A Comparison of Vector-based Approaches for Document Similarity Using the RELISH Corpus

Rohitha Ravinder[1,2], Tim Fellerhof[1,3], Vishnu Dadi[1,4], Lukas Geist[1,4], Guillermo Rocamora[1,5], Muhammad Talha[1,4], Dietrich Rebholz-Schuhmann[1,6] and Leyla Jael Castro[1]

[1] *ZB MED Information Centre for Life Sciences, Gleueler Str. 60, Cologne, 50931, Germany*
[2] *Bonn-Aachen International Centre for Information Technology (B-IT), University of Bonn, Friedrich-Hirzebruch-Allee 6, Bonn, 53115, Germany*
[3] *Heinrich-Heine University Düsseldorf, Universitätsstraße 1, Düsseldorf, 40225, Germany*
[4] *Hochschule Bonn-Rhein-Sieg, Grantham-Allee 20, Sankt Augustin, 53757, Germany*
[5] *Universidad de Murcia, Avda. Teniente Flomesta 5, Murcia, 30003, Spain*
[6] *University of Cologne, Albertus-Magnus-Platz, Cologne, 50923, Germany*

Abstract

The continuously increasing number of biomedical scholarly publications makes it challenging to construct document recommendation algorithms that can efficiently navigate through literature. Such algorithms would help researchers in finding similar, relevant, and related publications that align with their research interests. Natural Language Processing offers various alternatives to compare publications, ranging from entity recognition to document embeddings. In this paper, we present the results of a comparative analysis of vector-based approaches to assess document similarity in the RELISH corpus. We aim to determine the best approach that resembles relevance without the need for further training. Specifically, we employ five different techniques to generate vectors representing the text in the documents. These techniques employ a combination of various Natural Language Processing frameworks such as Word2Vec, Doc2Vec, dictionary-based Named Entity Recognition, and state-of-the-art models based on BERT. To evaluate the document similarity obtained by these approaches, we utilize different evaluation metrics that account for relevance judgment, relevance search, and re-ranking of the relevance search. Our results demonstrate that the most promising approach is an in-house version of document embeddings, starting with word embeddings and using centroids to aggregate them by document.

Keywords

Document similarity, Document relevance, Word embeddings, Named Entity Recognition, Recommendation systems

1. Introduction

Recommendation systems have shown to be a successful method to cope with information overload and retrieval, making it easier to navigate the ever-expanding public information available online. Such systems have become a key application for scientific publications and their corresponding repositories [1]. Researchers mainly use author-provided keywords, titles, author names, and references to locate new scientific literature, making document recommendation a content-based approach [2]. Over the years, numerous studies have been conducted to develop effective methods for recommending scientific literature. One of the earliest content-based recommendation systems was introduced in the CiteSeer [3] project, which utilized keywords matching, Term Frequency-Inverse Document Frequency (TF-IDF) for word information, and Common Citation-Inverse Document Frequency (CCIDF) for citation information. Other examples include Science Concierge [2], which employs Latent Semantic Analysis (LSA) and Rocchio

Proceedings SeWebMeDa-2023: 6th International Workshop on Semantic Web solutions for large-scale biomedical data analytics, May 29, 2023, Hersonissos, Greece
ljgarcia@zbmed.de (LJ. Castro)
0009-0004-4484-6283 (R. Ravinder); 0000-0002-8725-1317 (T. Fellerhof); 0000-0002-3082-7522 (V. Dadi); 0000-0002-2910-7982 (L. Geist); 0000-0002-4795-3648 (G. Rocamora); 0000-0002-1018-0370 (D. Rebholz-Schuhmann); 0000-0003-3986-0510 (LJ. Castro)

Algorithms; and PURE [4], a content-based recommendation system designed to work with PubMed articles. In [5], Lin and Wilbur presented a probabilistic topic-based content similarity model for PubMed articles. Recent experiments have focused on developing a content-based recommendation system that recommends articles from PubMed corresponding to datasets from Gene Expression Omnibus (GEO) [6].

