Scholoarly Research Recommender System: A Proposal

Shiquan He
WPI, PhD-DS Student

Daniel Fox
WPI, MS-DS Candidate

Cameron Morreale
WPI, BS-DS Candidate

April 19, 2023

Abstract

Literature search and research are important tasks for researchers. A good paper recommender system that quickly provides relevant research work is of great importance for both researchers and service providers. Currently, the major literature search engines like Google Scholar still use keyword searches for ranking retrieved papers. This type of keyword searching is fast and efficient, but it loses much of the contextual information during searches. In this research work, we propose to use contextualized document embeddings for paper retrieval and building paper recommender systems. We first introduce the background of this topic, the current state-of-art method and relevant work. Then we describe the document embedding methods, including the classic way like term-document matrix with TF-IDF weighting, doc2vec and our preferred contextualized document embedding method using SBERT and explain how to use them for a paper recommender system. We propose a solution for fast information retrieval. Next, we compare the performances of these methods for retrieving relevant papers through experiments and show that the contextualized document embedding method and TF-IDF methods exhibit better performance than TextRank. We summarize the comparison results and conclusions. We propose that using contextualized document embeddings is a good way to build a paper recommender systems. Finally, we discuss potential improvements and future work.

Keywords: paper recommender system, contextualized embedding, contextual embedding, document embedding, SBERT, document similarity, paper ranking

Introduction

We propose to use Natural Language Processing (NLP) methodologies to provide a solution for a paper recommender system. This target problem comes from a real world one. To the best of our knowledge, no one has attempted this before. We ourselves have scoured the Internet, especially using keyword searches at the Google Scholar site, to find any academic research paper that has proposed similar solutions. However, we cannot be 100% certain that there are no such papers, and this problem exists for all researchers, including graduate school researchers like ourselves, whether they come from academia or corporations. In addition, this task of literature searches can take up a considerable amount of time–time better spent on the research for the domain-specific problems we are endeavoring to solve. We feel very confident that improving the efficacy of searches and minimizing the time spent on such tasks will be welcomed not only by ourselves, but by countless academic researchers all over the world. We believe that there is monetary value in substantial improvements for this task. Researchers might part with some of their grant money to save the time and effort that they endure today. We assume that such a system can present relevant advertising to the end user. To justify the importance of our proposal, we can think about how important Google Scholar is to both academic researchers and to Google itself. If it is paramount for both of them, improvements over that service would be critical to implement and require our attention.

Google Scholar and other search engines using key word searching so far remain the current state-of-the-art. These engines index scholarly literature of all kinds. The advantages of these search engines, or information systems, are that they are fast and efficient and provide a plethora of resources to their end users. These engines tend to rank highly cited papers at the top of the search results and allow the end user to filter those results to a certain date range (by year). Both features are of great importance to the end user because many times the end user wants to read the most recent papers from influential experts in their domain of study. Furthermore, Google Scholar will provide the most cited authors in all research fields, which may help new researchers in case they are interested in reading highly influential research work. We can list many more advantages and usefulness to verify their success.

However, search engines like Google Scholar are not perfect and can make use of further improvement. One improvement would be a paper recommendation that does not rely solely on keyword information. Keyword information contains only highly summarized information of a paper and a majority of the contextual information of the content is lost during a literature search using simple keywords. This may explain why a researcher may spend a good amount of time on literature searches to find papers that are closely relevant to their research. Additionally, while we consider ranking papers based on the number of citations as a key feature, the problem with that is often lesser-known researchers, who may have come up with extraordinary research results, remain undiscovered in literature searches. And therefore some important academic papers may not draw the attention that they deserve. This is a critical point that should be taken seriously. In an ideal world, no great or innovative work should get ignored just because of an author's current fame. In fact, today's academic researchers may have placed much importance on Google Scholar or similar search engines, but the above listed issues are still to be investigated and resolved. Also, relying heavily on the number of citations will not yield very recent articles that may be extremely relevant and timely.

To further improve the current state-of-the-art in scientific publication search engines or information systems, we propose a novel solution that uses NLP methodologies to build a paper recommender system that uses the full text of papers. Any end user who wants to search for papers related to their research topic may simply upload a paper into the recommender system, which will then provide a ranked list of papers with similar topics or contents. The recommender would go beyond a search of a few keywords and provide a more comprehensive solution based on the entire text of relevant papers. Thus, with much more information introduced into the recommender system, it should provide a ranked list of papers that are closer in content to what appears in a Google Scholar list today. This, of course, would be a boon to all researchers, as an improvement in search efficiency would free them up for other activities. Furthermore, junior scholars would benefit greatly as their papers, which could be of significant importance, could be ranked amongst more influential researchers' papers. And scientific misconduct, especially plagiarism, could also be identified much more readily under such a system. Scholars can also use this system at an early stage of their research. As an example, they can upload their proposal or abstract into the system and check what other researchers have done. They can then contemplate what research they may either improve upon or outperform. Figure 1 conveys the processing of the recommender system, while figure 2 conveys the user work flow.

The task was to calculate document similarity and apply a series of methodologies for this application. The recommender system would look at the full text of papers but could also consider the year published, as well as citation information. As a baseline method, we constructed a term-document matrix.

We calculated term frequency-inverse document frequency (TFIDF) weighting and then computed the similarity among papers. We also employed some more advanced methods that used the contextual information in sentences and articles. Namely, we used document embedding that maintains or contains contextual information to calculate similarity among articles. Specifically, we applied a Bert-based method. Thus, for this proof of concept, we used the NIPS conference paper dataset, which contains the full text of all NeurIPS papers from 1987 through 2019, as our model development dataset.

We calculated document embeddings from different approaches and plugged that into the similarity calculation. We extracted from the NIPS dataset some example papers to retrieve relevant papers from the rest of this dataset to justify our idea.

Next, we explore related ranking algorithms that we want to improve upon in the related work section. A couple of examples that we explore include algorithms such as google scholar's PageRank, as well as the contextualized embedding approach. We then explore in depth the step-by-step process of how our ranking algorithm works in the methodology section, and explain how we quantified our results for our proof of concept. Since Google Scholar is still the current state-of-the art, we compared our results with theirs, and

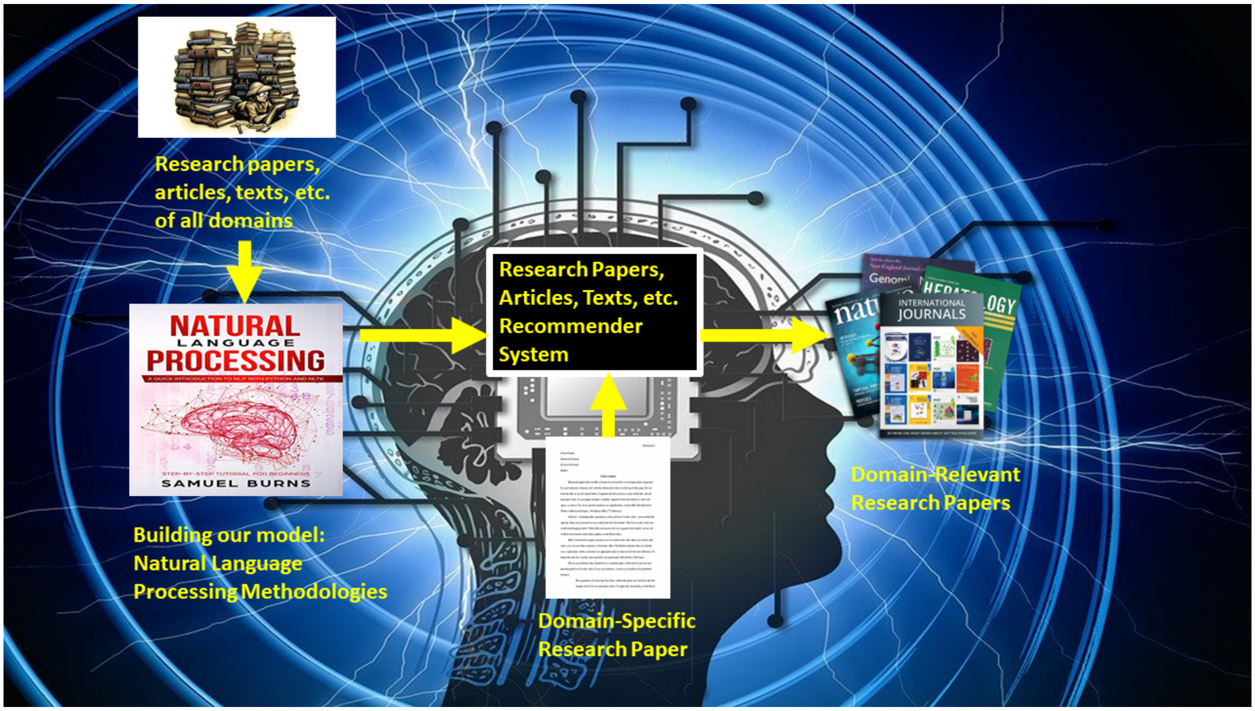
Research papers, articles, texts, etc. of all domains
NATURAL LANGUAGE PROCESSING
A QUICK INTRODUCTION TO NLP WITH PYTHON AND NLTK
STEP-BY-STEP TUTORIAL FOR BEGINNERS
SAMUEL BURNS
Building our model: Natural Language Processing Methodologies
Research Papers, Articles, Texts, etc. Recommender System
Domain-Specific Research Paper
INTERNATIONAL JOURNALS
Domain-Relevant Research Papers

Figure 1: Recommender System

explain in depth exactly how we did that in the project results and discussions section. We summarize other related work done on this topic, followed by the methodology we will use. We then discuss our results and share our conclusions based on those results. We finish with potential future work, followed by our references and an appendix showing our results in greater detail.

