similarity between each Document Pair

Figure 3 represents the cosine similarity of every document pair, i.e., the cosine matrix of array size 92X92. The above similarity matrix depicts a dense population with values ranging from 0.2 to 0. The vertical right side represents the similarity score, manifested in the diagram as

tiny dots with shades of various colours. Blue represents dissimilarity between documents, whilst yellow shows the closeness between the documents.

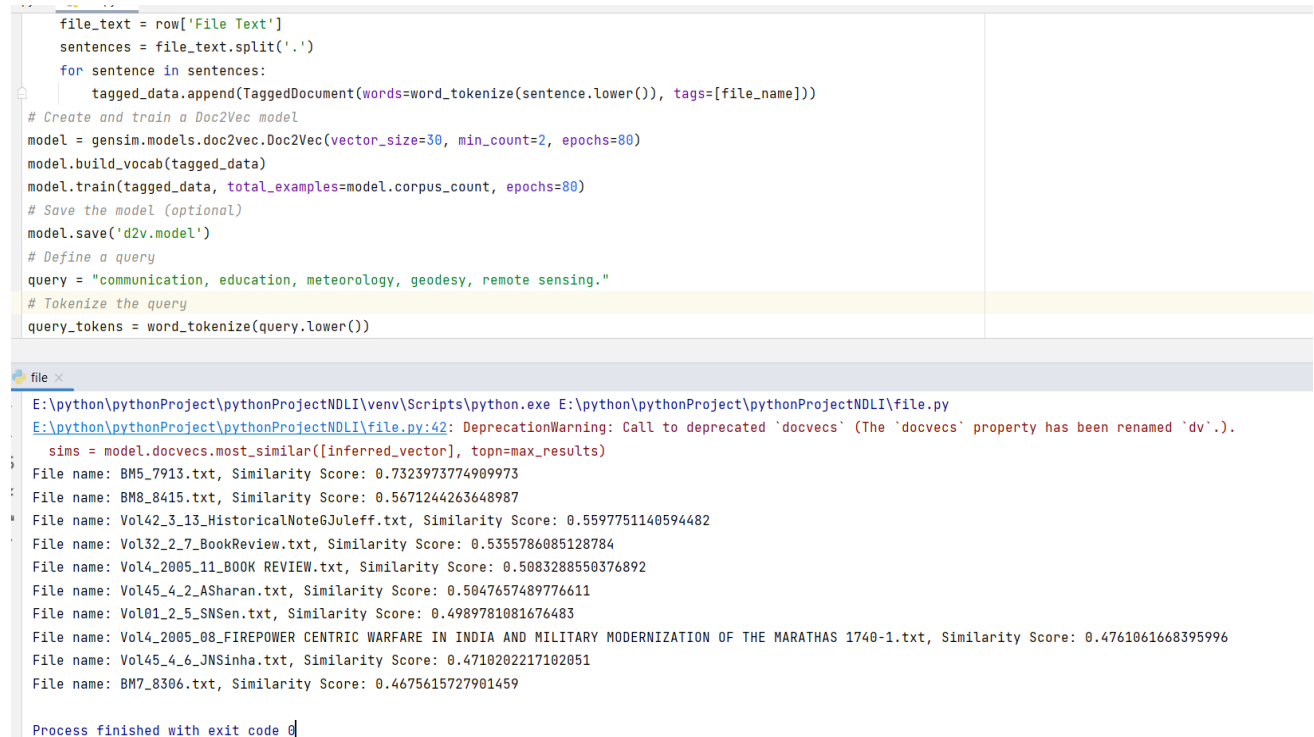
File Name	Similar Document
BM8_8416.txt	BM8_8416.txt, BM7_8310.txt, Vol39_3_4_ABasu.txt, BM7_8311.txt, BM8_8402.txt
Vol01_1_1_PRay.txt	Vol01_1_1_PRay.txt, Vol02_1_2_BVSubbarayappa.txt, Vol01_1_7_VSubbarayappa.txt, Vol04_1And2_5_TMPMahadevan.txt, Vol01_1_9_BRensch.txt
Vol01_1_2_JRRavetz.txt	Vol01_1_2_JRRavetz.txt, Vol22_4_8_SPGupta.txt, Vol04_1And2_5_TMPMahadevan.txt, Vol01_1_6_VRonchi.txt, Vol01_2_5_SNSen.txt
Vol01_1_3_MHoskin.txt	Vol01_1_3_MHoskin.txt, Vol22_3_5_SCKak.txt, Vol01_2_1_WPetri.txt, Vol01_1_5_SNSen.txt, Vol32_3_3_YOhashi.txt
Vol01_1_4_DJDSPrice.txt	Vol01_1_4_DJDSPrice.txt, Vol04_1And2_11_KSShukla.txt, Vol22_3_5_SCKak.txt, Vol01_2_5_SNSen.txt, Vol04_1And2_5_TMPMahadevan.txt