All of the aforementioned systems are document-to-document recommendation systems; they offer recommendations for documents that are similar to a particular, i.e., reference, document. Due to the lack of document-to-document relevance or similarity training datasets, it is a common approach to use document-to-topic relevance or similarity collections (e.g., those used in the Text Retrieval Conference – TREC). To overcome this situation, the RElevant LIterature SearcH (RELISH) consortium created a document-to-document relevant dataset [7]. The RELISH dataset consists of PubMed articles manually annotated with respect to relevance between pairs of articles. PubMed [8] is a well-known database consisting of millions of biomedical literature references, providing a comprehensive resource for literature search and analysis. The RELISH corpus majorly acts as a benchmark for comparing, improving, and translating newly developed literature search techniques.

Determining similarity between one document with respect to a reference one mainly boils down to analyzing the similarities between their corresponding texts - a task that is often tackled using Natural Language Processing (NLP). To achieve this, the textual data from these documents needs to be transformed into a structured form to enable machine-based document-to-document similarity approaches. One option is using vectors, i.e., representations of words that encode words' meaning, making it easier to, for instance, finding close words based on their distance in the vector space, which could also be similar in meaning or context. To generate these vectors, various state-of-the-art NLP methods, such as Word2Vec [9], Doc2Vec [10], Named Entity Recognition (NER), and BERT [11]-based models, have been developed using probabilistic vector space models.

Document comparison based on text-based vectors require a comparable measure of document similarity. Metrics such as the Dice coefficient, Jaccard coefficient, Overlap coefficient, Cosine similarity, Okapi Best Matching 25 (BM25), Term Frequency-Inverse Document Frequency (TF-IDF), and more sophisticated techniques that account for context, subject, and term-dependencies can be used to assess these similarities.

In this paper, we describe and compare five different techniques to generate vectors for every PubMed article of the RELISH corpus using NLP frameworks, namely: word2doc2vec, doc2vec, whatizit-dictionary, hybrid-doc2vec, and BERT-based approaches. As per the RELISH corpus, the relevance annotation for each article with respect to the others is categorized into three classes: relevant (definitely relevant), partial (partially relevant), and non-relevant. We employ this relevance categorization as our ground truth to evaluate the performance of our five different approaches. We made use of the Cosine similarity metric to calculate the document similarities and further analyze the document-to-document similarity using three evaluation approaches, namely: distribution-based analysis, normalized Discounted Cumulative Gain, and precision, to account for the categorization, ranking, and relevance of the similarity search.

2. Methods and Materials

2.1. RELISH corpus

In this study, we used the RElevance LIterature SearcH (RELISH) consortium [8], which is an expert-curated database for document similarity in biomedical literature that includes over 180,000 PubMed articles. The database v1 was downloaded from its corresponding FigShare record [12] on the 24th of January 2022. The database is in a JSON format and contains PubMed Ids (PMIDs) together with their corresponding document-to-document relevance assessments with respect to other PMIDs. The relevance is categorized into three categories; “relevant,” “partial,” or “non-relevant”.

2.2. Data preprocessing

Using the BioC API, we retrieved an XML file containing the PMID, title, and abstract for each unique entry in the RELISH JSON file. We also recorded the missing PMIDs whose retrieval failed or whose title/abstract was unavailable. In total, we retrieved about 163,189 XML files. This dataset was also transformed into a TSV file consisting of three columns, namely: PMID, title, and abstract, where the text in the title and abstract was preprocessed. Two different preprocessing pipelines were followed, depending on the vector-based approach used. The preprocessing pipeline for word2doc2vec, doc2vec, and the hybrid-doc2vec approaches included converting the text into lower case, removal of punctuations, removal of structural words (e.g., "BACKGROUND:," "CONCLUSION:," "METHODS:"), followed by tokenization.