Related Work

According to Beel, et. al. [17], other disadvantages of Google Scholar's ranking algorithm are: it puts a high weighting on words in the title rather than the inner content; it does not consider synonyms for words in the document; it does not put any weighting on frequency of terms; it favors recent papers over old ones, which could be a net positive if it didn't also rely heavily on the number of citations; it also weighs certain authors and journals higher than other authors and journals, which also puts recent papers by inexperienced authors at a distinct disadvantage. Beel, et. al. [17] also proposes the number of citations as a criterium for measuring the success of a recommender system, but that kind of thinking is precisely what we are trying to avoid. We want to find the most similar papers, even if they are lost in obscurity. Yes, it's possible that eventually any excellent paper will receive many citations, but even if that's the case, it takes time for those citations to roll in and gain steam as the paper's Google Scholar PageRank improves. We do not want to miss great quality papers from lesser known authors.

There are researchers attempting to improve upon PageRank systems for scholarly papers. Al-Lahham, et al. [11] propose a formula that favors popular authors and penalizes age in Google Scholar's PageRank system:

$$SRP\_Score = \frac{PR}{1 + log_2(A_i)} + AU$$

$$A = Y - Y_i$$

$$AU = \frac{AW_1 + ... + AW_m + AC_1 + ... + AC_m}{NxH}$$

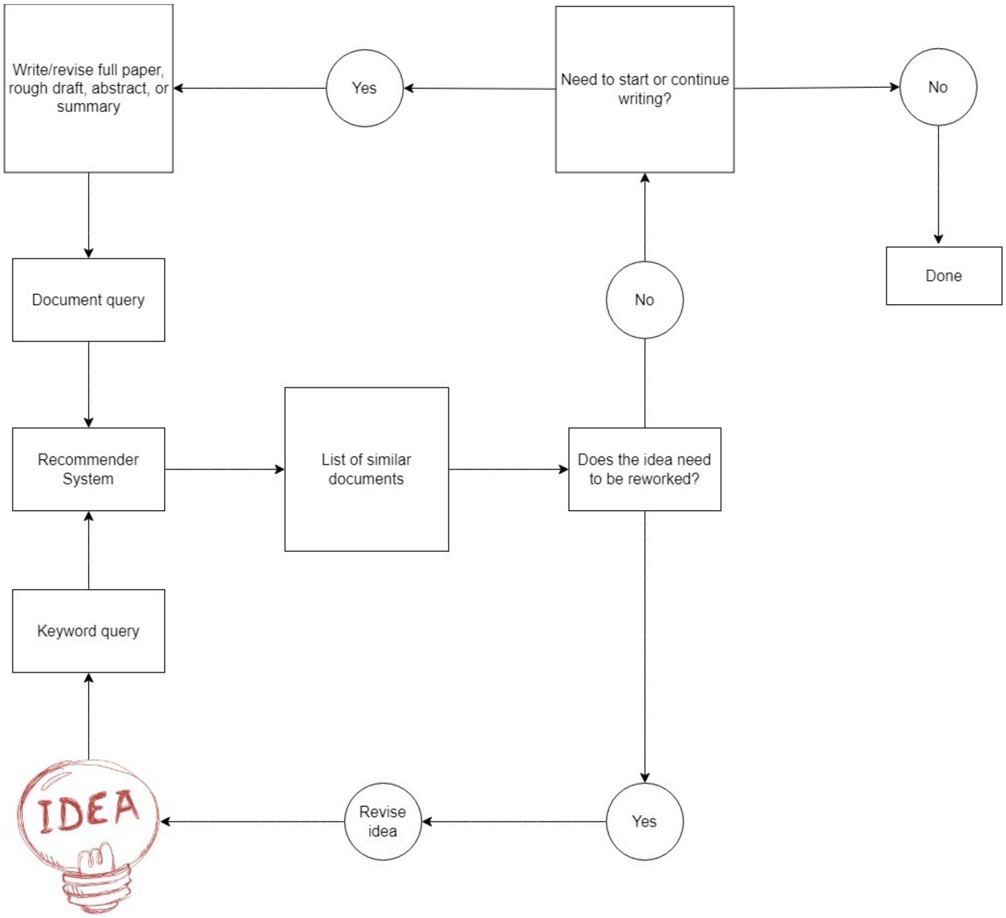
Write/revise full paper, rough draft, abstract, or summary
Yes
Need to start or continue writing?
No
Document query
No
Done
Recommender System
List of similar documents
Does the idea need to be reworked?
Keyword query
IDEA
Revise idea
Yes

Figure 2: Work Flow Diagram

Table 1: An Exmaple of Term-Document Matrix

Word	Paper 1	Paper 2	$\cdots$	Paper N
ai	90	65	$\cdots$	88
artificial	10	22	$\cdots$	17
intelligence	12	20	$\cdots$	17
machine	46	33	$\cdots$	46
learning	48	38	$\cdots$	42
deep	26	29	$\cdots$	19
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$

Here, SRP_Score is the score of the scientific research paper, A is the age of the paper in years, m denotes an author of the paper, AW_i is the number of papers published by the ith author, and AC_i is the citation count for the ith author.

However, favoring popular authors is something that we are trying to avoid. Instead, we want to favor content irrespective of how popular or unpopular the authors are.

MacAvaney et al. also proposed to use contextualized embedding for document ranking. However, their CEDR method uses neural ranking architectures which improve the ranking performance but with expensive computation costs, which is not practical for a large paper recommender system [9].

Methodology

As mentioned in the introduction section, we need to calculate the similarity among papers. Representing a scientific paper as a vector in a multi- or high dimensional semantic space that embeds the content of the paper is the essence. We explain here the methods used to measure similarity between documents, non-contextual embedding as well as contextualized methods.

1. Introduction of Methods

   (a) Measuring similarity
   By far, the most widely used similarity metric is cosine similarity, which is based on the inner product. Assume that we have two documents and their embeddings are $d_1 = (d_{11}, d_{12}, \ldots, d_{1m})^T$ and $d_2 = (d_{21}, d_{22}, \ldots, d_{2m})^T$ respectively, where m is the dimension of the document embedding. Then the cosine-similarity between these two documents is:

   $$cosine(d_1, d_2) = \frac{\vec{d_1} \cdot \vec{d_2}}{\|\vec{d_1}\|\|\vec{d_2}\|}$$

   As the formula shows, cosine similarity is a normalized version of the inner product between two vectors.

   (b) Term-document matrix
   Suppose that we have a collection of documents, and we have a vocabulary built out of the documents. A term-document matrix is like a table where each row represents a word in the vocabulary, and each column represents one document. Each cell in this matrix lists the number of times that a particular word occurs in a document. Table 1 shows a fake example of a term-document matrix. After constructing a term-document matrix, we can consider each row as a word embedding and each column as a document embedding. We can use the cosine-similarity to calculate word and document similarities. The general idea here is that similar words may appear in the same documents, and similar documents may share common words [18]. It is clear here that even though this matrix and these embeddings capture the word information, the contextual information is not preserved, which is a disadvantage of this method.

Table 2: An Exmaple of TF-IDF Weighted Term-Document Matrix

Word	Paper 1	Paper 2	$\cdots$	Paper N
ai	0	0	$\cdots$	0
artificial	0.045	0.061	$\cdots$	0.056
intelligence	0.049	0.060	$\cdots$	0.056
machine	0.037	0.034	$\cdots$	0.037
learning	0.037	0.035	$\cdots$	0.036
deep	0.100	0.103	$\cdots$	0.090
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$

(c) TF-IDF weighting

The term-document matrix uses the raw frequency of words. However, the raw count is not a good measure for assessing similarity or association between words or documents. Intuitively, a word happening 1000 times does not make it 1000 times more relevant to the content of a document [18]. For example, the words like 'there', 'are', 'is', 'it' and so on are very frequent in a document and across documents, but are not helpful in differentiating them. Term frequency and inverse document frequency (TF-IDF) was proposed to adjust the raw frequency of a word. Specifically, the TF-IDF is calculated using the following formula:

$$tf(t,d) = \begin{cases} 1 + \log_{10} count(t,d), \text{ if } count(t,d) > 0 \\ 0, \text{ otherwise} \end{cases}$$

, where $count(t,d)$ is the count of term t in document d.

$$idf_t = \log_{10}\left(\frac{N}{df_t}\right)$$

, where N is the total number of documents in the collection and df_t is the number of documents containing term t. The TF-IDF weighting of the term t in document d is

$$w_{t,d} = tf(t,d) \times idf_t$$

Table 2 provides a fake example of a TF-IDF weighted term-document matrix. In the assumed collection of documents, 'ai' appears in every paper (probably the collection are papers from AAAI where they are all AI related papers), so the weighting is 0, as it does not help us understand the content of each paper in that collection. Similarly, other values can be explained. After we have this TF-IDF weighted word-document matrix, we can calculate the word similarity or document similarity. A popular way to compute the document similarity is to represent a document by taking the vectors of all the words (after removing stop words, etc.) in the document and calculate the centroid of all the vectors [18]. Assume that there are p words in a document d_i, then we can represent it by its centroid as:

$$d_i = \frac{w_1 + w_2 + \cdots + w_p}{p} \text{ where } w_1,\ w_2,\ \cdots,\ w_p \text{ are the word vectors.}$$

(d) Contextual Word Embeddings

Contextual word embedding, as the name suggests, means representing a word with contextual information. The word embeddings from word2vec [15], GloVe, or the above mentioned embedding retrieved from a word-document matrix do not have contextual information, and each word learned from those methods has only one embedding. As an example, the word 'school' in 'middle school' and 'school of fishes' have the same embedding [1], which we may not want to see. Contextual embedding methods like BERT [1], ELMo [13] or GPT [14] provide the contextual embedding of each word. For this 'school' example, two embeddings would be provided.

