11-23-2023

Enhancing Inter-Document Similarity Using Sub Max

Richard Imorobebh Igbiriki
Rowan University

ENHANCING INTER-DOCUMENT SIMILARITY USING SUB MAX

by

Richard Igbiriki

A Thesis

Submitted to the
Department of Computer Science
College of Science and Mathematics
In partial fulfillment of the requirement
For the degree of
Master of Science in Computer Science
at
Rowan University
July 10, 2023

Thesis Chair: Anthony Breitzman, Ph.D., Professor, Department of Computer Science

Committee Members:
Shen-Shyang Ho, Ph.D., Associate Professor, Department of Computer Science
Bo Sun, Ph.D., Associate Professor, Department of Computer Science

Dedications

It is said “education is the key to success” but not everyone can readily access it. Thus, I would like to dedicate this to everyone who desired a graduate degree but could not afford it.

Acknowledgments

I would like to show my appreciation to Dr. Anthony Breitzman, who believed in me, pushed me to work harder, and provided me with all the help I needed to succeed.

I would also like to thank my family and friends for supporting and believing in me every step of the way.

Abstract

Richard Igbiriki
ENHANCING INTER-DOCUMENT SIMILARITY USING SUB MAX
2023-2024
Anthony Breitzman, Ph.D.
Master of Science in Computer Science

Document similarity, a core theme in Information Retrieval (IR), is a machine learning (ML) task associated with natural language processing (NLP). It is a measure of the distance between two documents given a set of rules. For this thesis, two documents are similar if they are semantically alike, and describe similar concepts. While document similarity can be applied to multiple tasks, we focus our work on the accuracy of models in detecting referenced papers as similar documents using their sub max similarity. Multiple approaches have been used to determine the similarity of documents regarding literature reviews. Some of such approaches use the number of similar citations, the similarity between the body of text, and the figures present in those documents. This researcher hypothesized that documents with sections of high similarity (sub max), but a global low similarity are prone to being overlooked by existing models as the global score of the documents are used to measure similarity. In this study, we aim to detect, measure, and show the similarity of documents based on the maximum similarity of their subsections. The sub max of any two given documents is the subsections of those documents with the highest similarity. By comparing subsections of the documents in our corpus and using the sub max, we were able to improve the performance of some models by over 100%.

Table of Contents

Table of Contents (Continued)

Table of Contents (Continued)

List of Figures

List of Figures (Continued)

Figure Page

List of Tables

Chapter 1

Introduction

The overarching goal is to build a system to automate literature reviews. However, such a system is beyond the scope of a single thesis. This thesis is more a proof of concept where we wish to see if we can train a machine to automatically identify the core papers in an area of research. The experiment is given an arbitrary set of papers, can we find a method that would identify a high percentage of the papers that were ultimately referenced by these target papers.

To make this thesis self-contained, we will describe the basics of Information Retrieval and Document Similarity in the following sections so that the experiment can be better understood.

1.1 Information Retrieval

Information retrieval, as a field of study, is finding materials of an unstructured nature that satisfies an information need from within large collections (now usually stored on computers) (Manning et al, 2009). Unstructured text is usually the data type of focus for information retrieval tasks. Historically, IR was more associated with librarians, researchers, lawyers/paralegals, etc. However, with the rise of the internet, millions of people conduct IR when they use a search engine and search their emails and/or messages. Generally, the field also provides users with the ability to filter or further process a set of previously retrieved documents.

In 1945, Vannevar Bush published his article “As We May Think” which propelled the concept(s) of automatic access/retrieval of large amounts of stored information. In the article, he argues for man's need for a fast and reliable means of accessing existing information and the ability of extending such existing knowledge (Bush, 1945). This concept evolved into more detailed explanations of how text archives could be automatically searched in the 1950s. The fundamental concept of computerized text searching was expanded upon in several works that appeared in the middle of the 1950s. In 1957, H.P. Luhn introduced one of the most effective techniques, in which he advocated utilizing words as indexing units for documents and assessing word overlap as a criterion for retrieval (Luhn, 1957).

Information retrieval also extends to other tasks such as correctly grouping a given set of related documents (clustering), or accurately specifying what class a document belongs to (classification). While clustering of documents can be completed automatically, classifying documents requires some subset of the documents to be correctly classified (often manually). The classified documents are used as training data for the classification model to enable it to automatically classify future documents (Manning et al, 2009).

1.2 Document Similarity

Applications across numerous domains frequently must search for similar documents given a query document. A news website, for instance, could want to suggest articles related to the one the visitor is reading. The PubMed search engine which provides access to the life sciences literature, implemented a “more like this” browsing feature as a simple lookup of document-document similarity scores, computed offline (Elsayed et al, 2008). However, implementing such functionality requires (i) an effective way to find pertinent documents throughout potentially vast corpora, and (ii) a concept of document similarity (Paul et al, 2016). It's important to have a defined concept of similarity as it is integral to measuring the success or failure of any document similarity task.

In 2005, Lee et al argued that the automated measurement of the similarity between text documents is fundamentally a psychological modeling problem. Thus, the different approaches now in use, which are frequently applied in information science applications, should be evaluated (at least in part) in terms of their capacity to simulate human performance. (Lee et al, 2005). Humans with natural stimuli can accurately detect document similarity based on the semantics of given documents, thus any automated attempt should provide similar results. Numerous methods have been devised for modeling text document similarity. These consist of the more complex methods like Latent Semantic Analysis (LSA: Deerwester et al., 1990; Landauer and Dumais, 1997) as well as straightforward ones like word-based, keyword-based, and n-gram measurements (e.g., Salton, 1989; Damashek, 1995). For this research, we will cover five machine learning models used in measuring document similarity, namely: Bidirectional Encoder Representations from Transformers (BERT), Global Vectors for Word Representation (GloVe), Word2Vec, Term Frequency-Inverse Document Frequency (TF-IDF), and Doc2Vec.

1.3 Problem Statement and Proposed Solution

During literature review, researchers are required to read bodies of work that are related to their area(s) of interest to gain the requisite knowledge for conducting their own research. While this is a requirement for all scientific research, it is still a time-consuming task as researchers often must read through papers that may appear related but provide no additional information or value to the researcher. Having spent countless hours reading research papers as part of my literature review, we decided to find ways to improve the literature review experience by improving the quality/similarity of recommended literature given a particular piece of literature.

Multiple approaches have been used to determine the similarity of documents regarding literature reviews. Various approaches use the number of similar citations, the similarity between the body of text, and the figures present in those documents. The hypothesis we wish to test is whether documents with sections of high similarity, but a global low similarity are prone to being overlooked by existing models as the overall score of the documents are used to measure similarity. In this study, we aim to detect, measure, and show the similarity of documents based on the similarity of their subsections.

$$sim(doc1,doc2) = max(sim(doc1\_1,doc2\_1), sim(doc1\_2,doc2\_2),...,sim(doc1\_n, doc2\_n))$$

The performance of the models will be calculated as a ratio of the references of a document present in the top fifty (50) similar documents of a given document.

$$perf(mi, dj) = references\_in\_dj \quad mi\_similar\_documents[0:50]/references\_in\_dj$$

That is, given a model *mi*, and a document *dj*, the performance of *mi* on *dj* is the ratio of the intersection of references in *dj* and the top fifty (50) similar documents of *mi* on *dj* to all the references in *dj*.

1.4 Thesis Outline

In Chapter 1 of this thesis, the concepts Information Retrieval (IR), and Document Similarity are discussed. Furthermore, the problem statement and solution are described. Chapter 2 covers the literature review on document similarity, its early days, current trends, and some related work. In chapter 3, document similarity is discussed in greater detail along with the machine learning models of focus. In Chapter 4, the experiment design is discussed along with data collection, preprocessing, and statistical analysis of the data. Chapter 5 discusses the results of the various models without any enhancement(s), and the results of the model(s) considering parts of the document rather than the whole document. In chapter 6, the results of the experiments are discussed, and techniques to improve performance are suggested. Finally, chapter 7 concludes the thesis and postulates future work.

Chapter 2

Literature Review

The method described in this thesis builds on essential work in Information Retrieval (IR) as well as key ideas from Natural Language Processing (NLP) and Text-Mining.

2.1 1957-1994

The rise of automated Information Retrieval really begins with H.P. Luhn in 1957. Although document searching goes back long before this period (Sanderson and Croft 2012), the methods used prior to Luhn including Boolean search are not relevant to our research. Our interest in this thesis is in what the IR community calls ‘ad-hoc’ retrieval, which refers to the task of returning information resources related to a user query formulated in natural language rather than a carefully defined Boolean query.

Luhn was interested in automatic retrieval as well as automatic summarization of documents while working at IBM. Luhn proposed a method where each document in a collection was assigned a score indicating its relevance to a query (Luhn 1957). In another paper, Luhn suggested “that the frequency of word occurrence in an article furnishes a useful measurement of word significance” (Luhn 1958).

Gerard Salton, a Professor at Cornell University whose research group developed the SMART (System for the Mechanical Analysis and Retrieval of Text) Information Retrieval System in the 1960s took Luhn’s work to another level. In a paper memorializing Salton after his death in 1995 (Crouch et al. 1996) said of Salton “He was a brilliant computer scientist and the man most responsible for the establishment, survival, and recognition of Information Retrieval as a vital and important discipline in computer science."

One of Salton's main contributions was the TF-IDF vector space model (discussed in chapter 2) which is widely used in both IR and NLP. The vector space model introduced in (Salton et al. 1975) views documents as vectors consisting of term frequencies (TF) multiplied by a weighting called the Inverse Document Frequency (IDF) which was developed by (Jones 1972). Although the vector space model was introduced in 1975 it was initially viewed as an indexing method used in the SMART system and not considered as an innovation for use in general IR until the early 1980s (Dubin 2004).

Another innovation of the SMART system was relevance feedback. The first relevance feedback algorithm was developed by JJ. Rocchio (Rocchio. 1965) and added to the SMART system shortly after (Salton 1971). The SMART system allowed users to successively broaden or refine searches and incorporated numerous relevance feedback techniques since as one researcher on the SMART team stated, "since the user's original query is often inadequate, some sort of user interaction with the retrieval operation is desirable" (Kelly and Sugimoto 2013).

Despite all the research in IR and NLP, commercial products developed during this time such as DIALOG, ERIC, MEDLARS, LEXIS, and LEADERMART (Kelly and Sugimoto 2013) which were widely used by professional searchers and librarians, were largely restricted to Boolean searching. This situation didn't change until the early to mid-1990s with systems such as WESTLAW's WIN system (Turtle 1994) and the growth of web search engines.

2.2 TREC and SIGIR

Research in IR was recognized as an important branch of computer science way back in 1978 when the Association for Computing Machinery (ACM) created the Special Interest Group on Information Retrieval (SIGIR) and the SIGIR conference where much of the research in IR has been published and presented for the last 44 years. In 1992 the National Institute of Standards and Technology (NIST) created TREC (Text Retrieval Conference), an annual conference where many international research groups collaborate to build test collections several orders of magnitude larger than had been in existence before. This was in response to the IR community's concern at the time that existing datasets were too small for adequate testing of IR systems (Sanderson and Croft 2012). 1995-2013

With the growth of the internet, searching for text documents goes from an activity done by professional searchers and librarians to an activity practiced by the public (Kelly and Sugimoto 2013). Since all TREC Proceedings papers from 1992 through 2021 are available at https://trec.nist.gov/pubs we can see that from 1992 to 2010 that a shift from Boolean searching to ad-hoc searches in web search engines is taking place. Much of the research is related to relevance ranking, query expanding, complex question answering, and multilingual systems. Even though relevance ranking existed since 1965 (Rocchio. 1965) it took on new relevance in the 1990s as Web search engines tried to differentiate themselves with their ranking of results. Ultimately Google became the dominant search engine with its PageRank algorithm which identified relevant documents from authoritative sources and eliminated pages from unscrupulous authors that discovered they could alter their ranking by manipulating the content of their pages

(Sanderson and Croft 2012). Query expanding also became a topic of new importance to search engines because users tend to use very short queries while hoping for good results. (In 2009 the average query length was 2.30 words, the same as that reported ten years before in 1999 (Carpineto and Romano 2012).) To see why query expansion is important for search engines consider the World Cup which is the most widely viewed and followed single sporting event in the world. A user searching World Cup on Google will receive 4 trillion results, however since the World Cup is going on now (at the time of this writing) in Qatar, most users typing in World Cup are interested in recent results or the upcoming schedule. Since Google keeps track of trending searches it knows this and will automatically expand a query from ‘World Cup’ to ‘World Cup 2022’ to get more accurate results. The topics which have dominated TREC in the years 1995-2013 are interesting to the IR community but not of interest to this thesis work. However, during this period one area of interest was TREC HARD (Highly Accurate Retrieval from Documents). This topic is relevant to our research because we wish to conduct literature reviews based on a single source document rather than a query or queries. However, the HARD track of TREC depends on user feedback which we wish to avoid in our method.