Table 5: Similar Documents using the k-nearest Neighbour Algorithm

Table 5 shows the list of similar documents against 5 sample documents using the k nearest neighbour machine learning algorithm by vectorising each document with tf-idf vectoriser and a model with `knn = NearestNeighbors(n_neighbors=n_neighbors, metric='cosine')` to identify the nearest neighbour document of each sample. The second column (Similar Document) of

Table 5 represents the five nearest documents for all occurrences as the variable `n_neighbors` is limited to 5.

4.3. Retrieving similar documents from the corpus of text:



```

file_text = row['File Text']
sentences = file_text.split('.')
for sentence in sentences:
    tagged_data.append(TaggedDocument(words=word_tokenize(sentence.lower()), tags=[file_name]))

# Create and train a Doc2Vec model
model = gensim.models.doc2vec.Doc2Vec(vector_size=30, min_count=2, epochs=80)
model.build_vocab(tagged_data)
model.train(tagged_data, total_examples=model.corpus_count, epochs=80)

# Save the model (optional)
model.save('d2v.model')

# Define a query
query = "communication, education, meteorology, geodesy, remote sensing."

# Tokenize the query
query_tokens = word_tokenize(query.lower())

sims = model.docvecs.most_similar([inferred_vector], topn=max_results)
File name: BM5_7913.txt, Similarity Score: 0.7323973774909973
File name: BM8_8415.txt, Similarity Score: 0.5671244263648987
File name: Vol42_3_13_HistoricalNote6Juleff.txt, Similarity Score: 0.5597751140594482
File name: Vol32_2_7_BookReview.txt, Similarity Score: 0.5355786085128784
File name: Vol4_2005_11_BOOK REVIEW.txt, Similarity Score: 0.5083288550376892
File name: Vol45_4_2_Asharan.txt, Similarity Score: 0.5047657489776611
File name: Vol01_2_5_SNSen.txt, Similarity Score: 0.4989781081676483
File name: Vol4_2005_08_FIREPOWER CENTRIC WARFARE IN INDIA AND MILITARY MODERNIZATION OF THE MARATHAS 1740-1.txt, Similarity Score: 0.4761061668395996
File name: Vol45_4_6_JNSinha.txt, Similarity Score: 0.4710202217102051
File name: BM7_8306.txt, Similarity Score: 0.4675615727901459

Process finished with exit code 0

```

Figure 4: Retrieval of Similar Documents satisfying the query

Doc2Vec algorithm in machine learning is used to retrieve contextually similar documents by vectorising each document and then analysing the similarity between the collection of documents. We trained the model with all the sample documents and retrieved semantically similar documents that satisfy the query we put in the code line. Figure 4 shows ten relevant documents from the corpus of 92 documents with a similarity score against the query. The query was inputted as a string, and embedded 'inferred vector' in Doc2Vec predicted a ranked list of k documents closely related to the assigned query. This result conforms with the undertone of the literature similarity search required for a systematic literature review (SLR).

5. Discussion:

The aim of the study is to develop a prototype for automated SLR using ML algorithms. In case of digital libraries, documents can be stored in a repository with varied collections. The specific goal is to find the top^K related literature having Semantic Textual Similarity (STS). Hence the work is accomplished using hierarchical approach, that flows in three segments, first by

identifying '**semantically similar sentences**' using sentence embedding, followed by *identifying 'document to document similarity'*, and lastly '*retrieval of top k documents from the document corpus through matching queries*'. The similarity score is paramount for identifying the homogeneity or heterogeneity of the corpus of documents for STS. In the same way, each document at the sentence level measures the closeness of sentences within the document. Sentence embedding is done with Sentence BERT (sBERT) algorithm, for identifying the contextual value of every document using cosine similarity. From the 'document pair similarity measure', the heterogeneity of the document collection (selected sample for the study) has been identified. If the articles had higher similarity scores that would have supported near-duplicate documents, plagiarised papers and recurring concepts that only lead to the overload of conceptually similar documents and storage issues. Hence, the heterogeneous collection of documents supports better search results and document quality (Varol et al.,2015). From a heterogeneous document pool, identifying relevant literature for literature review is one of the decisive tasks of a researcher. Thus, our study extends help in reviewing the literature systematically by identifying the most relevant documents from a corpus and thus may satisfy the user query, as shown in Figure 3. The computing algorithm seeks the document storage for reading, writing, and executing programme codes. However, all these codes are executed at the backend and thus a frontend is required for visualization of output/outcome. Thus, the web interface of the digital repository may be created using FLASK or Tornado framework for python which will provide the web-based GUI of the backend result.