BERT stands for bidirectional encoder representations from transformers, and was developed by researchers at Google as described in Devlin, et. al. [1]. Since publication in 2018, it became very popular in both academia and industry and has achieved state-of-the-art performance on many natural language understanding tasks. It is a language model pre-trained on an enormous source of text data, such as Wikipedia and Toronto BookCorpus, and it can be fine-tuned in downstream tasks. BERT is based on a transformer architecture, which means it is a sequence to sequence model with "self-attention". The original BERT was trained with 12 layers of transformer blocks with 12 multihead attentions. The model has about 110 million parameters. During the training, there were two primary tasks: masked word/token prediction and next sentence prediction. To have a good understanding of BERT, it is important to understand what are attention, self-attention, query, key, value and so on, but this is not a tutorial on BERT. We will not list all those formulas and idea behind it. Some modeling details like position encoding (as there is no word order information in the self-attention mechanism) are very interesting, but we will leave it to readers to read the original paper. The key takeaway for us is that contextualized embeddings are created through the self-attention mechanism or BERT. After we have the word embeddings from BERT, we can use them for downstream tasks like classification, clustering, etc. BERT also has its descendents like SciBert, BioBert, etc.

(e) Document Embedding

The above introduced the centroid way of representing a document is a document embedding, as we use that as a baseline method in this study. There are also other document embedding methods.

Doc2Vec extends word2vec, and it introduces a paragraph vector definition.

SentenceBERT provides a way to calculate embedding of a sentence [2] ('[cls]' token in BERT is also a sentence embedding). Using a similar technique as above, we can calculate the mean of SentenceBERT embeddings as the document embedding. One way to think about this mean embedding is that it has the minimum sum of squared distances to each of those word embeddings so that we can consider it as a representation of a document.

(f) ROUGE

The ROUGE metrics, or Recall-Oriented Understudy for Gisting Evaluation, are metrics intended to measure the closeness of a summary of a document to the full text of that document. We used ROUGE metrics to compare a retrieved document (using one of the methods above) to the source document. A higher score is better than a lower score.

We used two ROUGE metrics: ROUGE-1 and ROUGE-L.

ROUGE-1 measures the overlap of unigrams between the system document (the randomly selected source document in our case) and the reference summary (the retrieved document based on similarity).

ROUGE-L measures longest common subsequence (LCS) based statistics. This takes into account sentence-level structure similarity.

2. Proposed Solution

For a proof-of-concept, we used the NIPS dataset for our model development. This data contains 9,674 papers published in NeurIPS from 1987 to 2019. This data set includes the full text of the paper plus some metadata information like unique ID of each paper, publish year, title and abstract. From it, we can compute both non-contextual and contextual document embeddings.

We randomly selected five papers from the NIPS data set as our target papers, and our goal will be to find relevant papers from this NIPS data set. We selected these papers with no restriction, except that were included in the NIPS data set.

We calculated document embeddings from a TF-IDF weighted word-document matrix and the similarity between the target papers and each of the papers in the NIPS data set. This serves as the benchmark. We employed the aforementioned document embedding methods and BERT-based document embeddings to calculate similarities. Supposedly, they contain more complete information than the word count based method as they contain more contextual information.

TextRank is a graph-based ranking model for text and can be considered as a variation of Google's PageRank for text [12]. To imitate what Google Scholar does for paper ranking, we employed TextRank to extract top ranked papers from this collection of NIPs papers and compared them with the relevant papers we found. Specifically, TextRank extracts the top k (we used 30 in our experiment) keywords for each paper. Sentence similarity extends [12] to document similarity with the following formula:

$$similarity\left(d_i, d_j\right) = \frac{|\{w_k | w_k \in d_i \text{ and } w_k \in d_j\}|}{\log(|d_i|) + \log(|d_j|)}$$

, where d_i and d_j are two documents and w_k's are the common shared keywords. We considered that the most relevant papers has the highest similarities with a target paper or documents.

3. Potential Solution for Fast Information Retrieval
   The non-contextual embedding based methods do have their own advantages. For example, computation of the TF-IDF based document embeddings are relatively fast, and such embeddings can be stored with indexing. However, the computation for contextual embeddings can be heavier than that. Additionally, calculating the similarity with all papers in a search engine can be a heavy task, which we wanted to avoid. Here we provide some of our thinking on handling these issues. First, for each paper, we calculated document embedding only once, so it would not be an enormous task. Each paper's embedding would be stored in a file system, so that in real time, we only needed to extract that information from a file. There was no need to calculate the similarity between a target paper with all papers available in the search engine.

   Search engines can build clustering for different research fields. For example, let's assume that we build a cluster for AI and machine learning topic papers. Then, at runtime, a random paper in this cluster would be picked, and the similarity with the target paper would be measured. Based on that calculated similarity, a new jump would be performed and a new similarity can be evaluated. After a few jumps, the relevant papers can be found as the neighbors of the final picked paper. We can also imagine that the jump step would decay as steps continue. Of course, this paragraph is speculative and outside the scope of this project.

Experimental Results and Discussions

To verify the effectiveness of our proposed solution, we designed the following experiment.

Experiment

1. Randomly select n articles as our target documents
2. For each target document, retrieve the top ranked relevant papers using our proposed contextual document embedding method, non-contextual (term-document matrix with TF-IDF weighting)way and TextRank way.
3. For each target document, compare the papers retrieved from the above methods and evaluate which ones are more related to the target paper.
4. Summarize the finding from these n iterations of comparisons.

In the above experiment, we randomly selected 5 papers from the NIPS data set as our target papers and looked for the most relevant papers from the rest of NIPS data set. The details of the five chosen papers are summarized in table 3. For the above step 3, we used the ROUGE metrics [21] for evaluating the relevance between the target document and a candidate one. We need to note that ROUGE metrics may not be great candidates for this evaluation since they do not consider semantic information in the text. Synonyms are different words in this calculation, which actually does not favor contextualized embedding methodologies. We can also use some other numeric metrics (for example, use the document similarity introduced in the

Table 3: Randomly Selected Target Papers

Target Paper Index	Title	Author	Year	Citation
1	Computing with Finite and Infinite Networks	Winther	2000	5
2	Integrated Segmentation and Recognition of Hand-Printed Numerals	Keeler, Rumelhart, Leow	1990	211
3	Accelerated Training for Matrix-norm Regularization: A Boosting Approach	Zhang, Schuurmans, Yu	2012	107
4	Sodium entry efficiency during action potentials: A novel single-parameter family of Hodgkin-Huxley models	Singh, Jolivet, Magistretti, Weber	2010	3
5	Modeling Social Annotation Data with Content Relevance using a Topic Model	Iwata, Yamada,Ueda	2009	63

Table 4: Summary of Retrieved Papers from TF-IDF-Based Method

Most Relevant Paper	ROUGE-1 F_1 Top Paper	ROUGE-L F_1 Top Paper	ROUGE-1 F_1 Top 10 Mean	ROUGE-L F_1 Top 10 Mean
On-line Learning from Finite Training Sets in Nonlinear Networks	0.281	0.263	0.288	0.273
A Self-Organizing Integrated Segmentation and Recognition Neural Net	0.394	0.371	0.277	0.260
Convex Multi-view Subspace Learning	0.380	0.364	0.315	0.302
Energetically Optimal Action Potentials	0.275	0.255	0.241	0.224
A Discriminative Latent Model of Image Region and Object Tag Correspondence	0.266	0.251	0.246	0.234

method section defined only using word information) to measure the relevance. We cannot solely use the same k keywords again here as they have already been used when using the TextRank method and the ones returned from it already have the highest similarity scores. It is likely that this calculation will require the broadening the words. As ROUGE metrics may be more popular today, we employed it rather than the method we just mentioned here.