2.3 2013-Present

The semantic vector space models of language represent each word as a real-valued vector. Consequently, these vectors can be used as features in NLP tasks such as question answering, document classification, information retrieval, etc. (Pennington et al, 2014). Prior to 2013, global matrix factorization methods such as latent semantic analysis were the main family of models for learning word vectors. However, Mikolov et al. introduced the local context window methods such as skip-gram (2013c). Aside from TF-

IDF, all the models discussed in chapter 2 were developed using some variation of word vectors. As stated in 2.2.5, GloVe is a combination of the advantages of the two popular model families: global matrix factorization and local context window methods (Pennington et al, 2014). Mikolov et al. developed word2vec (2013a) for representing word vectors using their context to ensure that words of similar meanings and/or use are placed close to each other in vector space. In 2014, Mikolov et al. introduced doc2vec which was an improvement from word2vec that allowed the model to learn fixed-length feature representations from variable-length pieces of texts, such as sentences, paragraphs, and documents. While it is considerably slower than the others, doc2vec can represent sentences, paragraphs, and documents as vectors in the Vector Space Model.

Both the context and content of a body of text are integral to successfully translating or interpreting such text. Thus, it is essential that the application of Deep Learning (DL) models on texts should cover the morphological, syntactic, semantic, and pragmatic layers of natural language (Braşoveanu and Andonie, 2020). Due to the sparseness of training data, building models/networks that met all the requirements of text analysis and machine translation was a significant challenge. The first Transformer network (Vaswani et al., 2017) showed that it was possible to design networks that achieve good results for Natural Language Processing (NLP) tasks with a set of multiple sequential attention layers. Transformers generally contain an encoder and a decoder. Transformers (using their encoder and decoder) transform input sequences into output sequences in deep learning applications. An example of an input sequence could be the sentence “I am writing a paper while listening to music”. The corresponding output sequence could be a translation of the sentence to French or Italian. Using multiple layers, although typically paired, transformers encode the input sequence using multi-attention layers and a feed-forward layer. Due to its reliance on attention, transformers use a recurrent neural network (RNN) that passes all hidden states of the encoder as context to the decoder. While passing all hidden states to the decoder does result in more processing, it provides the decoder with full context of the input thus preventing any loss in translation of the output. In the original paper introduced by Vaswani et al. (2017), the transformer had six (6) encoders and six (6) decoders.

Over the last couple of years, hundreds of papers and language models inspired by Transformers have been published, the best-known being BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), AlBERT (Lan et al., 2019), etc. Some of the most popular Transformer models are included in the Transformers library, maintained by HuggingFace. We discussed BERT in chapter 2 as one of the models that we will be covering in this paper.

2.4 Papers Closely Related to This Thesis Research

As discussed in the introduction, the idea behind this project is to find a method to automate and enhance the conducting of literature reviews. We assume that the key papers in a literature review are those that are ultimately cited by the finished research paper. Therefore, we wish to build a system that takes the text of a draft paper and finds papers that should be cited by that draft. Of course, building such a system will take a team and resources beyond the scope of a Master’s thesis so we limit this research to testing multiple clustering and similarity methods such as TF-IDF, Word2Vec, Doc2Vec, BERT, etc.

One area of research related to automating literature review is the so-called Systematic Literature Review (SLR) (Feng et al. 2018). SLRs are very labor-intensive that can often take a year or more to compile and generally are restricted to broad areas of science. As an example, one might compile an SLR on all evidence-based-medical approaches to treating Diabetes. The goal in such an endeavor is to assemble possibly thousands of relevant papers and organize them to call out the most important of such papers. The Feng study discussed how text-mining techniques could be used to create an SLR of Software Engineering. However, while identifying all the important papers within a subfield of science is a worthy goal, it is not useful to the researcher who is working on a literature review within a very narrow area of science such as this thesis is trying to address.

Perhaps the closest work related to our topic is (Erekhinskaya et al. 2016) who wished to automate the work in doing a literature review as well. However, their approach is more of an extractive summarization approach where the method could search through a library of 100,000 articles per day per CPU core and automatically extract knowledge to populate the predefined document template for each article. In other words, their method found papers on predefined topics which is a completely different approach than what we propose.

In our approach we assume that most researchers have an idea for a paper and do a preliminary search to see if there is anything similar in the literature. If nothing is found, then the researcher begins to write an initial draft. The idea here is that the method can take that initial draft paper and automatically identify papers that are related to specific parts of the new paper and should be cited by it. The Erekhinskaya et al. method allows a researcher to identify papers relevant to specific topics, which is probably what researchers should do in a careful literature review. However, in new areas of research topic names might not be established. The method proposed here will find papers that have sections of text that are similar to sections of text within the target paper rather than a typical search which attempts to identify papers that are most similar overall to a target paper.

Chapter 3

Document Similarity

In this chapter, we will discuss document similarity, the different methods used in calculating document similarity, and the different machine learning models that are applicable to this thesis. This chapter thus provides the requisite knowledge or background for the rest of the thesis.

3.1 Overview

Document similarity is the measure of how similar (or not) a set of documents are given a query document. However, the concept of similarity between two documents is debatable as readers often have different rules for claiming similarity (Bar et al. 2011). Concerning the general concept of similarity, Goodman (1972), and Bar et al. (2011) argue that similarity is an ill-defined notion unless one can say to what aspects similarity relates. Goodman (1972) provides a useful illustration of how different people at an airport would consider luggage bags to be similar. The pilot just considers a bag's weight, whereas the passenger evaluates them based on ownership and destination, whereas a spectator might compare bags based on shape, size, or color.

Recommender systems provide researchers with relevant papers for their work using document similarity measures when user feedback is sparse or unavailable (Beel, 2016). Given that similarity can be ambiguous, similarity in research papers is often concerned with multiple facets of the presented research, e. g., method, findings (Huang et al., 2020). Document similarity can be applied to a series of tasks, for example: classifying authorship of a paper, plagiarism detection, paraphrase detection etc. Apropos of that, document similarity should be formalized based on the geometric model of conceptual spaces along three dimensions inherent to texts: *structure*, *style*, and *content* (Bar et al, 2011). *Structure* refers to the internal developments of a given text, e.g. the order of sections. *Style* refers to grammar, usage, mechanics, and lexical complexity (Attali and Burstein, 2006). *Content* addresses all facts and their relationships within a text. For the purposes of this thesis, we will be considering the *content* of the papers for the similarity of the documents.

Table 1 illustrates different tasks and their associated dimensions as outlined by Bar et al (2011).

Table 1

Natural Language Processing Tasks and Associated Dimensions

Task	*Structure*	*Style*	*Content*
Authorship Classification		X	
Automatic Essay Scoring	X	X	X
Information Retrieval	X	X	X
Paraphrase Recognition			X
Plagiarism Detection		X	X
Question Answering			X
Short Answer Grading	X	X	X
Summarization	X		X

Task	*Structure*	*Style*	*Content*
Text Categorization			X
Text Segmentation	X		X
Text Simplification	X		X
Word Sense Alignment			X

Document similarity is based on two concepts: data representation and similarity measure. In data representation, most documents are encoded based on the Vector Space Document (VSD) (Salton et al, 1975). A feature vector of the words that appear in all of the documents in a data collection serves as the foundation of the data model's framework. Because words are the fundamental units in most natural languages (including English), the VSD model typically considers a distinct word that appears in the texts to be an atomic feature term (Paul et al, 2016). Using one of the many similarity measures based on the two corresponding feature vectors, such as the cosine measure, Jaccard measure, and Euclidean distance, the similarity between two documents is calculated.

As Li and Han (2013) noted, numerous metrics such as Euclidean distance-based metric, Cosine, Jaccard, Dice, Jensen- Shannon Divergence based metric have been proposed for the calculation of similarity between two documents for multiple natural language processing tasks. Cosine, calculated as the dot-product of two normalized vectors, is the most popular one. It measures the angle between two vectors (Li and Han, 2013).

$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Figure 1. Formula for Calculating Cosine Similarity

The angle given by cosine is inversely proportional to the similarity of the two documents. Thus, the lower the angle, the more similar the two documents are. Given three vectors (A, B, C) and their angles as shown below,

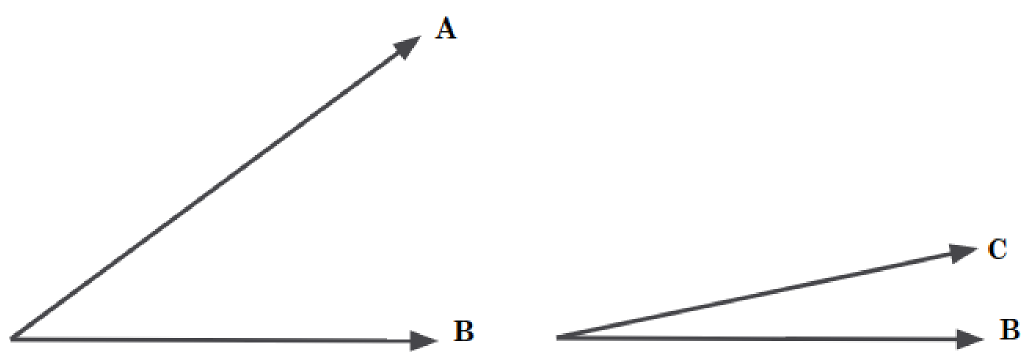

Figure 2. Cosine Similarity Measure

The above figure implies that vector C is more similar to vector B than vector A to B. For this thesis, all similarity metrics will be calculated using cosine similarity.

3.2 Machine Learning Models

One of the core concepts of document similarity is data representation in a vector space as mentioned above. In this section, we discuss the different machine learning models and techniques used to represent the document(s) in vector space.

3.2.1 TF-IDF

Term Frequency and Inverse Document Frequency (TF-IDF) is a numerical statistic that shows the relevance of keywords to some specific documents (Qaiser and Ali, 2018). Using TF-IDF, we can identify or classify documents based on the words appearing in those documents and their frequency. As the name suggests, TF-IDF is a combination of two concepts, Term Frequency (TF) and Inverse Document Frequency (IDF). TF is used to measure frequency of a given term in a document (Hakim et al, 2015). IDF is used to determine the importance of a word to a given document. Because TF treats all words equally, stop words (words with no significance such as “of”) are prone to being ranked high given their high frequency even though they do not provide any context for identifying a given document. IDF prevents this by assigning a lower weight to high frequency words and a higher weight to low frequency words. TF-IDF is the product of TF and IDF. Below, *Figure 3*, shows the mathematical formula for calculating TF, IDF, and TF-IDF.

$$\mathbf{tf}(t,d) = \frac{f_d(t)}{\max\limits_{w \in d} f_d(w)}$$

$$\mathbf{idf}(t,D) = \ln\left(\frac{|D|}{|\{d \in D : t \in d\}|}\right)$$

$$\mathbf{tfidf}(t,d,D) = \mathbf{tf}(t,d) \cdot \mathbf{idf}(t,D)$$

$$\mathbf{tfidf}'(t,d,D) = \frac{\mathbf{idf}(t,D)}{|D|} + \mathbf{tfidf}(t,d,D)$$

$f_d(t) :=$ frequency of term t in document d

$D :=$ corpus of documents

Figure 3. TF-IDF

From Figure 3 above, we can summarize that:

tf = number of times the term appears in a document/total number of words in the document

idf = log(number of documents/number of documents the term appears)

*tf-idf = tf * idf*

We calculate the similarity between all the papers using the cosine similarity metric. Cosine similarity, as defined in the previous chapter, is calculated as the dot-product of two normalized vectors. It measures the angle between two vectors (Li and Han, 2013).