For the BERT-based approach, only white spaces and newlines between the text were removed. The titles and abstracts for each document were combined as a single text input for all the approaches. In addition to this, another TSV file was created from the raw original RELISH JSON file consisting of three columns, namely: PMID1 (reference article), PMID2 (assessed article), and relevance (relevance between the two documents). For simplicity, the relevance was assessed by a score of 0, 1, and 2 for articles that were non-relevant, partially relevant, and definitely relevant, respectively. This TSV file was used for the evaluation tasks, and excluded all RELISH reference articles having a multiple relevance assessment due to being evaluated by more than one annotator.

2.3. Vector-based approaches

In this study, we evaluate document-to-document similarity by generating embeddings using five different approaches, namely word2doc2vec, doc2vec, whatizit-dictionary, hybrid-doc2vec, and BERT-based approach. We assess the similarity using the cosine similarity metric. The input for the dictionary-based NER approach (whatizit-dictionary) was the RELISH XML files, whereas the input for all the other four approaches was the RELISH TSV file consisting of three columns (PMID | title | abstract).

2.3.1. Word2doc2vec

In this approach, we made use of the Word2Vec framework [9], which is a two-layer neural network trained to reconstruct linguistic contexts of words, with each unique word being assigned to a corresponding vector. We generated word embeddings using the Word2Vec module in the Gensim Python library [13]. The Word2Vec model was trained on our RELISH dataset, and document embeddings were generated from these word embeddings by calculating the centroid of all word embeddings in each title and abstract of a document. The model and the corresponding embeddings were generated using various sets of hyperparameter combinations, as shown in Table 1.

2.3.2. Doc2vec

This approach utilized the Doc2Vec framework [10], which is an extension of the Word2Vec neural network that generates a numeric representation of a document regardless of its length. We employed the Doc2Vec module [14] of the Gensim python library to generate document embeddings. The Doc2Vec model was trained on the RELISH corpus with the same set of hyperparameter combinations as the word2doc2vec approach, shown in Table 1, was used to generate document embeddings.

2.3.3. Whatizit-dictionary

This approach utilizes a dictionary-based Named Entity Recognition method that employs the Whatizit tool. Whatizit is a text processing system based on MONQjfa, a deterministic and non-

deterministic finite automata for Java [15]. For our study, we make use of a minimal dockerized version of Whatizit, focusing mainly on the automata part of the MONQjfa [16]. The input for Whatizit is a dictionary that recognizes entities in a text, and normalizes them against a controlled vocabulary. In order to annotate the RELISH XML files, we used Medical Subject Headings (MeSH) [17] as our controlled vocabulary. The annotated XML files were then used for generating embeddings in the form of Term Frequency-Inverse Document Frequency (TF-IDF) vectors, where we evaluated the relevance of each MeSH term to a particular RELISH article in the entire RELISH corpus.

2.3.4. Hybrid-doc2vec

This is a hybrid approach that explores the combination of a dictionary-based Named Entity Recognition using the Whatizit tool and the Doc2Vec framework. The main idea of this approach is to transform annotated XML files after the Whatizit processing into a plain text dataset, preprocess the text, and then use the Doc2Vec model. Annotated MeSH terms in the article's title and abstract are replaced by their MeSH ID, and converted into plain text. This step is followed by applying a standard Doc2Vec process (similar to the Doc2Vec approach) to generate embeddings using the same set of hyperparameter combinations as shown in Table 1.