Results

For the non-contextualized embedding methods, we used TF-IDF and doc2vec methods. Tables 4 and 5 summarize the papers retrieved by TF-IDF and doc2vec, respectively. In these summaries, we only show the most relevant papers for each method and provide summary metrics for the top 10 similar papers. We provide the details of the retrieved papers in the appendix.

Google Scholar is still the SOTA, and many researchers and professionals rely on it nowadays, so we aimed to replicate Google Scholar's keyword search results so that we can compare our proposed method's results with theirs. However, Google Scholar does not support API calls for querying. So we cannot use this approach. Even if we perform a manual search on the Google Scholar site, many of the returned papers probably will not be from NIPS, which is a second issue. We need to check manually if there are any returned papers from NIPS or do some regular expression match after copy-paste the search results. If there are some returned papers from NIPS from the same year range as our model development data set, we can still compare them with the relevant papers we find. But this is cumbersome. Additionally, extracting keywords from a target paper objectively is problematic. It is hard to justify the precise methods for doing this and very likely time-consuming.

To overcome these issues and still imitate what Google Scholar search engine does, we used TextRank [12] for

Table 5: Summary of Retrieved Papers from doc2vec-Based Method

	ROUGE-1 F_1	ROUGE-L F_1	ROUGE-1 F_1	ROUGE-L F_1
Most Relevant Paper	Top Paper	Top Paper	Top 10 Mean	Top 10 Mean
Learning in large linear perceptrons and why the thermodynamic limit is relevant to the real world	0.267	0.260	0.279	0.264
A Self-Organizing Integrated Segmentation and Recognition Neural Net	0.394	0.371	0.290	0.268
Convex Multi-view Subspace Learning	0.380	0.364	0.315	0.302
Energetically Optimal Action Potentials	0.275	0.255	0.245	0.228
"Name That Song!" A Probabilistic Approach to Querying on Music and Text	0.239	0.225	0.258	0.243

Table 6: Summary of Retrieved Papers fromTextRank

	ROUGE-1 F_1	ROUGE-L F_1	ROUGE-1 F_1	ROUGE-L F_1
Most Relevant Paper	Top Paper	Top Paper	Top 10 Mean	Top 10 Mean
Gaussian Processes for Regression	0.280	0.269	0.278	0.265
Consonant Recognition by Modular Construction of Large Phonemic Time-Delay Neural Networks	0.240	0.223	0.233	0.215
PSDBoost: Matrix-Generation Linear Programming for Positive Semidefinite Matrices Learning	0.306	0.292	0.296	0.283
Temporally Asymmetric Hebbian Learning, Spike liming and Neural Response Variability	0.229	0.213	0.227	0.213
Automatic Annotation of Everyday Movements	0.232	0.217	0.219	0.207

this purpose. TextRank is a graph-based method for text processing and shares the same logic as Google's PageRank. It only relies on text information rather than citation or publish year information, and can perform text summarization tasks and extract keywords from paragraphs. Based on the keyword extraction capability of TextRank, we could objectively extract keywords from a target document or any documents. In this research, we fetched the top 20 keywords from each paper and used the document similarity mentioned in the Method section to calculate a paper's relevance to a target paper. Table 6 summarizes the retrieved 10 most relevant papers for the 5 target ones.

Comparing between Tables 6 and 7, in terms of ROUGE metrics, Bert-based contextualized embedding method outperformed TextRank in 4 out of 5 target papers (except the first target paper). For the first target paper, the most relevant paper retrieved by Bert-based method still showed a little higher ROUGE-1

Table 7: Summary of Retrieved Papers from SBERT-Based Contextulized Embedding

	ROUGE-1 F_1	ROUGE-L F_1	ROUGE-1 F_1	ROUGE-L F_1
Most Relevant Paper	Top Paper	Top Paper	Top 10 Mean	Top 10 Mean
Gaussian Process Regression with Mismatched Models	0.283	0.270	0.271	0.258
A Self-Organizing Integrated Segmentation and Recognition Neural Net	0.394	0.371	0.286	0.269
Convex Multi-view Subspace Learning	0.380	0.364	0.315	0.301
Energetically Optimal Action Potentials	0.275	0.255	0.239	0.222
Improving Topic Coherence with Regularized Topic Models	0.258	0.243	0.255	0.240

and ROUGE-L F_1 scores than the most relevant one from TextRank but the means of ROUGE metrics from top 10 relevant papers were lower.

Comparing contextualized document embedding with TF-IDF does not yield a universal winner. For the first target paper, contextualized embedding showed higher ROUGE metrics, but the mean metrics from the top 10 relevant papers were lower. For the second target paper, both methods retrieved the same most relevant paper, but the BERT-based method showed higher ROUGE metrics. For the third and fourth target papers, both contextualized and TF-IDF based methods retrieved the same most similar paper but TF-IDF based method showed a little higher mean ROUGE metrics (0.001 and 0.002). For the fifth target paper, the TF-IDF based method showed higher ROUGE values, but the mean ROUGE metrics were lower than the contextualized method. Similarly, the Doc2vec-based method does not show uniformly better performance than the BERT-based method. For the first target paper, the contextualized method showed better performance than doc2vec on the most relevant paper, but the mean performance from doc2vec was better. For the second, third and fourth target papers, they found the same most relevant papers. For 2nd target document, doc2vec showed better performance in mean ROUGE1 and the contextualized embedding based method showed slightly better, 0.001, ROUGE-L value. For the third target paper, doc2vec based retrieval showed 0.001 higher in terms of ROUGE-L. For the fourth target paper, doc2vec showed better mean ROUGE metrics. For the fifth target paper, the contextualized embedding method had better ROUGE values for the most relevant paper but had lower mean ROUGE values than doc2vec. In summary, we did not observe a winner among BERT-based method and non-contextualized methods in terms of ROUGE metrics, even though ROUGE in nature favors more TF-IDF type of methods.

Figures 3, 4, 5, and 6, illustrate the differences in ROUGE performance for each ranking method (TF-IDF, Doc2Vec, BERT, and TextRank, respectively) and for each calculation of the metric (i.e., ROUGE-1 of the most relevant paper, ROUGE-L of the most relevant paper, and an average of ROUGE-1 for the top ten relevant papers, and an average of ROUGE-L for the top ten relevant papers). As TextRank is our benchmark of comparison, it is important to point out that it performed poorly compared to the other three methods in all cases, except for the first random paper.

Another interesting thing is that our solution returned some less-cited papers. This is one advantage we considered as an important property. Outstanding research outcomes and topics can be immersed in tons of publications, and sometimes different authors can publish the same method across years. For example, the third most relevant paper returned from the first target paper only has six citations; the sixth most relevant paper returned from the first target paper only has 3 citations. The table in the appendix provides more information on this. This is sound evidence that our solution is a better one in this scenario, where junior researchers' work can be ranked high if the content is really relevant.

Discussions

We need to pay attention that ROUGE metrics may not be a good evaluation metric for this task, as ROUGE metrics are still word-based and therefore do not consider synonyms or contextualized information. Thus, it favors more raw-term based methods like TF-IDF rather than our solution. But still BERT-based contextualized embedding methods did not lose in this comparison. The authors still hold that contextualized embedding methods may incorporate more information from documents and are better types of methods. Paper recommender systems can benefit from contextualized embedding-based methods.

For the TextRank method, even though it works for texts and tries to imitate Google PageRank, it does not fully replicate how Google Scholar works, which also takes publication year and citations into consideration during ranking. Adding those two features can be applied, and some optimization on weights for publication year and citations is needed. However, we may not want to add too much weight to either of them. One reason is that some old work can also be very relevant. As an illustration, neural networks and deep learning have attracted a lot of attention in the past ten years. However, some neural network methods are not new, but have gained in popularity as stronger computer power has made these technologies more viable and deployable. In this case, if we want to do some literature search, we definitely do not want to miss any relevant methodologies or research that was previously done. Citations are another consideration. As mentioned earlier, research work from junior scholars may not attract the same attention as senior or famous scholars, but they can propose novel or even ground-breaking research. In this scenario, a paper recommender