3.2.1.1 Example. Given the example documents:

documents= [

"The quick brown fox jumped over the lazy dog",

"The quick grey fox jumped over the lazy cat",

"The slow mouse ambled into the woods"

]

The derived stop words from the documents are:

Stopwords= {the, into, over}

And the resulting dictionary of words to be used for calculating their similarities:

Dictionary= {amble, brown, cat, dog, fox, grey, jump, lazy, mouse, quick, slow, woods }

alphabetize.

The table below shows the TF-IDF vector of the documents.

Table 2

TF-IDF Document Vector

	amble	*brown*	*cat*	*dog*	*fox*	*grey*	*jumped*	*lazy*	*mouse*	*quick*	*slow*	*woods*
doc0	0.0	0.48	0.0	0.48	0.37	0.0	0.37	0.37	0.0	0.37	0.0	0.0
doc1	0.0	0.0	0.48	0.0	0.37	0.48	0.37	0.37	0.0	0.37	0.0	0.0
doc2	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	0.0	0.5	0.5

Using the vector of the words above, the similarity matrix of the documents is given below:

Table 3

Similarity of Documents Using TF-IDF

	Doc0	*Doc1*	*Doc2*
Doc0	1.0	0.54	0.0
Doc1	0.54	1.0	0.0
Doc2	0.0	0.0	1.0

3.2.2 BERT

The Bidirectional Encoder Representations from Transformers (BERT) is a language representation model introduced by Jacob et al in 2018. BERT was created with the intention of pre-training deep bidirectional representations from unlabeled text by concurrently conditioning on both left and right context in all layers. The main difference between BERT and its predecessors (language representation models) is that the previous models were unidirectional thus restricting the power of pre-trained representations (Jacob et al, 2018). Unlike its predecessors, BERT implements a masked language model which enables the representation to fuse the left and the right context, consequently allowing the pre-training of a deep bidirectional Transformer. BERT is both simple and powerful. As demonstrated by Jacob et al (2018), on eleven natural language processing tasks, it achieves new state-of-the-art results, raising the General Language Understanding Evaluation (GLUE) score to 80.5% (7.7%-point absolute improvement), MultiNLI accuracy to 86.7% (4.6% absolute improvement), SQuAD v1.1 question answering Test F1 to 93.2 (1.5-point absolute improvement), and SQuAD v2.0 Test F1 to 83.1. (5.1-point absolute improvement). These performance metrics make BERT a good choice for one of the models of our experiment.

3.2.2.1 Example. Given the example documents:

documents= [

"The quick brown fox jumped over the lazy dog",

"The quick grey fox jumped over the lazy cat",

"The slow mouse ambled into the woods"

]

The derived stop words from the documents are:

Stopwords= {the, into, over}

And the resulting dictionary of words to be used for calculating their similarities:

Dictionary= {amble, brown, cat, dog, fox, grey, jump, lazy, mouse, quick, slow, woods}

alphabetize.

Below is a table of the similarity matrix of the documents using BERT

Table 4

Similarity Matrix of Documents Using BERT

	Doc0	*Doc1*	*Doc2*
Doc0	1.0	0.84	0.38
Doc1	0.84	1.0	0.38
Doc2	0.38	0.38	1.0

3.2.3 Doc2Vec

Le and Mikolov (2014) proposed *doc2vec* as an extension of *word2vec* (Mikolov et al., 2013a). *Word2vec* is discussed in the next section. Doc2vec implements *Paragraph Vector,* an unsupervised algorithm that learns fixed-length feature representations from variable-length pieces of texts, such as sentences, paragraphs, and documents (Le and Mikolov, 2014). As Le and Mikolov noted, machine learning methods need that the input be represented as a feature vector of fixed-length. Regarding text and text related tasks, Bag of Words (Harris, 1954) is the most used method of achieving fixed-length features. Bag-of-words features, despite being widely used, have two significant flaws: they neglect the semantics of the words and lose the ordering of the words (Le and Mikolov, 2014). Doc2vec represents each document by a dense vector which is trained to predict words in the document. By developing both Paragraph Vectors and word vectors using stochastic gradient descent and backpropagation (Rumelhart et al., 1986), the vector representation for doc2vec is trained to predict words in a paragraph.

3.2.3.1 Example. Given the example documents:

documents= [

"The quick brown fox jumped over the lazy dog",

"The quick grey fox jumped over the lazy cat",

"The slow mouse ambled into the woods"

]

The derived stop words from the documents are:

Stopwords= {the, into, over}

And the resulting dictionary of words to be used for calculating their similarities:

Dictionary= {amble, brown, cat, dog, fox, grey, jump, lazy, mouse, quick, slow, woods} alphabetize.

Below is a table of the similarity matrix of the documents using Doc2Vec

Table 5

Similarity Matrix of Documents Using Doc2Vec

	Doc0	*Doc1*	*Doc2*
Doc0	1.0	0.99	0.97
Doc1	0.99	1.0	0.97
Doc2	0.97	0.97	1.0

Note: The similarity between Doc2 and the other documents is higher than expected due to the vector size used in running the model. The similarity can be optimized by fine tuning the vector size to match the dictionary. Vector sizes and its impact on the performance of our models will be discussed in Chapter 7.

3.2.4 Word2Vec

Proposed by Mikolov et al (2013a), *word2vec* is an architecture for computing continuous vector representations of words from very large data sets. Prior to word2vec, many of the existing natural language processing algorithms and techniques treated words as atomic units with no notion of similarity amongst words. However, word2vec represents word vectors using its context so similar words are in close proximity in vector space. Word2vec uses a previously proposed technique (Mikolov et al., 2013b) for measuring the quality of the resulting vector representations, with the expectation that not only will similar words tend to be close to each other, but that words can have multiple degrees of similarity (Mikolov et al., 2013b). Word2vec proposes two new model architectures for learning distributed representations of words: continuous bag-of-words (CBOW), and continuous skip-gram. The CBOW architecture predicts the current word based on the context, and the Skip-gram predicts surrounding words given the current word. Shown below are the architectures of CBOW and skip-gram models.

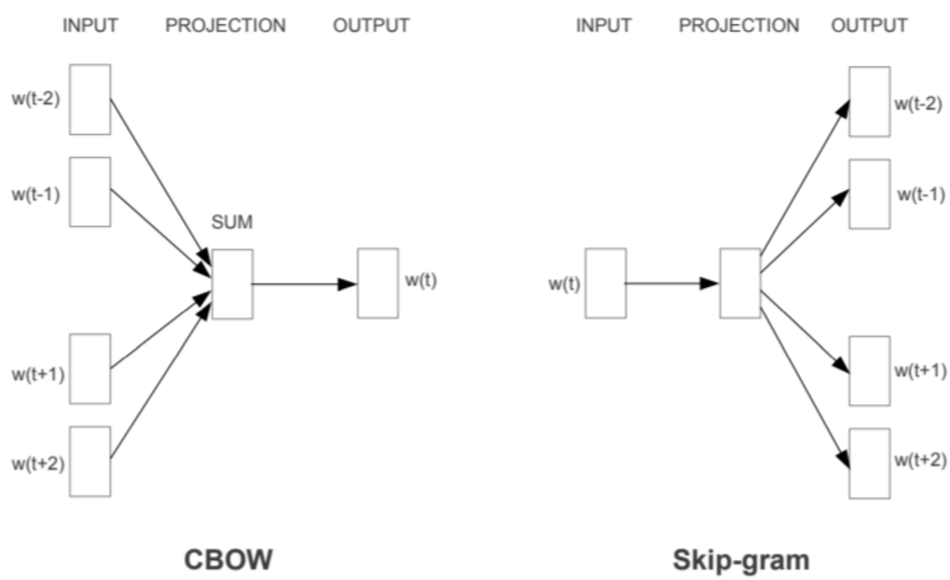

Figure 4. CBOW and Skip-Gram Model Architectures

3.2.4.1 Example. Given the example documents:

documents= [

"The quick brown fox jumped over the lazy dog",

"The quick grey fox jumped over the lazy cat", "The slow mouse ambled into the woods"

]

The derived stop words from the documents are:

Stopwords= {the, into, over}

And the resulting dictionary of words to be used for calculating their similarities:

Dictionary= {amble, brown, cat, dog, fox, grey, jump, lazy, mouse, quick, slow, woods}

alphabetize.

Below is a table of the similarity matrix of the documents using Word2Vec

Table 6

Similarity Matrix of Documents Using Word2Vec

	Doc0	*Doc1*	*Doc2*
Doc0	1.0	0.92	0.56
Doc1	0.92	1.0	0.55
Doc2	0.56	0.55	1.0

3.2.5 GloVe

Developed by Pennington et al in 2014, GloVe is a log-bilinear model with a weighted least-squares objective. It is a combination of the advantages of the two popular model families: global matrix factorization and local context window methods (Pennington et al, 2014). Although techniques like latent semantic analysis (LSA)(Deerwester et al, 1990), which is part of the global matrix factorization methods, effectively use statistical data, they perform poorly on the word analogy test, pointing to an inadequate vector space structure. Similarly, techniques like skip-gram (Mikolov et al, 2013c), which is part of the local context window methods, may perform better on the analogy task but because they are trained on individual local context windows rather than global co-occurrence counts, they do a poor job of utilizing the statistics of the corpus. However, by restricting training to the nonzero elements of a word-word co-occurrence matrix rather than the full sparse matrix or specific context windows in a huge corpus, Pennington et al (2014) were able to produce a model that performed at 75% on an analogy task while also improving its performance on similarity tasks.

3.2.5.1 Example. Given the example documents:

documents= [

"The quick brown fox jumped over the lazy dog",

"The quick grey fox jumped over the lazy cat",

"The slow mouse ambled into the woods"

]

The derived stop words from the documents are:

Stopwords= {the, into, over}

And the resulting dictionary of words to be used for calculating their similarities:

Dictionary= {amble, brown, cat, dog, fox, grey, jump, lazy, mouse, quick, slow, woods} alphabetize.

Below is a table of the similarity matrix of the documents using GloVe

Table 7

Similarity Matrix of Documents Using GloVe

	Doc0	*Doc1*	*Doc2*
Doc0	1.0	0.95	0.66
Doc1	0.95	1.0	0.65
Doc2	0.66	0.65	1.0

Chapter 4

Experiment Design

In this section, we will discuss our corpus, gathering criteria, statistics, and pre-processing.

4.1 Overview

We first wish to remind the reader that the idea behind this research is to find a method to automate and enhance the conducting of literature reviews. We assume that the key papers in a literature review are those that are ultimately cited by the finished research paper. Therefore, we need an experiment that will quantify how often an automated method would identify key papers that would be found in a traditional literature review.

The basic idea is that given a field of research (Neural Networks for example) we select a paper at random. We then test multiple methods to identify similar papers (e.g., TF-IDF, BERT, Doc2Vec etc.) and ask how many of the actual references are among the top scoring similar papers? One thing that makes such an experiment difficult is that there is not a universal corpus that contains all papers within subfields of computer science that make full-text papers available. One can purchase subsets of fields from Elsevier, IEEE, ACM, but getting a full set of all papers in several subfields would be cost-prohibitive. As a solution, we have created a corpus from the free set of pre-prints at Arxiv.org. The limitation with this data set is that most of the paper references will not be in corpus. We therefore compile a corpus for each subfield, randomly select 4 target papers, and then seed our corpus with additional full-text articles referenced by our target papers. Since the similarity methods only care about the text and not the source of the papers, any method that preferentially chooses a high percentage of the referenced papers is a candidate for automating and enhancing the conducting of literature reviews.

4.2 Data Collection

To perform the experimentation with multiple existing models, we gathered a corpus of 9088 documents from four different but related fields: Artificial Intelligence (AI), Neural Networks (NN), Virtual Reality (VR), and Natural Language Processing (NLP). The general similarity between the corpus set provides the appropriate environment for testing the accuracy of the models based on the number of references correctly identified as similar documents. All the documents in our corpus were retrieved from arxiv (https://arxiv.org/) by automating its document retrieval API. Using a python script, we retrieved documents matching categories outlined in the section above. The documents returned by the API were then converted to text documents using the python package tika. We limited the documents downloaded to those with thirty (30) or less pages. Across the corpus, the average number of pages was above fifteen (15), thus, testing all models against a relatively large body of text.