2.3.5. BERT-based

This approach explores and assesses document-to-document similarity using Bidirectional Encoder Representations (BERT) [11] - based embeddings. BERT from Transformers [18] is a transformer-based machine learning technique for NLP pre-training developed by Google. We used the Sentence-Transformers package [19] to run experiments with BERT models. In this approach, we used two state-of-the-art BERT models relevant to the biomedical domain: BioBERT [20] and SciBERT [21]. BioBERT is a pre-trained language representation model based on the BERT model, which was trained on a large corpus of biomedical text (PubMed abstracts and PMC full-text articles). We used two different versions of the BioBERT model by DMIS-Lab [22]: BioBERT-Base-cased-v1.1 [23] and BioBERT-Large-cased-v1.1 [24]. We obtained vectors of size 768 using the BioBERT-Base-cased-v1.1, and vectors of size 1024 using the BioBERT-Large-cased-v1.1. SciBERT is another pre-trained language model based on BERT but trained on a large corpus of scientific text. This model was trained on 1.14M full-paper corpus from semanticscholar.org [25]. We used the cased version model for SciBERT: scibert_scivocab_cased [26]. Similar to BioBERT-Base, we obtained vectors of size 768 for this model.

Table 1 summarizes the set of hyperparameter combinations that were used to generate the embeddings using the three approaches: word2doc2vec, doc2vec, and hybrid-doc2vec. The parameters include the training algorithm, epochs, min_count, vector_size, and window_size. The epochs and min_count were kept constant with values of 15 and 5, respectively. The training algorithm parameter was varied based on the approach: skip-gram (sg: 1) or the continuous bag of words (cbow: 0) for word2doc2vec, and distributed memory (dm: 1) or the distributed bag of words (dbow: 0) for doc2vec and the hybrid-doc2vec approach. Three different values were used for the vector_size: 200, 300, and 400, as well as three different values for the window_size: 5, 6, and 7.

Table 1
Hyperparameter combinations for word2vec, doc2vec, hybrid-doc2vec approaches

Algorithm	Vector size	Window
cbow	200	5, 6, 7
cbow	300	5, 6, 7
cbow	400	5, 6, 7
skip-gram	200	5, 6, 7

skip-gram	300	5, 6, 7
skip-gram	400	5, 6, 7

2.4. Hyperparameter optimization

We calculated the cosine similarity for all the existing pairs of PMIDs per the original RELISH JSON file and created a four-column matrix (PMID1 | PMID2 | relevance | cosine similarity). We then used a distribution-based analysis to obtain the best hyperparameter configuration. This analysis aims to provide a visual aid to understand the cosine similarity tendency for a given model, as well as to understand how a given model behaves with respect to each relevance category (definitely relevant, partially relevant, and non-relevant).

Firstly, we build a counting table in the form of a four-column matrix consisting of a cosine similarity interval ranging from 0 to 1 in steps of 0.01, giving us a total of 101 intervals, count of 0's, count 1's, and count of 2's. The purpose of this counting table is to represent how many comparisons of each relevance category are found in each discrete cosine interval. We believe that an optimal similarity model will tend to have non-relevant values (0's) in the lowest values of cosine similarity intervals, while expecting the definitely relevant values (2's) to be in the higher cosine similarity intervals. We visually studied this tendency for each set of hyperparameter combinations by plotting the counting table as a histogram distribution between the relevance counting vs. cosine intervals for each of the four approaches. We also included the possibility to normalize the distributions so that the cumulative sum of each category adds up to 1.

We further used this analysis to select the optimal model from our set of hyperparameter combinations for each approach by proposing a ROC One vs. All approach. The task at hand was viewed as a problem resembling a multi-class classification problem, where we attempted to categorize whether a cosine similarity value corresponds to a definitely relevant, partially relevant, or a non-relevant pair of documents. We proceeded by converting our problem to a simple binary classification problem by combining both the partially relevant (1's) and definitely relevant (2's) category into a single "relevant" category, thereby optimizing for a relevant vs. non-relevant classification. We calculated the True Positive Rate (TPR) and False Positive Rate (FPR) using the formula stated by equations 1 and 2 from the typical classification metrics, consisting of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). These values were calculated from the counting table. For a given cosine similarity, called CI_x, we calculated:

- TP: sum of the number of relevant documents for cosine intervals greater than or equal to CI_x.
- FP: sum of the number of non-relevant documents for cosine intervals greater than or equal to CI_x.
- FN: sum of the number of relevant documents for cosine intervals smaller than CI_x.
- TN: sum of the number of non-relevant documents for cosine intervals smaller than CI_x.

$$TPR = \frac{TP}{TP + FN} \quad (1)$$

$$FPR = \frac{FP}{FP + TN} \quad (2)$$

From the calculated TPR and FPR, we plotted the ROC curve as a representation of the TPR against the FPR and calculated the Area Under the Curve (AUC). The calculated AUC values act as a good measurement of the overall model performance; we used this metric to compare and choose which hyperparameter combination performed best in discriminating between relevant vs. non-relevant pairs of documents for our approaches. For each of our approaches, we ultimately only used the model with the optimal hyperparameters for further downstream evaluation approaches.

2.5. Evaluation

To evaluate our five approaches, we made use of the above-mentioned cosine similarity four-column matrix for all the existing pairs of PMIDs (PMID1 | PMID2 | relevance | cosine similarity) along with their relevance scores as per the RELISH JSON file. We evaluated the similarity scores with respect to the RELISH relevance assessment for all approaches using normalized Discounted Cumulative Gain (nDCG@N) and precision@N.

2.5.1. normalized Discounted Cumulative Gain (nDCG)

This is an evaluation metric used to rank the recommendations in a document recommendation system based on the relevance. For each of our five approaches, we used the existing pairs of PMIDs with its relevance score (as per the original RELISH JSON) and cosine similarity score between those pairs (PMID1 | PMID2 | relevance | cosine similarity) as input to this evaluation algorithm. The nDCG scores were calculated based on two scores: Discounted Cumulative Gain (DCG) and the ideal Discounted Cumulative Gain (iDCG). In order to account for these two scores, this algorithm works by creating two intermediary matrices, DCG and iDCG. The DCG matrix sorts each PMID entry based on the cosine similarity (from highest to lowest), whereas the iDCG matrix sorts each PMID entry based on the relevance scores (2's, 1's, 0's in that order). For every PMID, the DCG@N and the iDCG@N scores are then calculated using the DCG and the iDCG matrices based on the formula as stated by equations 3 and 4, respectively, where 'N' and 'n' stand for the number of documents for which we intend to calculate the nDCG score, and 'i' stands for the ith document in our document set. The 'rel' accounts for the relevance score between the pair of PMIDs (PMID1 and PMID2).

$$\mathrm{DCG@N} = \sum_{i=0}^{n} \frac{2^{rel_i} - 1}{log_2(i+1)} \tag{3}$$

$$\mathrm{iDCG@N} = \sum_{i=0}^{|rel_n|} \frac{2^{rel_i} - 1}{log_2(i+1)} \tag{4}$$

$$\mathrm{nDCG@N} = \frac{DCG@N}{iDCG@N} \tag{5}$$

In our case, the values of N we used were 5, 10, 15, 20, 25, and 50. Lastly, we calculated the nDCG@N scores for each PMID by dividing the DCG scores by the iDCG scores as stated by equation 5, and took the average nDCG scores for all the values of N for all five vector approaches.

2.5.2. precision@N

This performance metric accounts for the fraction of relevant instances among the retrieved instances, in our case, amongst the N retrieved documents. We take as input the cosine similarity four-column matrix (PMID1 | PMID2 | relevance | cosine similarity) which includes the relevance scores that come from the original RELISH JSON as our ground truth. We start by sorting the cosine similarity scores for each reference PMID (highest to lowest) and take the top N similar documents, where N is the precision parameter that takes values of 5, 10, 15, 20, 25, and 50, and count the number of true positives. Amongst the top N similar documents, we consider a document (assessed PMID) to be a true positive if it has a relevance score of 2 (indicating definite relevance) with respect to the reference PMID in the relevance scores file. Equation 6 states the formula for calculating the precision@N.

$$P@N = \frac{|true\ positives|}{N} \quad (6)$$

We repeat this step for all the documents-to-document pairs in the corpus and calculate the average precision scores for each N value for all five vector approaches.