6. Conclusion:

Every ML algorithm has its scope; thus, sBERT has limited scope in measuring document similarity and semantic coupling of documents. The corpus of this study is heterogeneous, and the usage of sBERT for sentence similarity measures yields satisfactory output. The sentence transformer model used in this study is a truncated version of BERT, i.e., 'paraphrase MiniLM model' with six layers, 384 hidden layers and version 2. This language model is one of the modest 'sentence vectorizers' for contextual similarity, which leads to document similarity measurement to assess the homogeneity or heterogeneity of the documents in the corpus. This measure reflects that the corpus is heterogeneous and in tune with the study's objective. The performed tests detected sensitivity within the document corpus. The unusable words within each document are responsible for the quality of results (Trstenjak et al.,2014). The tf-idf matrix with cosine metric, k nearest neighbour algorithm identified near matches of each sample

document. Doc2Vec is the algorithm which vectorises a document at the paragraph level and tries to identify the semantic tone of related paragraphs. We used the algorithm to retrieve contextually similar documents supporting logical and systematic literature review (SLR). If the model is implemented through a web interface within a large corpus of documents, it will help to identify semantically similar literature to produce high-quality research. Therefore, the paper projects a bird's-eye view of the semantic coupling of documents and their granular components to validate identifying related documents required for SLR.

References:

1. Barco Ranera, L. T., Solano, G. A., & Oco, N. (2019). Retrieval of Semantically Similar Philippine Supreme Court Case Decisions using Doc2Vec. *2019 International Symposium on Multimedia and Communication Technology (ISMAC)*, 1-6.
2. Chu, Y., Cao, H., Diao, Y., & Lin, H. (2023). Refined SBERT: Representing sentence BERT in manifold space. *Neurocomputing*, 555, 126453. <https://doi.org/https://doi.org/10.1016/j.neucom.2023.126453>
3. Dewey, A. & Drahota, A. (2016). Introduction to systematic reviews: online learning module *Cochrane Training* <https://training.cochrane.org/interactivelearning/module-1-introduction-conducting-systematic-reviews>
4. Dymant, E., Darmoni, S. J., Lejeune, É., Kerdelhué, G., Leroy, J.-P., Lequertier, V., Canu, S., & Grosjean, J. (2019). Doc2Vec on the PubMed corpus: study of a new approach to generate related articles. *ArXiv*, *abs/1911.11698*.
5. Linnenluecke, M. K., Marrone, M., & Singh, A. K. (2020). Conducting systematic literature reviews and bibliometric analyses. *Australian Journal of Management*, 45(2), 175–194. <https://doi.org/10.1177/0312896219877678>
6. Oghbaie, M., & Mohammadi Zanjireh, M. (2018). Pairwise document similarity measure based on present term set. *Journal of Big Data*, 5(1). <https://doi.org/10.1186/s40537-018-0163-2>

7. Opitz, J., & Frank, A. (2022). SBERT studies Meaning Representations: Decomposing Sentence Embeddings into Explainable AMR Meaning Features. *ArXiv*, *abs/2206.07023*.
8. Paul, J., Lim, W. M., O'Cass, A., Hao, A. W., & Bresciani, S. (2021). Scientific procedures and rationales for systematic literature reviews (SPAR-4-SLR). *International Journal of Consumer Studies*, 45(4), O1-O16. <https://doi.org/https://doi.org/10.1111/ijcs.12695>
9. Prakoso, D. W., Abdi, A., & Amrit, C. (2021). Short text similarity measurement methods: a review. *Soft Computing*, 25(6), 4699–4723. <https://doi.org/10.1007/s00500-020-05479-2>
10. Sun, X., Meng, Y., Ao, X., Wu, F., Zhang, T., Li, J., & Fan, C. (2021). Sentence Similarity Based on Contexts. *Transactions of the Association for Computational Linguistics*, 10, 573-588.
11. Trstenjak, B., Mikac, S., & Donko, D. (2014). KNN with TF-IDF-based Framework for Text Categorization. *Procedia Engineering*, 69, 1356-1364. <https://doi.org/10.1016/j.proeng.2014.03.129>
12. Varol, C., & Hari, S. (2015). Detecting near-duplicate text documents with a hybrid approach. *Journal of Information Science*, 41(4), 405-414. <https://doi.org/10.1177/0165551515577912>