Table 8

Data Set Statistics of Each Document Category

Category	*Average Page Count*	*Average Reference Count*	*Total Documents*
Virtual Reality	17	40	1627
Neural Networks	16	35	4152
Natural Language Processing	17	30	1388
Artificial Intelligence	15	25	2133

4.3 Test Data

To test and measure the performance of existing models, we needed to build a dataset of papers and their references. Given a document, the goal of the models will be to provide the referenced documents as part of the most similar documents to that document. In each of the categories, four (4) documents were randomly selected to be used as test documents. Twenty (20) references were randomly selected from each of the chosen documents, downloaded, and added to the general corpus.

Table 9

Test Data Document Statistics

Category	*Average Page Count*	*Average # References*	*Total Documents*
Virtual Reality	22	89	4
Neural Networks	14	35	4
Natural Language Processing	17	30	4
Artificial Intelligence	15	25	4

4.4 Data Pre-Processing

Firstly, the documents downloaded from arxiv were limited to documents within the range of nine (9) and thirty (30) pages. This provides a sizable corpus from which we can get an accurate experiment based on the number of splits each document can be split into. All documents collected (in PDF) were converted into text only documents using a python package, tika. For the final step, we removed stop-words from the texts. Stop-words are frequently occurring, inconsequential words in natural languages; in English, they are often categorized as prepositions, conjunctions, and adverbs, for example: and the, is, of etc. Stop-word removal is an important preprocessing technique used in Natural Language processing tasks to improve the performance of the models associated with the tasks (Raulji and Saini, 2016).

Chapter 5

Results

In this chapter, we will evaluate the performance of the different base models on the task of accurately detecting referenced papers as similar documents. The score of each model is calculated as a ratio of the number of references in the top fifty (50), and hundred (100) similar documents as projected by each model.

model_score1 = number_of_referenced_papers_in_top_50/50

model_score2 = number_of_referenced_papers_in_top_100/100

5.1 Base Model Evaluations

5.1.1 TF-IDF

Using scikit-learn, we implemented a TF-IDF model with *max_features* of 64. According to the scikit-learn documentation, the model builds a vocabulary that only considers the top *max_features* ordered by frequency across the corpus. During experimentation, we tried multiple values for max_features (32, 128, 200) but maintained sixty-four (64) because it provided the best result and performance. Stop-words were already removed in our preprocessing step; thus, we did not have to provide the model with the *stop-words* argument.

Let us consider the performance of TF-IDF on the paper "Uni-Perceiver-MoE: Learning Sparse Generalist Models with Conditional MoEs". This paper has 17 references as shown in the figure below:

```
['NeurIPS-2021-vatt-transformers-for-multimodal-self-supervised-learning-from-raw-video-audio-and-text-Paper.txt',
 'GradNorm- Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks.txt',
 'Is Space-Time Attention All You Need for Video Understanding_.txt',
 'AN IMAGE IS WORTH 16X16 WORDS- TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE.txt',
 'BERT- Pre-training of Deep Bidirectional Transformers for Language Understanding.txt',
 'UNITER- UNiversal Image-TExt Representation Learning.txt',
 'Masked-attention Mask Transformer for Universal Image Segmentation.txt',
 'Embodied Multimodal Multitask Learning.txt',
 'Microsoft COCO Captions- Data Collection and Evaluation Server.txt',
 'Layer Normalization.txt',
 'ImageNet_a_Large-Scale_Hierarchical_Image_Database.txt',
 'Conceptual 12M- Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts.txt',
 'Biased Mixtures Of Experts- Enabling Computer Vision Inference Under Data Transfer Limitations.txt',
 'BAM! Born-Again Multi-Task Networks for Natural Language Understanding.txt',
 'ViViT- A Video Vision Transformer.txt',
 'Collecting Highly Parallel Data for Paraphrase Evaluation.txt',
 'Massively Multilingual Neural Machine Translation in the Wild- Findings and Challenges.txt']
```

Figure 5. Uni-Perceiver-MoE: Learning Sparse Generalist Models with Conditional MoEs References

After running the TF-IDF model, the top hundred (100) similar papers to the input paper are shown below:

```
models_performance("tfidf")
```

```
CALCULATING AI PAPERS
Paper: BAM!_Born-Again_Multi-Task_Networks_for_Natural_Language_Understanding.txt 0.7929317059293904
Paper: BAM! Born-Again Multi-Task Networks for Natural Language Understanding.txt 0.7929317059293904
Paper: NeurIPS-2020-the-lottery-ticket-hypothesis-for-pre-trained-bert-networks-Paper.txt 0.7755472443796702
Paper: Generic_Neural_Architecture_Search_via_Regression.txt 0.759294951462387
Paper: Multi-task_learning_for_natural_language_processing_in_the_2020s:_where
__are_we_going?.txt 0.7576417584103552
Paper: Targeting_the_Benchmark:_On_Methodology_in_Current_Natural_Language
__Processing_Research.txt 0.7395698876995841
Paper: Enabling_Robots_to_Draw_and_Tell:_Towards_Visually_Grounded_Multimodal
__Description_Generation.txt 0.7386462627400106
Paper: Beneficial_Perturbation_Network_for_designing_general_adaptive
__artificial_intelligence_systems.txt 0.7385809787138006
Paper: Embodied Multimodal Multitask Learning.txt 0.7380932504321542
Paper: MULTI-TASK LEARNING WITH DEEP NEURAL NETWORKS- A SURVEY.txt 0.7373038247481581
Paper: SurgeonAssist-Net:_Towards_Context-Aware_Head-Mounted_Display-Based
__Augmented_Reality_for_Surgical_Guidance.txt 0.7336595370059497
Paper: Molecular_representation_learning_with_language_models_and
__domain-relevant_auxiliary_tasks.txt 0.7319664346651199
Paper: What_can_we_learn_from_Semantic_Tagging?.txt 0.7203012744932494
Paper: Multi-Task_Trust_Transfer_for_Human-Robot_Interaction.txt 0.7168599444241178
Paper: A_Generative_Language_Model_for_Few-shot_Aspect-Based_Sentiment_Analysis.txt 0.7157084500693336
Paper: SBNet:_Segmentation-based_Network_for_Natural_Language-based_Vehicle
__Search.txt 0.715299103101266
Paper: HOUDINI:_Lifelong_Learning_as_Program_Synthesis.txt 0.7151547817871661
Paper: Conceptual 12M- Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts.txt 0.7132029653110755
Paper: NLX-GPT:_A_Model_for_Natural_Language_Explanations_in_Vision_and
__Vision-Language_Tasks.txt 0.7099650013710743
Paper: Experimenting_with_Self-Supervision_using_Rotation_Prediction_for_Image
__Captioning.txt 0.7089430880492869
Paper: Task-Oriented_Dialogue_System_as_Natural_Language_Generation.txt 0.7042958296718699
Paper: Energon:_Towards_Efficient_Acceleration_of_Transformers_Using_Dynamic
__Sparse_Attention.txt 0.7032972302620845
Paper: GAIA:_A_Transfer_Learning_System_of_Object_Detection_that_Fits_Your
__Needs.txt 0.7030968000951869
Paper: GradNorm- Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks.txt 0.6981947082953861
Paper: UBERT:_A_Novel_Language_Model_for_Synonymy_Prediction_at_Scale_in_the
__UMLS_Metathesaurus.txt 0.6967890387077637
Paper: Multitask Learning.txt 0.6934226440115278
Paper: On_Hiding_Neural_Networks_Inside_Neural_Networks.txt 0.691234041100718
Paper: Surgical_Visual_Domain_Adaptation:_Results_from_the_MICCAI_2020
__SurgVisDom_Challenge.txt 0.6890120888933141
Paper: Pre-trained_Models_for_Natural_Language_Processing:_A_Survey.txt 0.6881613959294952
Paper: SiT:_Self-supervised_vIsion_Transformer.txt 0.6877295437884684
Paper: A_Generalizable_Approach_to_Learning_Optimizers.txt 0.6868139828051456
Paper: RoboTurk:_A_Crowdsourcing_Platform_for_Robotic_Skill_Learning_through
__Imitation.txt 0.6836485614413972
Paper: UNITER- UNiversal Image-TExt Representation Learning.txt 0.6825531623557104
```


```
__Imitation.txt 0.6836485614413972
Paper: UNITER- UNiversal Image-TExt Representation Learning.txt 0.6825531623557104
```


```
Paper: Gradual_Tuning:_a_better_way_of_Fine_Tuning_the_parameters_of_a_Deep
__Neural_Network.txt 0.6810478900407452
Paper: Heterogeneous_Multi-task_Learning_for_Human_Pose_Estimation_with_Deep
__Convolutional_Neural_Network.txt 0.6806903280134166
Paper: Deductive_Association_Networks.txt 0.679815830353668
Paper: Image-based_Natural_Language_Understanding_Using_2D_Convolutional_Neural
__Networks.txt 0.677025881447275
Paper: An_Open-Source_Dataset_and_A_Multi-Task_Model_for_Malay_Named_Entity
__Recognition.txt 0.6756405662791546
Paper: Cranial_Implant_Design_via_Virtual_Craniectomy_with_Shape_Priors.txt 0.6741637648441431
Paper: Classification_of_Long_Sequential_Data_using_Circular_Dilated
__Convolutional_Neural_Networks.txt 0.673771914474152
Paper: Recurrent_Neural_Network_for_Text_Classification_with_Multi-Task
__Learning.txt 0.6726141424337985
Paper: Explaining_Chest_X-ray_Pathologies_in_Natural_Language.txt 0.6723668046585752
Paper: Which_Tasks_Should_Be_Learned_Together_in_Multi-task_Learning?.txt 0.6701831801133641
Paper: Natural_language_understanding_for_task_oriented_dialog_in_the
__biomedical_domain_in_a_low_resources_context.txt 0.6697013790162479
Paper: Grounding_Natural_Language_Instructions:_Can_Large_Language_Models
__Capture_Spatial_Information?.txt 0.669633418337065
Paper: One-shot_Scene_Graph_Generation.txt 0.6690285388891449
Paper: A_Mobile_Manipulation_System_for_One-Shot_Teaching_of_Complex_Tasks_in
__Homes.txt 0.6659656656762449
Paper: Real_vs_Simulated_Foveated_Rendering_to_Reduce_Visual_Discomfort_in
__Virtual_Reality.txt 0.6653198830442411
Paper: Localizing_Catastrophic_Forgetting_in_Neural_Networks.txt 0.6650098947522538
Paper: Sparse_Meta_Networks_for_Sequential_Adaptation_and_its_Application_to
__Adaptive_Language_Modelling.txt 0.6635994979067406
Paper: Making_Pre-trained_Language_Models_End-to-end_Few-shot_Learners_with
__Contrastive_Prompt_Tuning.txt 0.6609728009514174
Paper: HumanMeshNet:_Polygonal_Mesh_Recovery_of_Humans.txt 0.6605495091877839
Paper: Development of NASA-TLX (Task Load Index)- Results of Empirical and Theoretical Research.txt 0.6605017419263891
Paper: Making_the_Most_of_Text_Semantics_to_Improve_Biomedical_Vision--Language
__Processing.txt 0.6596004041621752
Paper: Language Models are Few-Shot Learners.txt 0.6593961422242932
Paper: HealthPrompt:_A_Zero-shot_Learning_Paradigm_for_Clinical_Natural
__Language_Processing.txt 0.6592932092913972
Paper: NeurIPS-2021-vatt-transformers-for-multimodal-self-supervised-learning-from-raw-video-audio-and-text-Paper.txt 0.6585218132685783
Paper: Exploring_Software_Naturalness_through_Neural_Language_Models.txt 0.6576044825090896
Paper: Component_Analysis_for_Visual_Question_Answering_Architectures.txt 0.6573281253572738
Paper: EyeNet:_A_Multi-Task_Network_for_Off-Axis_Eye_Gaze_Estimation_and_User
__Understanding.txt 0.6562251762923181
Paper: NaRLE:_Natural_Language_Models_using_Reinforcement_Learning_with_Emotion
__Feedback.txt 0.6551853000216318
Paper: Language_Tasks_and_Language_Games:_On_Methodology_in_Current_Natural
__Language_Processing_Research.txt 0.655048384774142
Paper: FRAGE:_Frequency-Agnostic_Word_Representation.txt 0.6526987913036083
Paper: CRUR:_Coupled-Recurrent_Unit_for_Unification,_Conceptualization_and
__Context_Capture_for_Language_Representation_--_A_Generalization_of_Bi
__Directional_LSTM.txt 0.6499414543730199
Paper: LEARNING END-TO-END GOAL-ORIENTED DIALOG.txt 0.648839320500277
Paper: Do_CNNs_Encode_Data_Augmentations?.txt 0.6483844818966868
Paper: An_Exploration_of_Prompt_Tuning_on_Generative_Spoken_Language_Model_for
__Speech_Processing_Tasks.txt 0.6470435887827347
Paper: IL-Net:_Using_Expert_Knowledge_to_Guide_the_Design_of_Furcated_Neural
__Networks.txt 0.6457804328490742
Paper: Transfer_Learning_from_LDA_to_BiLSTM-CNN_for_Offensive_Language
__Detection_in_Twitter.txt 0.6451545668684044
Paper: Help,_Anna!_Visual_Navigation_with_Natural_Multimodal_Assistance_via
__Retrospective_Curiosity-Encouraging_Imitation_Learning.txt 0.644128637810721
Paper: Multi-task_learning_for_virtual_flow_metering.txt 0.6438716039054917
```