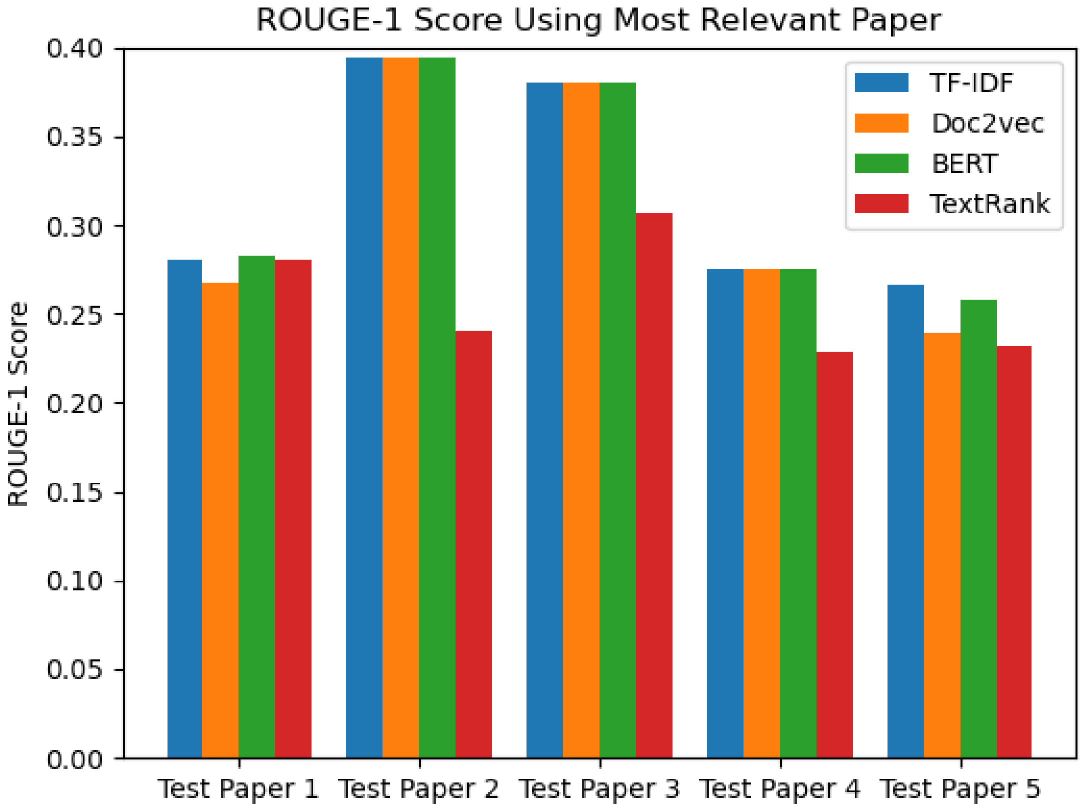

Figure 3: ROUGE-1 Score for Most Relevant Paper

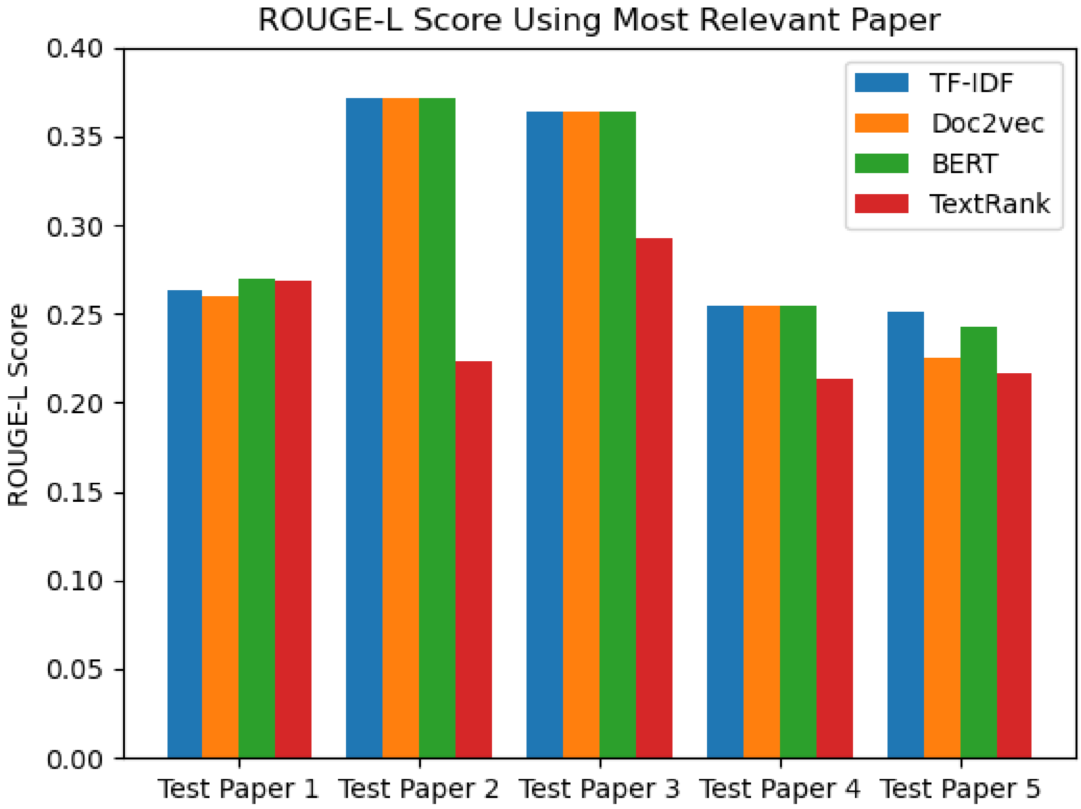

Figure 4: ROUGE-L Score for Most Relevant Paper

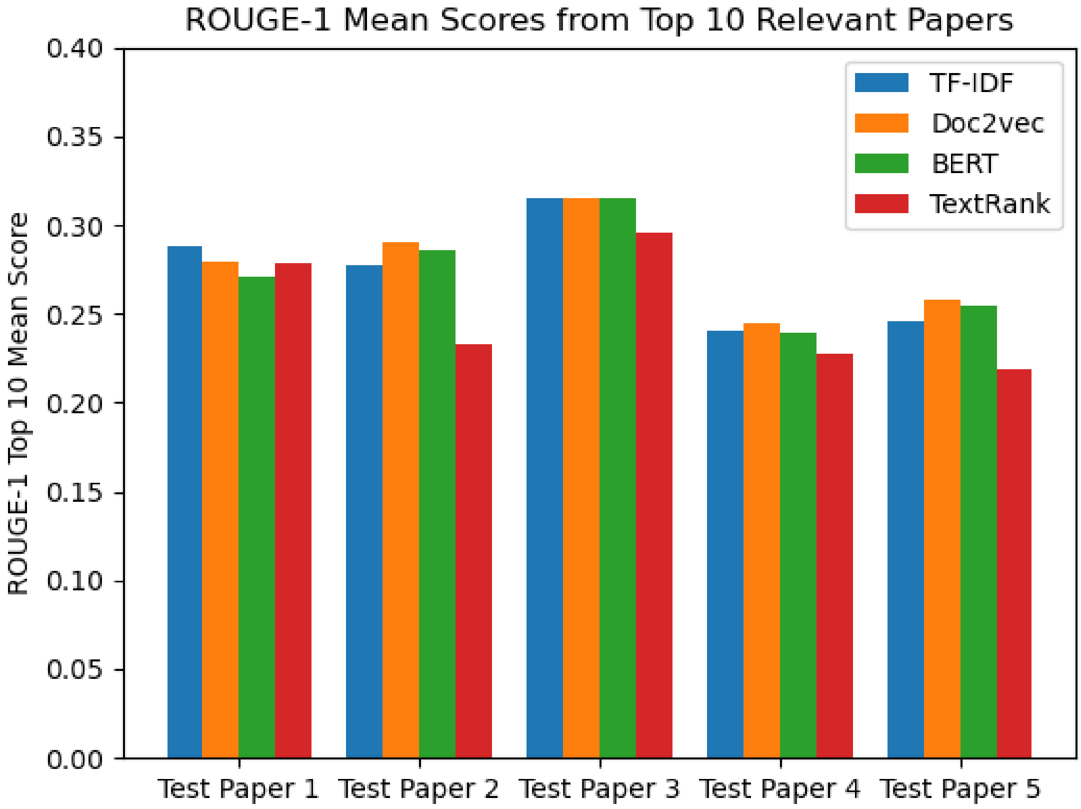

Figure 5: ROUGE-1 Mean Scores from Top 10 Relevant Papers

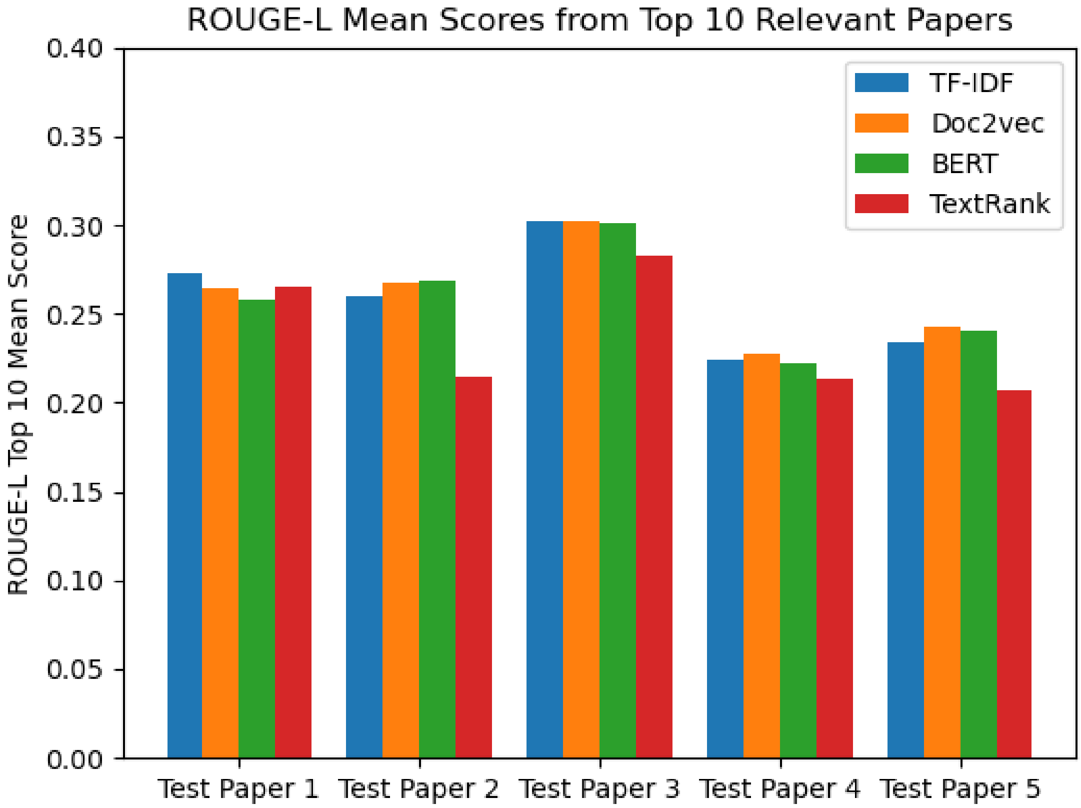

Figure 6: ROUGE-L Mean Scores from Top 10 Relevant Papers

system that recommends papers by focusing on content, rather than relying too much on citations, can be helpful. We also admit that complete removal of citation information is not optimal as sometimes researchers may want to read classic or recognized papers in a certain field rather than reading some low-cited papers. There is a tradeoff here, and only giving little or too much weight on them may be not the optimal solution.

Retrieval speed is an important feature of a recommender system, and we need to consider this during a recommender system development. We definitely do not want a system that responds too slowly. If a method provides a good ranking but with expensive computation costs, it is not practical for a large paper recommender system. However, this may change as computation becomes more and more powerful.

Conclusions and Future Work

We had set out evaluate the following models for such a paper recommendation system: Term-Document, TF-IDF, doc2vec, and BERT. We quickly realized that TF-IDF was just a superior form of Term-Document, so although we wrote and tested the code for Term-Document, we ultimately chose not to run it. Although the Gantt chart shows us running Google Scholar and comparing ranks, we also identified in our progress report using TextRank instead, as a stretch goal. That we were able to do, and we discovered that TextRank scored poorly compared to all the others. We also said we would identify the best model, but our results were mixed, and therefore could not identify one. We did, though, do enough work that we would have been able to identify a best model, if there was a clear winner. Ultimately, we used the ROUGE metric to judge success, and that was a work item not on the Gantt chart. Therefore, we feel we accomplished everything that we put on the Gantt chart in our progress report, and in the timeline schedule of our proposal, or at least its equivalent, and more.

It may take a significant amount of time for a researcher to execute a literature search and find relevant target papers that are of great importance for them. Thus, we can imagine how paramount a good paper recommender system is. Google Scholar and other search engines still use keyword searching, which has advantages like fast speed. However, they are not perfect and can use further improvement. A paper recommender system that uses full contextual information from documents rather than only keywords can comprise more of the actual content of papers and, accordingly, return more relevant papers during searches. This would be an influential improvement, considering the widespread use of literature search engines among scholars and professionals. Other advantages, such as the aforementioned attention to junior scholar's work, as well as automatic detection of scientific misconduct, are also meaningful. Last but not least, our proposed solution does not need a human to select the keywords as is required by Google Scholar. Thus, it is an important topic and work.

Future work can make further improvements. One of them is to find a suitable method that can objectively prove that contextualized embedding based methods are better than the keywords-based method. In this paper, we used ROUGE metrics, but it is not optimal for this type of task. Some human judgement can also be involved for future work. Another thing to mention is that our work is still a proof-of-concept. In the future, with further improvements, it could be productized, and that would be a fruitful outcome.

Another future research topic is acquiring better document embedding. In this paper, we use the arithmetic mean of SBERT embeddings as the document embedding, which can be improved through further research. A good document embedding can help not only this paper recommender system work but also broader applications of work. We believe that this is important future work and can imagine that more research work will come out of this topic.

References

[1] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[2] Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084.

[3] Dai, A. M., Olah, C., & Le, Q. V. (2015). Document embedding with paragraph vectors. arXiv preprint arXiv:1507.07998.