3. Results and Discussion

In this study, we evaluated various approaches for measuring similarity between scientific articles using different vector representations. Table 2 displays the AUC scores obtained from the distribution-based analysis for the BERT-based models that we used. We observed that the scibert-scivocab-cased model outperformed the other two models, with an AUC score of 0.65. Based on this finding, we selected scibert as our optimal BERT-based model for the other two evaluation approaches.

Table 2
Hyperparameter combinations for BERT-based models

Model	Vector size	AUC
biobert-base-cased-v1.1	786	0.61
biobert-large-cased-v1.1	1024	0.60
scibert-scivocab-cased	786	0.65

We also obtained the optimal hyperparameters for the word2vec, doc2vec, and hybrid-doc2vec approaches through the distribution-based analysis, as shown in Table 3. Tables 4 and 5 present the results of the two other evaluation approaches with different vector representations.

Table 3
Optimal hyperparameters for word2vec, doc2vec, and hybrid-doc2vec approaches based on distribution analysis

Approach	dm/sg	Epochs	Min count	Vector size	Window	Workers	AUC
word2doc2vec	1	15	5	400	7	8	0.6046
doc2vec	1	15	5	200	5	8	0.05960
hybrid-doc2vec	1	15	5	200	6	8	0.5990

Our findings revealed that the word2doc2vec approach outperformed all the other approaches in terms of nDCG and precision scores, followed by the doc2vec approach and the hybrid-doc2vec approach. On the other hand, the traditional TF-IDF method using the whatizit-dictionary approach was the least effective out of all.

Table 4
Normalized Discounted Cumulative Gain (nDCG@N)

Approach	nDCG@5	nDCG@10	nDCG@15	nDCG@20	nDCG@25	nDCG@50
word2doc2vec	**0.7816**	**0.7521**	**0.7428**	**0.7426**	**0.7504**	**0.8346**
doc2vec	0.6576	0.6462	0.6503	0.6597	0.6754	0.7869
whatizit-dictionary	0.5230	0.5334	0.5517	0.5749	0.6020	0.7564
hybrid-doc2vec	0.6551	0.6463	0.6502	0.6608	0.6762	0.7872
scibert-scivocab-cased	0.6464	0.6290	0.6320	0.6407	0.6553	0.7714

We found that the use of only one controlled vocabulary, MeSH, may have contributed to the low scores observed in our evaluation. This could be due to the limited number of MeSH terms compared to the vocabulary that emerges from our dataset, or the inefficiency of replacing terms recognized by MeSH into normalized text. We will investigate this further. Moreover, the precision scores for all of our approaches were not particularly high, indicating that none of them naturally captures the relevance. The most promising approach in this regard was word2doc2vec.

Table 5
precision (P@N)

Approach	P@5	P@10	P@15	P@20	P@25	P@50
word2doc2vec	**0.6574**	**0.5821**	**0.5346**	**0.4975**	**0.4711**	**0.3817**
doc2vec	0.5347	0.4938	0.4686	0.4475	0.4329	0.3778
whatizit-dictionary	0.3952	0.391	0.3913	0.3931	0.3965	0.3878
hybrid-doc2vec	0.5306	0.4937	0.4684	0.4496	0.4342	0.3781
scibert-scivocab-cased	0.5167	0.4717	0.4494	0.4306	0.4169	0.3716

Overall, our results suggest that there is no definite winning approach in this study based on the very close AUC scores using the distribution-based analysis and extremely low precision scores. All the evaluation metrics that were used are tailored to a different task at hand. AUC using the distribution-based analysis gives us an idea about the coverage of relevance judgment. In our study, it was tailored to two relevant categories, and converted into a binary classification problem; in a future run, we will turn it into a multi-class classification so we can take into account the three relevance classes (definitely relevant, partially relevant, non-relevant) rather than merging definitely with partially relevant (which is a common practice when working with topic-to-document relevance).