Figure 6. Top 100 Similar References for Uni-Perceiver-MoE: Learning Sparse Generalist Models with Conditional MoEs Using TF-IDF

Consequently, we calculate the performance of the model on the paper by comparing the number of references accurately suggested as similar papers. In each of the figures below, we show the references in our paper that are part of the top fifty(50), and hundred(100) similar documents as predicted by our TF-IDF model. In each figure, we show the reference and its similarity score to the input paper. Furthermore, we calculate the percentage of references found and display it at the bottom of the list.

```
models_performance("tfidf")

CALCULATING AI PAPERS
Paper: BAM! Born-Again Multi-Task Networks for Natural Language Understanding.txt 0.7929317059293904
Paper: Embodied Multimodal Multitask Learning.txt 0.7380932504321542
Paper: Conceptual 12M- Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts.txt 0.7132029653110755
Paper: GradNorm- Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks.txt 0.6981947082953861
Paper: UNITER- UNiversal Image-TExt Representation Learning.txt 0.6825531623557104
Paper: NeurIPS-2021-vatt-transformers-for-multimodal-self-supervised-learning-from-raw-video-audio-and-text-Paper.txt 0.6585218132685783
Paper: Masked-attention Mask Transformer for Universal Image Segmentation.txt 0.6364917324658278
Paper: BERT- Pre-training of Deep Bidirectional Transformers for Language Understanding.txt 0.6201999196410577
0.47058823529411764
```

Figure 7. TF-IDF Performance on Sample Paper I

```
models_performance("tfidf")

CALCULATING AI PAPERS
Paper: BAM! Born-Again Multi-Task Networks for Natural Language Understanding.txt 0.7929317059293904
Paper: Embodied Multimodal Multitask Learning.txt 0.7380932504321542
Paper: Conceptual 12M- Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts.txt 0.7132029653110755
Paper: GradNorm- Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks.txt 0.6981947082953861
Paper: UNITER- UNiversal Image-TExt Representation Learning.txt 0.6825531623557104
0.29411764705882354
```

Figure 8. TF-IDF Performance on Sample Paper II

As shown above, our TF-IDF model predicted 29.41% of the actual references as part of the top fifty (50), and 47.05% when considering the top hundred (100) similar documents. Below is a table of the results for the TF-IDF model, for all the test documents.

Table 10

TF-IDF Model Performance on Corpus

Paper	*Category*	*Score (Top 50)*	*Score (Top 100)*
Functional Code Building Genetic Programming	AI	31.25%	31.35%
TwiBot-22: Towards Graph-Based Twitter Bot Detection	AI	0%	5.56%
Jewelry Shop Conversational Chatbot	AI	0%	0%
Uni-Perceiver-MoE: Learning Sparse Generalist Models with Conditional MoEs	AI	29.41%	47.05%
Visualization in virtual reality: a systematic review	VR	10%	10%
Joint Compute-Caching-Communication Control for Online Data-Intensive Service Delivery	VR	36.84%	47.37%

Paper	*Category*	*Score (Top 50)*	*Score (Top 100)*
6G Survey on Challenges, Requirements, Applications, Key Enabling Technologies, Use Cases, AI integration issues and Security aspects	VR	6.67%	6.67%
Quantifying the Effects of Working in VR for One Week	VR	44.44%	61.11%
Neo-GNNs: Neighborhood Overlap-aware Graph Neural Networks for Link Prediction	NN	5%	10%
Learning Vehicle Trajectory Uncertainty	NN	0%	10%
Early Transferability of Adversarial Examples in Deep Neural Networks	NN	0%	0%
Face-Dubbing++: Lip-Synchronous, Voice Preserving Translation of Videos	NN	16.67%	22.22%
NLU for Game-based Learning in Real: Initial Evaluations	NLP	6.25%	18.75%

Paper	Category	Score (Top 50)	Score (Top 100)
Multi-Agent Reinforcement Learning is A Sequence Modeling Problem	NLP	23.53%	35.29%
Differentially Private Model Compression	NLP	6.25%	6.25%
Quantum Neural Network Classifiers: A Tutorial	NLP	0%	0%

Based on Table 10, the category with the highest average score is VR, with an average score of 24.49% for the top 50 papers and 31.29% for the top 100 papers. This is higher than the average scores for the other categories, which are AI (15.17% and 20.99% for the top 50 and top 100, respectively), NLP (9.01% and 15.07% for the top 50 and top 100, respectively), and NN (5.42% and 10.55% for the top 50 and top 100 papers, respectively).

5.1.2 BERT

To calculate the cosine similarity of the documents using BERT, we need a pre-trained model to generate our document embeddings. For this purpose, we used *sentence-transformers*, a model that maps sentences and paragraphs to a 768-dimensional dense vector space and can be used in natural language processing tasks (Reimers and

Gurevych, 2019), and *bert-base-nli-mean-tokens.* While BERT was considerably faster than Doc2Vec, and Word2Vec, it is also less accurate and produces the least performance in terms of references detected as similar documents. Given the specificity of our dataset, it is possible that the tokens used did not provide enough context or information to the model. A possible path of future exploration would be to use a different token set for generating the sentence embeddings. Using our sample input paper, none of its references are shown as part of the top fifty (50) or hundred (100) similar documents.

Table 11

BERT Model Performance on Corpus

Paper	*Category*	*Score (Top 50)*	*Score (Top 100)*
Functional Code Building Genetic Programming	AI	6.25%	12.5%
TwiBot-22: Towards Graph-Based Twitter Bot Detection	AI	0%	0%
Jewelry Shop Conversational Chatbot	AI	0%	0%
Uni-Perceiver-MoE: Learning Sparse Generalist Models with Conditional MoEs	AI	0%	0%
Visualization in virtual reality: a systematic review	VR	5%	5%
Multi-Agent Reinforcement Learning is A Sequence Modeling Problem	NLP	0%	0%

Paper	*Category*	*Score (Top 50)*	*Score (Top 100)*
Joint Compute-Caching- Communication Control for Online Data-Intensive Service Delivery	VR	0%	0%
6G Survey on Challenges, Requirements, Applications, Key Enabling Technologies, Use Cases, AI integration issues and Security aspects	VR	6.67%	6.67%
Quantifying the Effects of Working in VR for One Week	VR	5.56%	5.56%
Neo-GNNs: Neighborhood Overlap-aware Graph Neural Networks for Link Prediction	NN	0%	0%
Learning Vehicle Trajectory Uncertainty	NN	0%	0%
Early Transferability of Adversarial Examples in Deep Neural Networks	NN	0%	0%
Face-Dubbing++: Lip-Synchronous, Voice Preserving Translation of Videos	NN	5.56%	5.56%
NLU for Game-based Learning in Real: Initial Evaluations	NLP	0%	0%
Differentially Private Model Compression	NLP	0%	0%
Quantum Neural Network Classifiers: A Tutorial	NLP	0%	0%

5.1.3 Doc2Vec

As stated in the previous section, Doc2Vec tokenizes sentences and documents to improve the performance of the model on natural language processing tasks. During experimentation, this approach shows obvious differences in the execution time of the model. While other models executed successfully within two (2) hours, Doc2Vec takes over forty-eight (48) hours to execute and return similar documents. Although the significant difference in run time (albeit negative) is a downside to using Doc2Vec, its performance regarding the task was the most impressive. We maintain the same vector size (100) as with the other models, provide a learning rate of 0.025, and ignore all words with a count of 1. As shown below, we see a significant difference and improvement in the number of references identified as similar documents to the given documents.

Figure 5 shows the references in the paper “Uni-Perceiver-MoE: Learning Sparse Generalist Models with Conditional MoEs”.

After running the Doc2Vec model, the top hundred (100) similar papers to the input paper are shown below:

```
[44]: models_performance("doc2vec")
```

```
CALCULATING AI PAPERS
Paper: UNITER- UNiversal Image-TExt Representation Learning.txt 0.7848378015718148
Paper: Object-aware_Video-language_Pre-training_for_Retrieval.txt 0.7218249848163333
Paper: Self-Training_Vision_Language_BERTs_with_a_Unified_Conditional_Model.txt 0.7105426430359956
Paper: Conceptual 12M- Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts.txt 0.6855020093609183
Paper: RegionViT:_Regional-to-Local_Attention_for_Vision_Transformers.txt 0.6713504733375976
Paper: Flamingo- a Visual Language Model for Few-Shot Learning.txt 0.6673440894452304
Paper: Masked-attention Mask Transformer for Universal Image Segmentation.txt 0.6563299112006764
Paper: Is Space-Time Attention All You Need for Video Understanding_.txt 0.649287209373354
Paper: Incorporating_Convolution_Designs_into_Visual_Transformers.txt 0.6451771399012851
Paper: Zero-Shot Text-to-Image Generation.txt 0.6196668957628159
Paper: SOFT:_Softmax-free_Transformer_with_Linear_Complexity.txt 0.6119902971334422
Paper: Masked Autoencoders Are Scalable Vision Learners.txt 0.6096169508649094
Paper: Explicit_Sparse_Transformer:_Concentrated_Attention_Through_Explicit
__Selection.txt 0.6043705287997916
Paper: Learning_Multilingual_Representation_for_Natural_Language_Understanding
__with_Enhanced_Cross-Lingual_Supervision.txt 0.6013413300557324
Paper: Visual_Representation_Learning_with_Self-Supervised_Attention_for
__Low-Label_High-data_Regime.txt 0.6010472438207272
Paper: ST-MoE:_Designing_Stable_and_Transferable_Sparse_Expert_Models.txt 0.5993456975231845
Paper: UTNet:_A_Hybrid_Transformer_Architecture_for_Medical_Image_Segmentation.txt 0.5992835225714213
Paper: Rethinking_and_Improving_Relative_Position_Encoding_for_Vision
__Transformer.txt 0.5903249398925389
Paper: NeurIPS-2021-vatt-transformers-for-multimodal-self-supervised-learning-from-raw-video-audio-and-text-Paper.txt 0.586656583171477
Paper: GAIA:_A_Transfer_Learning_System_of_Object_Detection_that_Fits_Your
__Needs.txt 0.5863584629733276
Paper: Aggregated_Pyramid_Vision_Transformer:_Split-transform-merge_Strategy
__for_Image_Recognition_without_Convolutions.txt 0.5830792502422544
Paper: CTAL:_Pre-training_Cross-modal_Transformer_for_Audio-and-Language
__Representations.txt 0.5815224550310376
Paper: Shifted_Chunk_Transformer_for_Spatio-Temporal_Representational_Learning.txt 0.5808336774748987
Paper: Container:_Context_Aggregation_Network.txt 0.5741536038018951
Paper: Embodied Multimodal Multitask Learning.txt 0.5711372428334973
Paper: Visual_Grounding_Strategies_for_Text-Only_Natural_Language_Processing.txt 0.5706603825797181
Paper: Zero-Shot_Transfer_VQA_Dataset.txt 0.5690586074755023
Paper: Visual_Question_Answering_as_Reading_Comprehension.txt 0.5658575509421206
Paper: NLX-GPT:_A_Model_for_Natural_Language_Explanations_in_Vision_and
__Vision-Language_Tasks.txt 0.5647867486303865
Paper: Object_Ordering_with_Bidirectional_Matchings_for_Visual_Reasoning.txt 0.5606528613216422
Paper: A_Survey_on_Dynamic_Neural_Networks_for_Natural_Language_Processing.txt 0.5597746689666773
Paper: AN IMAGE IS WORTH 16X16 WORDS- TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE.txt 0.5592047562681618
Paper: Explainable_Semantic_Space_by_Grounding_Language_to_Vision_with
__Cross-Modal_Contrastive_Learning.txt 0.552857211569001
Paper: DICT-MLM:_Improved_Multilingual_Pre-Training_using_Bilingual
__Dictionaries.txt 0.5511232369600739
Paper: A_Survey_of_Natural_Language_Generation.txt 0.5460116082820743
Paper: Hierarchically-Attentive_RNN_for_Album_Summarization_and_Storytelling.txt 0.5447991334038375
Paper: ViViT- A Video Vision Transformer.txt 0.5435878980188429
Paper: Energon:_Towards_Efficient_Acceleration_of_Transformers_Using_Dynamic
__Sparse_Attention.txt 0.5431457906352866
Paper: Facial_Expression_Recognition_with_Visual_Transformers_and_Attentional
__Selective_Fusion.txt 0.5401747330866346
Paper: Universal_Sentence_Representation_Learning_with_Conditional_Masked
__Language_Model.txt 0.5393830492635355
Paper: A_Survey_on_Contextual_Embeddings.txt 0.5388262714051802
Paper: Joint_Unsupervised_and_Supervised_Training_for_Multilingual_ASR.txt 0.5376866529417599
Paper: MAGIC:_Multimodal_relAtional_Graph_adversarIal_inferenCe_for_Diverse_and
__Unpaired_Text-based_Image_Captioning.txt 0.5373281663666771
Paper: MuVAM:_A_Multi-View_Attention-based_Model_for_Medical_Visual_Question
__Answering.txt 0.535958000557606
Paper: Massively Multilingual Neural Machine Translation in the Wild- Findings and Challenges.txt 0.5343780926649031
```