[4] Bianchi, F., Terragni, S., & Hovy, D. (2020). Pre-training is a hot topic: Contextualized document embeddings improve topic coherence. arXiv preprint arXiv:2004.03974.

[5] Le, Q., & Mikolov, T. (2014, June). Distributed representations of sentences and documents. In International conference on machine learning (pp. 1188-1196). PMLR.

[6] Teofili, T. (2019). Deep learning for search. Simon and Schuster.

[7] Baladevi, C., & Harikumar, S. (2018, August). Semantic representation of documents based on matrix decomposition. In 2018 International Conference on Data Science and Engineering (ICDSE) (pp. 1-6). IEEE.

[8] Mitra, B., & Craswell, N. (2018). An introduction to neural information retrieval. Foundations and Trends® in Information Retrieval, 13(1), 1-126.

[9] MacAvaney, S., Yates, A., Cohan, A., & Goharian, N. (2019, July). CEDR: Contextualized embeddings for document ranking. In Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval (pp. 1101-1104).

[10] Wu, L., Fisch, A., Chopra, S., Adams, K., Bordes, A., & Weston, J. (2018, April). Starspace: Embed all the things!. In Proceedings of the AAAI conference on artificial intelligence (Vol. 32, No. 1).

[11] Al-Lahham, Y., Al-Hatta, F., & Hassan, M. (2016) An Efficient Ranking Algorithm for Scientific Research Papers. In 2016 International Arab Conference on Information Technology.

[12] Mihalcea, R., & Tarau, P. (2004, July). Textrank: Bringing order into text. In Proceedings of the 2004 conference on empirical methods in natural language processing (pp. 404-411).

[13] Matthew, P., Mark, N., Mohit, I., Matt, G., Christopher, C., Kenton, L., & Luke, Z. (2018). Deep contextualized word representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). Association for Computational Linguistics.

[14] Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.

[15] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

[16] Zhang, Y., Wang, M., Saberi, M. et al. Analysing academic paper ranking algorithms using test data and benchmarks: an investigation. Scientometrics 127, 4045–4074 (2022). https://doi.org/10.1007/s11192-022-04429-z

[17] Beel, J., Gipp, B., Langer, S. et al. Research-paper recommender systems: a literature survey. Int J Digit Libr 17, 305–338 (2016). https://doi.org/10.1007/s00799-015-0156-0

[18] Jurafsky, D., & Martin, J. H. (2008). Speech and Language Processing: An introduction to speech recognition, computational linguistics and natural language processing. Upper Saddle River, NJ: Prentice Hall.

[19] Bert implementation from Huggingface https://huggingface.co/docs/transformers/model_doc/bert

[20] Doc2vec implementation https://radimrehurek.com/gensim/models/doc2vec.html

[21] Lin, C. Y. (2004, July). Rouge: A package for automatic evaluation of summaries. In Text summarization branches out (pp. 74-81).

Appendix

Table 8: Top 10 Retrieved Papers from BERT-Based Method for the First Target Paper

Retrieved Paper	ROUGE-1 F_1	ROUGE-L F_1	Citation
Gaussian Process Regression with Mismatched Models	0.283	0.270	39
On-line Learning of Dichotomies	0.279	0.257	20
Understanding Stepwise Generalization of Support Vector Machines: a Toy Model	0.290	0.279	6
Gaussian Processes for Regression	0.280	0.269	1560
Dynamics of Training	0.277	0.268	missing
On-line Learning from Finite Training Sets in Nonlinear Networks	0.281	0.263	3
Learning Curves for Gaussian Processes	0.270	0.259	64
Discovering Hidden Features with Gaussian Processes Regression	0.266	0.256	48
Dynamics of Supervised Learning with Restricted Training Sets	0.266	0.256	3
The committee machine: Computational to statistical gaps in learning a two-layers neural network	0.223	0.204	79

Table 9: Top 10 Retrieved Papers from BERT-Based Method for the Second Target Paper

Retrieved Paper	ROUGE-1 F_1	ROUGE-L F_1	Citation
A Self-Organizing Integrated Segmentation and Recognition Neural Net	0.394	0.371	81
Recognizing Overlapping Hand-Printed Characters by Centered-Object Integrated	0.293	0.277	56
Handwritten Digit Recognition with a Back-Propagation Network	0.299	0.273	5529
Globally Trained Handwritten Word Recognizer using Spatial Representation, Convolutional Neural Networks	0.276	0.258	159
Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks	0.259	0.240	1321
Multi-Digit Recognition Using a Space Displacement Neural Network	0.276	0.265	229
Learning to See Where and What: Training a Net to Make Saccades and Recognize Handwritten Characters	0.281	0.269	14
A Large-Scale Neural Network Which Recognizes Handwritten Kanji Characters	0.244	0.231	35
Recognition-based Segmentation of On-Line Hand-printed Words	0.279	0.261	41
Digital Realisation of Self-Organising Maps	0.262	0.242	20

Table 10: Top 10 Retrieved Papers from BERT-Based Method for the Third Target Paper