Although the hybrid-doc2vec approach was promising, it did not perform well. Our expectation was that using NER before the embeddings would improve results; however, this was not the case. A possible reason is the even poorer performance from the dictionary-based approach that was also used in the hybrid-doc2vec approach. As for the direct use of similarity to assess relevance, our results suggest that this is not possible, at least not for the RELISH corpus. None of the approaches showed high values for precision (they are close to 50%, except for word2doc2vec on P@5 with 0.6574), and none were high enough for a search engine or recommendation system with respect to the nDCG except for the word2doc2vec approach which exhibited the highest nDCG score of 0.78 for nDCG@5.

4. Conclusions and Future Work

Our initial experiments (our project is still a work in progress) aimed to generate vector representations for the particular corpus at hand, with the exception of the BERT-based approach which utilized pre-trained models. Different hyperparameter configurations were assessed using the AUC to obtain the best hyperparameter combination for each approach. Our evaluation based on nDCG@n and precision@n revealed that the cosine similarity can serve as an effective re-ranking mechanism for a known resultset. Although the hybrid-doc2vec and BERT-based approaches were initially expected to perform better, the low nDCG for the hybrid-doc2vec approach could be attributed to the NER process before embeddings, which exhibited the lowest nDCG when used alone.

Our comparison allows us to find a similarity-based approach that exhibits a natural resemblance to relevance without requiring additional training. However, it is important to note

that relevance and similarity are related concepts but not interchangeable, thus further fine-tuning is necessary to optimize for relevance. Our future work will explore this subject as well as further variations of the word2doc2vec and combinations with NER approaches, using multi-classification for the distribution-based analysis used to optimize the hyperparameters (rather than reducing three assessments to two). We also want to try other options for the hybrid approach, such as revising the dictionary, using a different dictionary-based annotator, or doing the replacement after getting the vector space. Additionally, we will use a classifier approach to find a sound approach for relevance. Our ultimate goal is to provide a semantic-based approach that can assist researchers to find relevant literature not only in the well-covered biomedical domain, particularly with respect to Medline abstracts), but also in the agricultural domain, where there is a need for better coverage of non-traditional and non-peer-reviewed publications.

Acknowledgements

This work was partially supported by the STELLA project funded by the Deutsche Forschungsgemeinschaft DFG (project no. 407518790), the NFDI4DataScience project also funded by DFG (project no. 460234259), and the BMBF-funded de.NBI Cloud within the German Network for Bioinformatics Infrastructure (de.NBI) (031A532B, 031A533A, 031A533B, 031A534A, 031A535A, 031A537A, 031A537B, 031A537C, 031A537D, 031A538A).

References

[1] Zhu J, Patra BG, Yaseen A. Recommender system of scholarly papers using public datasets. AMIA Jt Summits Transl Sci Proc. 2021 May 17;2021:672-679. PMID: 34457183; PMCID: PMC8378599.

[2] Achakulvisut T, Acuna DE, Ruangrong T, Kording K. Science Concierge: A Fast Content-Based Recommendation System for Scientific Publications. PLoS One. 2016 Jul 6;11(7):e0158423. doi: 10.1371/journal.pone.0158423. PMID: 27383424; PMCID: PMC4934767.

[3] Bollacker KD, Lawrence S, Giles CL. CiteSeer: An autonomous web agent for automatic retrieval and identification of interesting publications. Proceedings of the second international conference on Autonomous agents. 1998 May 1. pp. 116–123.

[4] Yoneya T, Mamitsuka H. PURE: a PubMed article recommendation system based on content-based filtering. Genome Inform. 2007;18:267-76. PMID: 18546494.