Paper: Massively Multilingual Neural Machine Translation in the Wild- Findings and Challenges.txt 0.5343780926649031

```
Paper: Pre-trained_Models_for_Natural_Language_Processing:_A_Survey.txt 0.5335828840379745
Paper: I2C2W:_Image-to-Character-to-Word_Transformers_for_Accurate_Scene_Text
__Recognition.txt 0.5329992973902961
Paper: Slimmable_Neural_Networks.txt 0.5315720901719174
Paper: DeFormer- Decomposing Pre-trained Transformers for Faster Question Answering.txt 0.5308210282724971
Paper: Assessing_the_Impact_of_Attention_and_Self-Attention_Mechanisms_on_the
__Classification_of_Skin_Lesions.txt 0.5292846830905078
Paper: Making_the_Most_of_Text_Semantics_to_Improve_Biomedical_Vision--Language
__Processing.txt 0.5284747028881163
Paper: A_Survey_on_Model_Compression_for_Natural_Language_Processing.txt 0.527108735509823
Paper: Improving_Biomedical_Pretrained_Language_Models_with_Knowledge.txt 0.5257910486945081
Paper: BERT- Pre-training of Deep Bidirectional Transformers for Language Understanding.txt 0.5253419110981818
Paper: Super_Interaction_Neural_Network.txt 0.5249535548473653
Paper: BERT:_Pre-training_of_Deep_Bidirectional_Transformers_for_Language
__Understanding.txt 0.5231168827110709
Paper: Overcoming_Catastrophic_Forgetting_beyond_Continual_Learning:_Balanced
__Training_for_Neural_Machine_Translation.txt 0.5219165629447738
Paper: i-Mix:_A_Domain-Agnostic_Strategy_for_Contrastive_Representation
__Learning.txt 0.5217364502620839
Paper: Cross-lingual_Adaption_Model-Agnostic_Meta-Learning_for_Natural_Language
__Understanding.txt 0.5207289144094864
Paper: Searching_for_fingerspelled_content_in_American_Sign_Language.txt 0.5173980608519991
Paper: PointDistiller:_Structured_Knowledge_Distillation_Towards_Efficient_and
__Compact_3D_Detection.txt 0.5162489071077865
Paper: nnFormer:_Interleaved_Transformer_for_Volumetric_Segmentation.txt 0.5149750500411064
Paper: Machine_Reading_Comprehension:_The_Role_of_Contextualized_Language
__Models_and_Beyond.txt 0.514784910897855
Paper: ANNA:_Enhanced_Language_Representation_for_Question_Answering.txt 0.5147532323318688
Paper: Multi-Modal_Learning_for_AU_Detection_Based_on_Multi-Head_Fused
__Transformers.txt 0.5144808579893821
Paper: Distilling_Virtual_Examples_for_Long-tailed_Recognition.txt 0.5141971700602131
Paper: lamBERT:_Language_and_Action_Learning_Using_Multimodal_BERT.txt 0.5132628889448917
Paper: Learning_Versatile_Neural_Architectures_by_Propagating_Network_Codes.txt 0.5128698677554105
Paper: Ouroboros:_On_Accelerating_Training_of_Transformer-Based_Language_Models.txt 0.5126199027292841
Paper: Transformers_in_Medical_Imaging:_A_Survey.txt 0.511609251371496
Paper: MULTI-TASK LEARNING WITH DEEP NEURAL NETWORKS- A SURVEY.txt 0.5091622854431701
Paper: GraphFPN:_Graph_Feature_Pyramid_Network_for_Object_Detection.txt 0.5086805345053285
Paper: Component_Analysis_for_Visual_Question_Answering_Architectures.txt 0.5076857081816504
Paper: ProphetNet-X:_Large-Scale_Pre-training_Models_for_English,_Chinese,
__Multi-lingual,_Dialog,_and_Code_Generation.txt 0.5069783808552241
Paper: SiT:_Self-supervised_vIsion_Transformer.txt 0.5045613177648309
Paper: A_Generative_Language_Model_for_Few-shot_Aspect-Based_Sentiment_Analysis.txt 0.5025894031612705
Paper: An_Exploration_of_Prompt_Tuning_on_Generative_Spoken_Language_Model_for
__Speech_Processing_Tasks.txt 0.5012362722229002
Paper: XFBoost:_Improving_Text_Generation_with_Controllable_Decoders.txt 0.5005144445004944
Paper: Biased Mixtures Of Experts- Enabling Computer Vision Inference Under Data Transfer Limitations.txt 0.4984007681344957
Paper: Pre-trained_Language_Model_Based_Active_Learning_for_Sentence_Matching.txt 0.49690358483096436
Paper: Learning_Hierarchical_Information_Flow_with_Recurrent_Neural_Modules.txt 0.49660421413084205
Paper: Pretraining_with_Artificial_Language:_Studying_Transferable_Knowledge_in
__Language_Models.txt 0.4954452035362444
Paper: Vision_Xformers:_Efficient_Attention_for_Image_Classification.txt 0.49399599163916563
Paper: TAVAT:_Token-Aware_Virtual_Adversarial_Training_for_Language
__Understanding.txt 0.4921207410034701
Paper: Reshaping_Robot_Trajectories_Using_Natural_Language_Commands:_A_Study_of
__Multi-Modal_Data_Alignment_Using_Transformers.txt 0.4918246632874326
Paper: Dynamic_Capacity_Networks.txt 0.4916609746640269
Paper: Dynamic_Routing_on_Deep_Neural_Network_for_Thoracic_Disease
__Classification_and_Sensitive_Area_Localization.txt 0.4905761602624775
Paper: Object-Centric_Representation_Learning_for_Video_Question_Answering.txt 0.4901292876036232
Paper: Efficient_Transfer_Learning_via_Joint_Adaptation_of_Network_Architecture
__and_Weight.txt 0.48974182674582295
Paper: One-shot_Scene_Graph_Generation.txt 0.48926313035745506
```

Figure 9. Top 100 Similar Documents for Uni-Perceiver-MoE: Learning Sparse Generalist Models with Conditional MoEs Using Doc2Vec

As with the previous sections, we show the references in our paper that are part of the top fifty (50), and hundred (100) similar documents as predicted by our Doc2Vec model. In each figure, we show the reference and its similarity score to the input paper. Finally, we calculate the percentage of references accurately predicted.

```
[27]: models_performance("doc2vec")

CALCULATING AI PAPERS
Paper: UNITER- UNiversal Image-TExt Representation Learning.txt 0.7848378015718148
Paper: Conceptual 12M- Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts.txt 0.6855020093609183
Paper: Masked-attention Mask Transformer for Universal Image Segmentation.txt 0.6563299112006764
Paper: Is Space-Time Attention All You Need for Video Understanding_.txt 0.649287209373354
Paper: NeurIPS-2021-vatt-transformers-for-multimodal-self-supervised-learning-from-raw-video-audio-and-text-Paper.txt 0.586656583171477
Paper: Embodied Multimodal Multitask Learning.txt 0.5711372428334973
Paper: AN IMAGE IS WORTH 16X16 WORDS- TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE.txt 0.5592047562681618
Paper: ViViT- A Video Vision Transformer.txt 0.5435878980188429
Paper: Massively Multilingual Neural Machine Translation in the Wild- Findings and Challenges.txt 0.5343780926649031
Paper: BERT- Pre-training of Deep Bidirectional Transformers for Language Understanding.txt 0.5253419110981818
Paper: Biased Mixtures Of Experts- Enabling Computer Vision Inference Under Data Transfer Limitations.txt 0.4984007681344957
Paper: GradNorm- Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks.txt 0.4856498042494148
0.7058823529411765
```

Figure 10. Doc2Vec Performance on Sample Paper I

```
: models_performance("doc2vec")

CALCULATING AI PAPERS
Paper: UNITER- UNiversal Image-TExt Representation Learning.txt 0.7848378015718148
Paper: Conceptual 12M- Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts.txt 0.6855020093609183
Paper: Masked-attention Mask Transformer for Universal Image Segmentation.txt 0.6563299112006764
Paper: Is Space-Time Attention All You Need for Video Understanding_.txt 0.649287209373354
Paper: NeurIPS-2021-vatt-transformers-for-multimodal-self-supervised-learning-from-raw-video-audio-and-text-Paper.txt 0.586656583171477
Paper: Embodied Multimodal Multitask Learning.txt 0.5711372428334973
Paper: AN IMAGE IS WORTH 16X16 WORDS- TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE.txt 0.5592047562681618
Paper: ViViT- A Video Vision Transformer.txt 0.5435878980188429
Paper: Massively Multilingual Neural Machine Translation in the Wild- Findings and Challenges.txt 0.5343780926649031
0.5294117647058824
```

Figure 11. Doc2Vec Performance on Sample Paper II

As shown above, our Doc2Vec model predicted 52.94% of the references as part of the top fifty (50), and 70.59% when considering the top hundred (100) similar documents, thus producing the highest accuracy on the sample document. Below is a table of the results for the Doc2Vec model, for all the test documents.