Retrieved Paper	ROUGE-1 F_1	ROUGE-L F_1	Citation
Convex Multi-view Subspace Learning	0.380	0.364	178
Multi-output Polynomial Networks and Factorization Machines	0.318	0.307	12
QUIC & DIRTY: A Quadratic Approximation Approach for Dirty Statistical Models	0.325	0.311	14
Scalable Robust Matrix Factorization with Nonconvex Loss	0.298	0.285	9
PSDBoost: Matrix-Generation Linear Programming for Positive Semidefinite Matrices Learning	0.306	0.292	25
Efficient Learning using Forward-Backward Splitting	0.318	0.304	127
Efficient Optimization for Discriminative Latent Class Models	0.312	0.297	21
A Convergent Gradient Descent Algorithm for Rank Minimization and Semidefinite Programming from...	0.303	0.291	194
Proximal Quasi-Newton for Computationally Intensive L1 -regularized M-estimators	0.296	0.282	27
Efficient Recovery of Jointly Sparse Vectors	0.292	0.276	70

Table 11: Top 10 Retrieved Papers from BERT-Based Method for the Fourth Target Paper

Retrieved Paper	ROUGE-1 F_1	ROUGE-L F_1	Citation
Energetically Optimal Action Potentials	0.275	0.255	12
A Hodgkin-Huxley Type Neuron Model That Learns Slow Non-Spike Oscillation	0.241	0.224	21
Analytical Solution of Spike-timing Dependent Plasticity Based on Synaptic Biophysics	0.238	0.224	19
Simulations Suggest Information Processing Roles for the Diverse Currents in Hippocampal Neurons	0.242	0.221	6
Inferring synaptic conductances from spike trains with a biophysically inspired point process model	0.243	0.224	18
A Systematic Study of the Input/Output Properties of a 2 Compartment Model Neuron With Active Membranes	0.224	0.212	2
3 state neurons for contextual processing	0.240	0.221	3
Nonlinear Pattern Separation in Single Hippocampal Neurons with Active Dendritic Membrane	0.208	0.196	52
A VLSI Implementation of the Adaptive Exponential Integrate-and-Fire Neuron Model	0.259	0.238	86
Hybrid Circuits of Interacting Computer Model and Biological Neurons	0.219	0.204	23

Table 12: Top 10 Retrieved Papers from BERT-Based Method for the Fifth Target Paper

Retrieved Paper	ROUGE-1 F_1	ROUGE-L F_1	Citation
Improving Topic Coherence with Regularized Topic Models	0.258	0.243	236
Reading Tea Leaves: How Humans Interpret Topic Models	0.266	0.252	2814
Discriminative Topic Modeling with Logistic LDA	0.287	0.271	17
Modeling General and Specific Aspects of Documents with a Probabilistic Topic Model	0.250	0.226	266
The Missing Link - A Probabilistic Model of Document Content and Hypertext Connectivity	0.243	0.228	635
Learning a Concept Hierarchy from Multi-labeled Documents	0.237	0.223	21
Latent Dirichlet Allocation	0.240	0.226	756
The Doubly Correlated Nonparametric Topic Model	0.255	0.241	31
Syntactic Topic Models	0.269	0.253	12
Word Features for Latent Dirichlet Allocation	0.249	0.235	128

Table 13: Top 10 Retrieved Papers from TF-IDF-Based Method for the First Target Paper

Retrieved Paper	ROUGE-1 F_1	ROUGE-L F_1
On-line Learning from Finite Training Sets in Nonlinear Networks	0.281	0.263
Computing with Infinite Networks	0.300	0.280
Dynamics of stochastic gradient descent for two-layer neural networks in the teacher-student setup	0.252	0.239
Gaussian Process Regression with Mismatched Models	0.283	0.270
A Mean Field Algorithm for Bayes Learning in Large Feed-forward Neural Networks	0.313	0.296
Globally Optimal On-line Learning Rules	0.297	0.279
Mean Field Methods for Classification with Gaussian Processes	0.327	0.315
A Realizable Learning Task which Exhibits Overfitting	0.283	0.272
General Bounds on Bayes Errors for Regression with Gaussian Processes	0.318	0.299
Global Optimisation of Neural Network Models via Sequential Sampling	0.223	0.216

Table 14: Top 10 Retrieved Papers from TF-IDF-Based Method for the Second Target Paper

Retrieved Paper	ROUGE-1 F_1	ROUGE-L F_1
A Self-Organizing Integrated Segmentation and Recognition Neural Net	0.394	0.371
Learning to See Where and What: Training a Net to Make Saccades and Recognize Handwritten Characters	0.281	0.269
Recognizing Overlapping Hand-Printed Characters by Centered-Object Integrated Segmentation and Recognition	0.293	0.277
KODAK lMAGELINK™ OCR Alphanumeric Handprint Module	0.281	0.264
Globally Trained Handwritten Word Recognizer using Spatial Representation, Convolutional Neural Networks, and Hidden Markov Models	0.276	0.258
A Large-Scale Neural Network Which Recognizes Handwritten Kanji Characters	0.244	0.231
Human Reading and the Curse of Dimensionality	0.238	0.221
Neural Networks that Learn to Discriminate Similar Kanji Characters	0.267	0.245
Searching for Character Models	0.253	0.235
Effective Training of a Neural Network Character Classifier for Word Recognition	0.246	0.225

Table 15: Top 10 Retrieved Papers from TF-IDF-Based Method for the Third3 Target Paper

Retrieved Paper	ROUGE-1 F_1	ROUGE-L F_1
On the Linear Convergence of the Proximal Gradient Method for Trace Norm Regularization	0.303	0.293
Tight convex relaxations for sparse matrix factorization	0.332	0.31
Convex Multi-view Subspace Learning	0.38	0.364
Maximum-Margin Matrix Factorization	0.287	0.272
Practical Large-Scale Optimization for Max-norm Regularization	0.315	0.302
Decentralized sketching of low rank matrices	0.315	0.298
Collaborative Filtering in a Non-Uniform World: Learning with the Weighted Trace Norm	0.306	0.29
Online Optimization for Max-Norm Regularization	0.311	0.299
Efficient Structured Matrix Rank Minimization	0.34	0.324
PSDBoost: Matrix-Generation Linear Programming for Positive Semidefinite Matrices Learning	0.306	0.292

Table 16: Top 10 Retrieved Papers from TF-IDF-Based Method for the Fourth Target Paper

Retrieved Paper	ROUGE-1 F_1	ROUGE-L F_1
Energetically Optimal Action Potentials	0.275	0.255
A Hodgkin-Huxley Type Neuron Model That Learns Slow Non-Spike Oscillation	0.241	0.224
3 state neurons for contextual processing	0.24	0.221
Estimating time-varying input signals and ion channel states from a single voltage trace of a neuron	0.227	0.208
Channel Noise in Excitable Neural Membranes	0.242	0.225
Information Capacity and Robustness of Stochastic Neuron Models	0.253	0.239
Active dendrites: adaptation to spike-based communication	0.253	0.234
Hybrid Circuits of Interacting Computer Model and Biological Neurons	0.219	0.204
Dynamic Modulation of Neurons and Networks	0.198	0.186
Large-scale biophysical parameter estimation in single neurons via constrained linear regression	0.26	0.242

Table 17: Top 10 Retrieved Papers from TF-IDF-Based Method for the Fifth Target Paper

Retrieved Paper	ROUGE-1 F_1	ROUGE-L F_1
A Discriminative Latent Model of Image Region and Object Tag Correspondence	0.266	0.251
A Model for Learning the Semantics of Pictures	0.264	0.253
Computing with Infinite Networks	0.300	0.280
Automatic Annotation of Everyday Movements	0.232	0.217
Semi-crowdsourced Clustering with Deep Generative Models	0.234	0.222
Latent Dirichlet Allocation	0.240	0.226
Reading Tea Leaves: How Humans Interpret Topic Models	0.266	0.252
A Bayesian Model for Simultaneous Image Clustering, Annotation and Object Segmentation	0.266	0.255
On some provably correct cases of variational inference for topic models	0.228	0.222
A provable SVD-based algorithm for learning topics in dominant admixture corpus	0.223	0.208
DiscLDA: Discriminative Learning for Dimensionality Reduction and Classification	0.246	0.233

Table 18: Top 10 Retrieved Papers from doc2vec-Based Method for the First Target Paper

Retrieved Paper	ROUGE-1 F_1	ROUGE-L F_1
Learning in large linear perceptrons and why the thermodynamic limit is relevant to the real world	0.267	0.26
Dynamics of Training	0.277	0.268
Gaussian Process Regression with Mismatched Models	0.283	0.27
Learning Curves for Gaussian Processes	0.271	0.259
Dynamics of Supervised Learning with Restricted Training Sets and Noisy Teachers	0.254	0.236
Computing with Infinite Networks	0.3	0.28
On-line Learning of Dichotomies	0.279	0.257
A Mean Field Algorithm for Bayes Learning in Large Feed-forward Neural Networks	0.313	0.296
Discovering Hidden Features with Gaussian Processes Regression	0.266	0.256
On-line Learning from Finite Training Sets in Nonlinear Networks	0.281	0.263