[5] Lin J, Wilbur WJ. PubMed related articles: a probabilistic topic-based model for content similarity. BMC Bioinformatics. 2007 Oct 30;8:423. doi: 10.1186/1471-2105-8-423. PMID: 17971238; PMCID: PMC2212667.

[6] Zhu J, Patra BG, Yaseen A. Recommender system of scholarly papers using public datasets. AMIA JT Summits Transl Sci Proc. 2021 May 17;2021:672-679. PMID: 34457183; PMCID: PMC8378599.

[7] Brown P; RELISH Consortium, Zhou Y. Large expert-curated database for benchmarking document similarity detection in biomedical literature search. Database (Oxford). 2019 Jan 1;2019:baz085. doi: 10.1093/database/baz085. PMID: 33326193; PMCID: PMC7291946.

[8] National Center for Biotechnology Information at the National Library of Medicine part of the National Institutes of Health. https://pubmed.ncbi.nlm.nih.gov/ [last accessed 25 November 2022]

[9] Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv preprint arXiv. 2013 Jan 16;1301.3781.

[10] Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems. 2013:3111–3119.

[11] Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv. 2018 Oct 11;1810.04805
[12] Brown, Peter (2019): RELISH_v1. figshare. Dataset. https://doi.org/10.6084/m9.figshare.7722905.v1
[13] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. Valletta, Malta: ELRA, May 2010, p. 45–50. https://radimrehurek.com/gensim/models/word2vec.html#module-gensim.models.word2vec, [last accessed 11 November 2022].
[14] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. Valletta, Malta: ELRA, May 2010, p. 45–50. https://radimrehurek.com/gensim/models/doc2vec.html, last accessed 11 November 2022].
[15] Dietrich Rebholz-Schuhmann, Miguel Arregui, Sylvain Gaudan, Harald Kirsch, Antonio Jimeno. Text processing through Web services: calling Whatizit, Bioinformatics, Volume 24, Issue 2, 15 January 2008, Pages 296–298, https://doi.org/10.1093/bioinformatics/btm557.
[16] Benjamin Wolff, Leyla Jael Castro, Dietrich Rebholz-Schuhmann. Docker version for a minimal Whatizit. Available at https://github.com/zbmed-semtec/simple-whatizit-docker.
[17] Lipscomb CE. Medical Subject Headings (MeSH). Bull Med Libr Assoc. 2000 Jul;88(3):265-6. PMID: 10928714; PMCID: PMC35238.
[18] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., … Brew, J. (2019). HuggingFace's Transformers: State-of-the-art Natural Language Processing. CoRR, abs/1910.03771. http://arxiv.org/abs/1910.03771
[19] Reimers, N., & Gurevych, I. (11 2020). Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing. https://arxiv.org/abs/2004.09813
[20] Lee J, Yoon W, Kim S, Kim D, Kim S, So CH, Kang J. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics. 2020 Feb 15;36(4):1234-1240. doi: 10.1093/bioinformatics/btz682. PMID: 31501885; PMCID: PMC7703786.
[21] Beltagy I, Lo K, Cohan A. SciBERT: A pretrained language model for scientific text. arXiv preprint arXiv:1903.10676. 2019 Mar 26
[22] Data Mining and Information Systems Lab at Korea University. https://huggingface.co/dmis-lab [last access 25 November 2022]
[23] Data Mining and Information Systems Lab at Korea University. https://huggingface.co/dmis-lab/biobert-base-cased-v1.1 [last accessed 25 November 2022].
[24] Data Mining and Information Systems Lab at Korea University. https://huggingface.co/dmis-lab/biobert-large-cased-v1.1 [last accessed 25 November 2022].
[25] Semantic scholar | AI-Powered Research Tool [Internet]. Semanticscholar.org 2020. [cited 18 August 2020]. Available from: https://www.semanticscholar.org/
[26] Beltagy, I., Lo, K., & Cohan, A. (2019). SciBERT: A Pretrained Language Model for Scientific Text. EMNLP. https://www.aclweb.org/anthology/D19-1371