Table 12

Doc2Vec Model Performance on Corpus

Paper	*Category*	*Score (Top 50)*	*Score (Top 100)*
Functional Code Building Genetic Programming	AI	81.25%	87.5%
TwiBot-22: Towards Graph-Based Twitter Bot Detection	AI	77.78%	83.33%
Jewelry Shop Conversational Chatbot	AI	7.69%	7.69%
Uni-Perceiver-MoE: Learning Sparse Generalist Models with Conditional MoEs	AI	52.94%	70.59%
Visualization in virtual reality: a systematic review	VR	35%	45%
Joint Compute-Caching-Communication Control for Online Data-Intensive Service Delivery	VR	68.42%	89.47%

Paper	*Category*	*Score (Top 50)*	*Score (Top 100)*
6G Survey on Challenges, Requirements, Applications, Key Enabling Technologies, Use Cases, AI integration issues and Security aspects	VR	66.67%	73.33%
Quantifying the Effects of Working in VR for One Week	VR	61.11%	72.22%
Neo-GNNs: Neighborhood Overlap-aware Graph Neural Networks for Link Prediction	NN	35%	55%
Learning Vehicle Trajectory Uncertainty	NN	70%	75%
Early Transferability of Adversarial Examples in Deep Neural Networks	NN	47.06%	52.94%
Face-Dubbing++: Lip-Synchronous, Voice Preserving Translation of Videos	NN	83.33%	88.89%
NLU for Game-based Learning in Real: Initial Evaluations	NLP	43.75%	62.5%

Paper	*Category*	*Score (Top 50)*	*Score (Top 100)*
Multi-Agent Reinforcement Learning is A Sequence Modeling Problem	NLP	64.71%	64.71%
Differentially Private Model Compression	NLP	37.5%	50%
Quantum Neural Network Classifiers: A Tutorial	NLP	44.44%	44.44%

Based on Table 12, we can infer that like TF-IDF, the category with the highest average score is VR, with an average score of 60.86% for the top 50 papers and 70.01% for the top 100 papers. This is higher than the average scores for the other categories, which are NN (58.85% and 67.96% for the top 50 and top 100, respectively), AI (54.92% and 62.28% for the top 50 and top 100, respectively), and NLP (47.6% and 55.41% for the top 50 and top 100 papers, respectively).

5.1.4 Word2Vec

To implement word2vec, we needed pre-trained word embeddings. Each word in the embedding (Google-news-300) we used is represented as a three hundred (300) dimensional vector. Finally, all documents were tokenized using the *Tokenizer* from keras (keras.preprocessing.text), and padded using *pad_sequences*from keras (keras_preprocessing.sequence). By padding all documents, we ensured that all the documents are of the same size. As with the previous models, we explore the performance of word2vec in respect to the top documents that were returned as similar documents.

Figure 5 shows the references in the paper “Uni-Perceiver-MoE: Learning Sparse Generalist Models with Conditional MoEs”. After running the Word2Vec model, the top hundred (100) similar papers to the input paper are shown below:

```
[41]: models_performance("word2vec")

CALCULATING AI PAPERS
Paper: Experimenting_with_Self-Supervision_using_Rotation_Prediction_for_Image
__Captioning.txt 0.9270268186102184
Paper: NLX-GPT:_A_Model_for_Natural_Language_Explanations_in_Vision_and
__Vision-Language_Tasks.txt 0.9262714738529939
Paper: A_Generative_Language_Model_for_Few-shot_Aspect-Based_Sentiment_Analysis.txt 0.922709984145486
Paper: One-shot_Scene_Graph_Generation.txt 0.9209182472093829
Paper: SBNet:_Segmentation-based_Network_for_Natural_Language-based_Vehicle
__Search.txt 0.9197421699012296
Paper: Heterogeneous_Multi-task_Learning_for_Human_Pose_Estimation_with_Deep
__Convolutional_Neural_Network.txt 0.918108946676329
Paper: SurgeonAssist-Net:_Towards_Context-Aware_Head-Mounted_Display-Based
__Augmented_Reality_for_Surgical_Guidance.txt 0.9159138385766483
Paper: Conceptual 12M- Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts.txt 0.9109779603793144
Paper: Task-Oriented_Dialogue_System_as_Natural_Language_Generation.txt 0.9097139882812167
Paper: UNITER- UNiversal Image-TExt Representation Learning.txt 0.909588430803436
Paper: EyeNet:_A_Multi-Task_Network_for_Off-Axis_Eye_Gaze_Estimation_and_User
__Understanding.txt 0.9094440944826658
Paper: HumanMeshNet:_Polygonal_Mesh_Recovery_of_Humans.txt 0.9079961295967669
Paper: SiT:_Self-supervised_vIsion_Transformer.txt 0.9064557958345905
Paper: Enabling_Robots_to_Draw_and_Tell:_Towards_Visually_Grounded_Multimodal
__Description_Generation.txt 0.9059005726360057
Paper: On_Hiding_Neural_Networks_Inside_Neural_Networks.txt 0.9058175528695807
Paper: Molecular_representation_learning_with_language_models_and
__domain-relevant_auxiliary_tasks.txt 0.8991834210228882
Paper: An_Open-Source_Dataset_and_A_Multi-Task_Model_for_Malay_Named_Entity
__Recognition.txt 0.8990974593841861
Paper: Generative_Prior_Knowledge_for_Discriminative_Classification.txt 0.8985919537252216
Paper: Context_based_lemmatizer_for_Polish_language.txt 0.8975642918305944
Paper: Beneficial_Perturbation_Network_for_designing_general_adaptive
__artificial_intelligence_systems.txt 0.897441105346088
Paper: Multitask Learning.txt 0.8957659084075007
Paper: NeurIPS-2020-the-lottery-ticket-hypothesis-for-pre-trained-bert-networks-Paper.txt 0.8952017900987779
Paper: Making_the_Most_of_Text_Semantics_to_Improve_Biomedical_Vision--Language
__Processing.txt 0.8945695391746351
Paper: Making_Pre-trained_Language_Models_End-to-end_Few-shot_Learners_with
__Contrastive_Prompt_Tuning.txt 0.8944260481725571
Paper: Surgical_Visual_Domain_Adaptation:_Results_from_the_MICCAI_2020
__SurgVisDom_Challenge.txt 0.8940383704081171
Paper: Grounding_Natural_Language_Instructions:_Can_Large_Language_Models
__Capture_Spatial_Information?.txt 0.894020940448087
Paper: BAM!_Born-Again_Multi-Task_Networks_for_Natural_Language_Understanding.txt 0.8939858277855423
Paper: BAM! Born-Again Multi-Task Networks for Natural Language Understanding.txt 0.8939858277855423
Paper: Recurrent_Models_of_Visual_Attention.txt 0.8939552795769514
Paper: A_Generalizable_Approach_to_Learning_Optimizers.txt 0.8933992742294001
Paper: Pseudo-Recursal:_Solving_the_Catastrophic_Forgetting_Problem_in_Deep
__Neural_Networks.txt 0.8930975259913028
Paper: Masked-attention Mask Transformer for Universal Image Segmentation.txt 0.8920986639286603
Paper: Language Models are Few-Shot Learners.txt 0.8918804493288656
Paper: Measuring_Information_Transfer_in_Neural_Networks.txt 0.8917329905509203
Paper: Classification_of_Long_Sequential_Data_using_Circular_Dilated
__Convolutional_Neural_Networks.txt 0.8915939407291049
Paper: Multi-Task_Trust_Transfer_for_Human-Robot_Interaction.txt 0.8913411806910538
Paper: Localizing_Catastrophic_Forgetting_in_Neural_Networks.txt 0.8908444395513595
Paper: Defense against Adversarial Attacks Using High-Level Representation Guided Denoiser.txt 0.8887456273556296
Paper: Multi-task_learning_for_virtual_flow_metering.txt 0.888544589194417
Paper: Self-Training_Vision_Language_BERTs_with_a_Unified_Conditional_Model.txt 0.8880236507832007
Paper: NaRLE:_Natural_Language_Models_using_Reinforcement_Learning_with_Emotion
__Feedback.txt 0.8879306646469777
Paper: GAIA:_A_Transfer_Learning_System_of_Object_Detection_that_Fits_Your
__Needs.txt 0.8878531415468172
Paper: RoboTurk:_A_Crowdsourcing_Platform_for_Robotic_Skill_Learning_through
```

```
__Needs.txt 0.8878531415468172
Paper: RoboTurk:_A_Crowdsourcing_Platform_for_Robotic_Skill_Learning_through
__Imitation.txt 0.8871315918756308
Paper: Transfer_Learning_from_LDA_to_BiLSTM-CNN_for_Offensive_Language
__Detection_in_Twitter.txt 0.8867801074156936
Paper: Autonomous_Driving_in_Reality_with_Reinforcement_Learning_and_Image
__Translation.txt 0.8864526554523449
Paper: Value_Iteration_Networks.txt 0.8861059869257285
Paper: Self-supervised_Auxiliary_Learning_for_Graph_Neural_Networks_via
__Meta-Learning.txt 0.8855999464440218
Paper: Beyond_Mono_to_Binaural:_Generating_Binaural_Audio_from_Mono_Audio_with
__Depth_and_Cross_Modal_Attention.txt 0.8855265850985605
Paper: Cranial_Implant_Design_via_Virtual_Craniectomy_with_Shape_Priors.txt 0.8850768025796373
Paper: Deductive_Association_Networks.txt 0.8847077835787548
Paper: Learning_Hierarchical_Information_Flow_with_Recurrent_Neural_Modules.txt 0.8842826155670069
Paper: SINGA-Easy:_An_Easy-to-Use_Framework_for_MultiModal_Analysis.txt 0.8825326188871228
Paper: Neural_Module_Networks.txt 0.8824291286854204
Paper: Image-based_Natural_Language_Understanding_Using_2D_Convolutional_Neural
__Networks.txt 0.8820513624792988
Paper: FRAGE:_Frequency-Agnostic_Word_Representation.txt 0.8813498455236808
Paper: Exploring_Software_Naturalness_through_Neural_Language_Models.txt 0.8794873451863382
Paper: Object-aware_Video-language_Pre-training_for_Retrieval.txt 0.8792289201606133
Paper: Smart-PGSim:_Using_Neural_Network_to_Accelerate_AC-OPF_Power_Grid
__Simulation.txt 0.8791911049264554
Paper: Visual_Re-ranking_with_Natural_Language_Understanding_for_Text_Spotting.txt 0.8784193349278216
Paper: Explaining_Chest_X-ray_Pathologies_in_Natural_Language.txt 0.8782356833094944
Paper: An_Exploration_of_Prompt_Tuning_on_Generative_Spoken_Language_Model_for
__Speech_Processing_Tasks.txt 0.8778209248022131
Paper: Semantic_speech_retrieval_with_a_visually_grounded_model_of
__untranscribed_speech.txt 0.8776975018247843
Paper: A Multi-Level Typology of Abstract Visualization Tasks.txt 0.8771159058402062
Paper: Stochastic_reconstruction_of_an_oolitic_limestone_by_generative
__adversarial_networks.txt 0.8756425168098796
Paper: LightRel_SemEval-2018_Task_7:_Lightweight_and_Fast_Relation
__Classification.txt 0.8755892755484118
Paper: Perspective_Taking_in_Deep_Reinforcement_Learning_Agents.txt 0.8752969141852238
Paper: Can_neural_networks_learn_persistent_homology_features?.txt 0.8752234827476078
Paper: Lip Movements Generation at a Glance.txt 0.8742836685631752
Paper: Neural_network_gradient-based_learning_of_black-box_function_interfaces.txt 0.8742827172561027
Paper: HOUDINI:_Lifelong_Learning_as_Program_Synthesis.txt 0.8739699032245586
Paper: SOFT:_Softmax-free_Transformer_with_Linear_Complexity.txt 0.8739218715201905
Paper: Human_Visual_Attention_Prediction_Boosts_Learning_&_Performance_of
__Autonomous_Driving_Agents.txt 0.8738366613548284
Paper: Unsupervised_Pre-Training_on_Patient_Population_Graphs_for_Patient-Level
__Predictions.txt 0.8736136553644019
Paper: Neural_Task_Representations_as_Weak_Supervision_for_Model_Agnostic
__Cross-Lingual_Transfer.txt 0.8735814820738339
Paper: Continuous_Operator_Authentication_for_Teleoperated_Systems_Using_Hidden
__Markov_Models.txt 0.8735345339068392
Paper: NeurIPS-2021-vatt-transformers-for-multimodal-self-supervised-learning-from-raw-video-audio-and-text-Paper.txt 0.8734970069945344
Paper: Maximum-Margin_Structured_Learning_with_Deep_Networks_for_3D_Human_Pose
__Estimation.txt 0.8734319716480934
Paper: What_can_we_learn_from_Semantic_Tagging?.txt 0.8733421711653717
Paper: Security_of_Deep_Learning_Methodologies:_Challenges_and_Opportunities.txt 0.8726455519380564
Paper: MemBERT:_Injecting_Unstructured_Knowledge_into_BERT.txt 0.8725593439353532
Paper: Natural_Language_Object_Retrieval.txt 0.8723116660708472
Paper: Evolution_of_transfer_learning_in_natural_language_processing.txt 0.8716786585002856
Paper: Neural_Image_Compression_for_Gigapixel_Histopathology_Image_Analysis.txt 0.8716544478743936
Paper: Visual_Question_Answering_as_Reading_Comprehension.txt 0.8715119546620256
Paper: Multi-sense_Definition_Modeling_using_Word_Sense_Decompositions.txt 0.8713958619560249
Paper: Discriminative_Neural_Topic_Models.txt 0.8705661600472814
Paper: e-SNLI:_Natural_Language_Inference_with_Natural_Language_Explanations.txt 0.8704585525599065
Paper: Generic_Neural_Architecture_Search_via_Regression.txt 0.8700353109675467
Paper: Boosting_Neural_Image_Compression_for_Machines_Using_Latent_Space
```