Table 19: Top 10 Retrieved Papers from doc2vec-Based Method for the Second Target Paper

Retrieved Paper	ROUGE-1 F_1	ROUGE-L F_1
A Self-Organizing Integrated Segmentation and Recognition Neural Net	0.394	0.371
Recognizing Overlapping Hand-Printed Characters by Centered-Object Integrated Segmentation and Recognition	0.293	0.277
Handwritten Digit Recognition with a Back-Propagation Network	0.299	0.273
KODAK lMAGELINK™ OCR Alphanumeric Handprint Module	0.281	0.264
Boosting the Performance of RBF Networks with Dynamic Decay Adjustment	0.25	0.228
Dimensionality Reduction and Prior Knowledge in E-Set Recognition	0.292	0.26
Neural Networks that Learn to Discriminate Similar Kanji Characters	0.267	0.245
Training Stochastic Model Recognition Algorithms as Networks can Lead to Maximum Mutual Information Estimation of Parameters	0.238	0.216
Learning to categorize objects using temporal coherence	0.3	0.276
Learning to See Where and What: Training a Net to Make Saccades and Recognize Handwritten Characters	0.281	0.269

Table 20: Top 10 Retrieved Papers from doc2vec-Based Method for the Third Target Paper

Retrieved Paper	ROUGE-1 F_1	ROUGE-L F_1
Convex Multi-view Subspace Learning	0.38	0.364
Scalable nonconvex inexact proximal splitting	0.286	0.275
Dropping Symmetry for Fast Symmetric Nonnegative Matrix Factorization	0.312	0.303
Scalable Robust Matrix Factorization with Nonconvex Loss	0.298	0.285
ShareBoost: Efficient multiclass learning with feature sharing	0.331	0.319
QUIC & DIRTY: A Quadratic Approximation Approach for Dirty Statistical Models	0.325	0.311
Positive Semidefinite Metric Learning with Boosting	0.281	0.273
Faster Projection-free Convex Optimization over the Spectrahedron	0.312	0.295
CUR from a Sparse Optimization Viewpoint	0.307	0.291
A Dual Framework for Low-rank Tensor Completion	0.317	0.303

Table 21: Top 10 Retrieved Papers from doc2vec-Based Method for the Fourth Target Paper

Retrieved Paper	ROUGE-1 F_1	ROUGE-L F_1
Energetically Optimal Action Potentials	0.275	0.255
Dynamic Modulation of Neurons and Networks	0.198	0.186
A Summating, Exponentially-Decaying CMOS Synapse for Spiking Neural Systems	0.246	0.231
Simulations Suggest Information Processing Roles for the Diverse Currents in Hippocampal Neurons	0.242	0.221
Bifurcation Analysis of a Silicon Neuron	0.221	0.204
Dopamine Induced Bistability Enhances Signal Processing in Spiny Neurons	0.27	0.253
A VLSI Implementation of the Adaptive Exponential Integrate-and-Fire Neuron Model	0.259	0.238
Analytical Solution of Spike-timing Dependent Plasticity Based on Synaptic Biophysics	0.238	0.224
Optoelectronic Implementation of a FitzHugh-Nagumo Neural Model	0.237	0.221
Large-scale biophysical parameter estimation in single neurons via constrained linear regression	0.26	0.242

Table 22: Top 10 Retrieved Papers from doc2vec-Based Method for the Fifth Target Paper

Retrieved Paper	ROUGE-1 F_1	ROUGE-L F_1
"Name That Song!" A Probabilistic Approach to Querying on Music and Text	0.239	0.225
Modeling General and Specific Aspects of Documents with a Probabilistic Topic Model	0.25	0.226
Improving Topic Coherence with Regularized Topic Models	0.258	0.243
Hierarchically Supervised Latent Dirichlet Allocation	0.279	0.263
Learning a Concept Hierarchy from Multi-labeled Documents	0.237	0.223
A Model for Learning the Semantics of Pictures	0.264	0.253
Multi-view Anomaly Detection via Robust Probabilistic Latent Variable Models	0.286	0.271
Symmetric Correspondence Topic Models for Multilingual Text Analysis	0.265	0.251
Reading Tea Leaves: How Humans Interpret Topic Models	0.266	0.252
Sparse Additive Text Models with Low Rank Background	0.234	0.227

Table 23: Top 10 Retrieved Papers from TextRank-Based Method for the First Target Paper

Retrieved Paper	ROUGE-1 F_1	ROUGE-L F_1
Gaussian Processes for Regression	0.28	0.269
Learning Curves for Gaussian Processes	0.271	0.259
Training Methods for Adaptive Boosting of Neural Networks	0.23	0.217
General Bounds on Bayes Errors for Regression with Gaussian Processes	0.318	0.299
Mean Field Methods for Classification with Gaussian Processes	0.327	0.315
Regression with Input-dependent Noise: A Gaussian Process Treatment	0.294	0.281
Finite-Dimensional Approximation of Gaussian Processes	0.264	0.258
Microscopic Equations in Rough Energy Landscape for Neural Networks	0.255	0.243
Backpropagation Convergence Via Deterministic Nonmonotone Perturbed Minimization	0.245	0.232
Computing with Infinite Networks	0.3	0.28

Table 24: Top 10 Retrieved Papers from TextRank-Based Method for the Second Target Paper

Retrieved Paper	ROUGE-1 F_1	ROUGE-L F_1
Consonant Recognition by Modular Construction of Large Phonemic Time-Delay Neural Networks	0.24	0.223
Leaning by Combining Memorization and Gradient Descent	0.264	0.243
An Attractor Neural Network Model of Recall and Recognition	0.226	0.207
Sigma-Pi Learning: On Radial Basis Functions and Cortical Associative Learning	0.258	0.233
Speaker Independent Speech Recognition with Neural Networks and Speech Knowledge	0.257	0.238
The Effect of Catecholamines on Performance: From Unit to System Behavior	0.248	0.231
Tight Bounds for the VC-Dimension of Piecewise Polynomial Networks	0.207	0.194
Connecting to the Past	0.253	0.236
Neural Networks: The Early Days	0.168	0.152
Explorations with the Dynamic Wave Model	0.212	0.191

Table 25: Top 10 Retrieved Papers from TextRank-Based Method for the Third Target Paper

Retrieved Paper	ROUGE-1 F_1	ROUGE-L F_1
PSDBoost: Matrix-Generation Linear Programming for Positive Semidefinite Matrices Learning	0.306	0.292
Recovery of Sparse Probability Measures via Convex Programming	0.28	0.267
Nearest Neighbor based Greedy Coordinate Descent	0.27	0.256
Matrix reconstruction with the local max norm	0.297	0.28
Unified Methods for Exploiting Piecewise Linear Structure in Convex Optimization	0.29	0.278
Efficient Structured Matrix Rank Minimization	0.34	0.324
Structured Matrix Recovery via the Generalized Dantzig Selector	0.29	0.276
Stochastic Variance Reduction Methods for Saddle-Point Problems	0.277	0.267
New Insight into Hybrid Stochastic Gradient Descent: Beyond With-Replacement Sampling and Convexity	0.303	0.289
Quadratic Decomposable Submodular Function Minimization	0.311	0.297

Table 26: Top 10 Retrieved Papers from TextRank-Based Method for the Fourth Target Paper

Retrieved Paper	ROUGE-1 F_1	ROUGE-L F_1
Temporally Asymmetric Hebbian Learning, Spike liming and Neural Response Variability	0.229	0.213
Integrate-and-Fire models with adaptation are good enough	0.309	0.29
Energetically Optimal Action Potentials	0.275	0.255
Know Thy Neighbour: A Normative Theory of Synaptic Depression	0.249	0.231
A Model of Neural Oscillator for a Unified Submodule	0.211	0.194
A Spike Based Learning Neuron in Analog VLSI	0.226	0.217
The Effect of Eligibility Traces on Finding Optimal Memoryless Policies in Partially Observable Markov Decision Processes	0.173	0.161
Dynamic Modulation of Neurons and Networks	0.198	0.186
Homeostasis in a Silicon Integrate and Fire Neuron	0.213	0.207
The Asymptotic Convergence-Rate of Q-learning	0.183	0.17

Table 27: Top 10 Retrieved Papers from TextRank-Based Method for the Fifth Target Paper

Retrieved Paper	ROUGE-1 F_1	ROUGE-L F_1
Automatic Annotation of Everyday Movements	0.232	0.217
Deep Representations and Codes for Image Auto-Annotation	0.243	0.229
A Discriminative Latent Model of Image Region and Object Tag Correspondence	0.266	0.251
Semi-crowdsourced Clustering with Deep Generative Models	0.234	0.222
Bit-Serial Neural Networks	0.184	0.172
Barzilai-Borwein Step Size for Stochastic Gradient Descent	0.212	0.201
LazySVD: Even Faster SVD Decomposition Yet Without Agonizing Pain	0.182	0.172
A Probabilistic Framework for Deep Learning	0.221	0.21
Without-Replacement Sampling for Stochastic Gradient Methods	0.201	0.191
Learning to Communicate with Deep Multi-Agent Reinforcement Learning	0.215	0.208