Figure 12. Top 100 Similar Documents for Uni-Perceiver-MoE: Learning Sparse Generalist Models with Conditional MoEs Using Word2Vec

As with the previous sections, we show the references in our paper that are part of the top fifty (50), and hundred (100) similar documents as predicted by our Word2Vec model. In each figure, we show the reference and its similarity score to the input paper. Finally, we calculate the percentage of references accurately predicted.

```
[24]: models_performance("word2vec")

CALCULATING AI PAPERS
Paper: Conceptual 12M- Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts.txt 0.9109779603793144
Paper: UNITER- UNiversal Image-TExt Representation Learning.txt 0.909588430803436
Paper: BAM! Born-Again Multi-Task Networks for Natural Language Understanding.txt 0.8939858277855423
Paper: Masked-attention Mask Transformer for Universal Image Segmentation.txt 0.8920986639286603
Paper: NeurIPS-2021-vatt-transformers-for-multimodal-self-supervised-learning-from-raw-video-audio-and-text-Paper.txt 0.8734970069945344
0.29411764705882354
```

Figure 13. Word2Vec Performance on Sample I

```
models_performance("word2vec")

CALCULATING AI PAPERS
Paper: Conceptual 12M- Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts.txt 0.9109779603793144
Paper: UNITER- UNiversal Image-TExt Representation Learning.txt 0.909588430803436
Paper: BAM! Born-Again Multi-Task Networks for Natural Language Understanding.txt 0.8939858277855423
Paper: Masked-attention Mask Transformer for Universal Image Segmentation.txt 0.8920986639286603
0.23529411764705882
```

Figure 14: Word2Vec Performance on Sample Document II

As shown above, our Word2Vec model predicted 23.53% of the references as part of the top fifty (50), and 29.41% when considering the top hundred (100) similar documents, thus producing the highest accuracy on the sample document. Below is a table of the results for the Word2Vec model, for all the test documents.

Table 13

Word2Vec Model Performance on Corpus

Paper	*Category*	*Score (Top 50)*	*Score (Top 100)*
Functional Code Building Genetic Programming	AI	25%	31.25%
TwiBot-22: Towards Graph-Based Twitter Bot Detection	AI	0%	11.11%
Jewelry Shop Conversational Chatbot	AI	0%	0%
Uni-Perceiver-MoE: Learning Sparse Generalist Models with Conditional MoEs	AI	23.53%	29.41%
Visualization in virtual reality: a systematic review	VR	5%	5%
Joint Compute-Caching-Communication Control for Online Data-Intensive Service Delivery	VR	47.37%	52.63%

Paper	*Category*	*Score (Top 50)*	*Score (Top 100)*
6G Survey on Challenges, Requirements, Applications, Key Enabling Technologies, Use Cases, AI integration issues and Security aspects	VR	6.67%	13.33%
Quantifying the Effects of Working in VR for One Week	VR	22.22%	27.78%
Neo-GNNs: Neighborhood Overlap-aware Graph Neural Networks for Link Prediction	NN	5%	5%
Learning Vehicle Trajectory Uncertainty	NN	5%	10%
Early Transferability of Adversarial Examples in Deep Neural Networks	NN	0%	0%
Face-Dubbing++: Lip-Synchronous, Voice Preserving Translation of Videos	NN	22.22%	22.22%
NLU for Game-based Learning in Real: Initial Evaluations	NLP	6.25%	6.25%

Paper	*Category*	*Score (Top 50)*	*Score (Top 100)*
Multi-Agent Reinforcement Learning is A Sequence Modeling Problem	NLP	17.65%	17.65%
Differentially Private Model Compression	NLP	6.25%	6.25%
Quantum Neural Network Classifiers: A Tutorial	NLP	0%	0%

Based on Table 13, we can infer that like TF-IDF and Doc2Vec, the category with the highest average score is VR, with an average score of 20.32% for the top 50 papers and 24.69% for the top 100 papers. This is higher than the average scores for the other categories, which are AI (12.13% and 17.94% for the top 50 and top 100, respectively), NN (8.01% and 9.25% for the top 50 and top 100, respectively), and NLP (7.5% and 7.5% for the top 50 and top 100 papers, respectively).

5.1.5 GloVe

We implemented the GloVe model using the word embeddings provided by GloVe and Tokenizer from keras. While TF-IDF does not keep the original sequence of words, we ensured to maintain the sequence of words from the documents to ensure optimal performance by the model. The embeddings were represented as a one hundred (100) dimension vector.

Figure 5 shows the references in the paper “Uni-Perceiver-MoE: Learning Sparse Generalist Models with Conditional MoEs”. After running the GloVe model, the top hundred (100) similar papers to the input paper are shown below:

```
[42]: models_performance("glove")

CALCULATING AI PAPERS
Paper: SiT:_Self-supervised_vIsion_Transformer.txt 0.980504545046666
Paper: SurgeonAssist-Net:_Towards_Context-Aware_Head-Mounted_Display-Based
__Augmented_Reality_for_Surgical_Guidance.txt 0.979361835320573
Paper: Task-Oriented_Dialogue_System_as_Natural_Language_Generation.txt 0.9775188313584233
Paper: Molecular_representation_learning_with_language_models_and
__domain-relevant_auxiliary_tasks.txt 0.9774133484502859
Paper: Conceptual 12M- Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts.txt 0.977224644447225
Paper: Measuring_Information_Transfer_in_Neural_Networks.txt 0.9769856061252906
Paper: GAIA:_A_Transfer_Learning_System_of_Object_Detection_that_Fits_Your
__Needs.txt 0.9764925581006045
Paper: Experimenting_with_Self-Supervision_using_Rotation_Prediction_for_Image
__Captioning.txt 0.9763229762135479
Paper: An_Open-Source_Dataset_and_A_Multi-Task_Model_for_Malay_Named_Entity
__Recognition.txt 0.9757302118273986
Paper: One-shot_Scene_Graph_Generation.txt 0.9740280765971709
Paper: NaRLE:_Natural_Language_Models_using_Reinforcement_Learning_with_Emotion
__Feedback.txt 0.9739835573036693
Paper: Autonomous_Driving_in_Reality_with_Reinforcement_Learning_and_Image
__Translation.txt 0.9739125361336841
Paper: Cranial_Implant_Design_via_Virtual_Craniectomy_with_Shape_Priors.txt 0.9737634663094413
Paper: Surgical_Visual_Domain_Adaptation:_Results_from_the_MICCAI_2020
__SurgVisDom_Challenge.txt 0.9735489425773449
Paper: HumanMeshNet:_Polygonal_Mesh_Recovery_of_Humans.txt 0.9731336967967703
Paper: NeurIPS-2021-vatt-transformers-for-multimodal-self-supervised-learning-from-raw-video-audio-and-text-Paper.txt 0.9730943262988445
Paper: UNITER- UNiversal Image-TExt Representation Learning.txt 0.9730517041992804
Paper: A_Generalizable_Approach_to_Learning_Optimizers.txt 0.9729868783652584
Paper: A_Generative_Language_Model_for_Few-shot_Aspect-Based_Sentiment_Analysis.txt 0.9722060202915214
Paper: Localizing_Catastrophic_Forgetting_in_Neural_Networks.txt 0.97210419113299
Paper: Classification_of_Long_Sequential_Data_using_Circular_Dilated
__Convolutional_Neural_Networks.txt 0.9712592824140721
Paper: Making_the_Most_of_Text_Semantics_to_Improve_Biomedical_Vision--Language
__Processing.txt 0.971241036428477
Paper: NLX-GPT:_A_Model_for_Natural_Language_Explanations_in_Vision_and
__Vision-Language_Tasks.txt 0.971136684469851
Paper: Security_of_Deep_Learning_Methodologies:_Challenges_and_Opportunities.txt 0.971115272625083
Paper: EyeNet:_A_Multi-Task_Network_for_Off-Axis_Eye_Gaze_Estimation_and_User
__Understanding.txt 0.9711069642547697
Paper: FRAGE:_Frequency-Agnostic_Word_Representation.txt 0.9710259135677843
Paper: On_Hiding_Neural_Networks_Inside_Neural_Networks.txt 0.9708714792329485
Paper: SBNet:_Segmentation-based_Network_for_Natural_Language-based_Vehicle
__Search.txt 0.970724988057915
Paper: Defense against Adversarial Attacks Using High-Level Representation Guided Denoiser.txt 0.9706726544145086
Paper: Multi-task_learning_for_virtual_flow_metering.txt 0.9705239557297112
Paper: RoboTurk:_A_Crowdsourcing_Platform_for_Robotic_Skill_Learning_through
__Imitation.txt 0.9703781269642964
Paper: SINGA-Easy:_An_Easy-to-Use_Framework_for_MultiModal_Analysis.txt 0.9699974524426929
Paper: Context_based_lemmatizer_for_Polish_language.txt 0.969994447047087
Paper: Making_Pre-trained_Language_Models_End-to-end_Few-shot_Learners_with
__Contrastive_Prompt_Tuning.txt 0.9699873168648214
Paper: Heterogeneous_Multi-task_Learning_for_Human_Pose_Estimation_with_Deep
__Convolutional_Neural_Network.txt 0.9698348283014425
Paper: Exploring_Software_Naturalness_through_Neural_Language_Models.txt 0.9697846887070123
Paper: Hyperparameter-Free_Losses_for_Model-Based_Monocular_Reconstruction.txt 0.9697439971658725
Paper: Language Models are Few-Shot Learners.txt 0.9695189444742971
Paper: An_Encoder-Decoder_Based_Audio_Captioning_System_With_Transfer_and
__Reinforcement_Learning.txt 0.9693394920304368
Paper: NeurIPS-2020-the-lottery-ticket-hypothesis-for-pre-trained-bert-networks-Paper.txt 0.9691035469593822
```

```
Paper: NeurIPS-2020-the-lottery-ticket-hypothesis-for-pre-trained-bert-networks-Paper.txt 0.9691035469593822
Paper: SelfHAR:_Improving_Human_Activity_Recognition_through_Self-training_with
__Unlabeled_Data.txt 0.9687007735649017
Paper: Image-based_Natural_Language_Understanding_Using_2D_Convolutional_Neural
__Networks.txt 0.9685196458515638
Paper: Smart-PGSim:_Using_Neural_Network_to_Accelerate_AC-OPF_Power_Grid
__Simulation.txt 0.968340168248197
Paper: Self-Training_Vision_Language_BERTs_with_a_Unified_Conditional_Model.txt 0.9682581697404404
Paper: Multitask Learning.txt 0.9681203138396189
Paper: Visual_Question_Answering_for_Cultural_Heritage.txt 0.9680875065870309
Paper: MemBERT:_Injecting_Unstructured_Knowledge_into_BERT.txt 0.9680529909376943
Paper: Neural_Task_Representations_as_Weak_Supervision_for_Model_Agnostic
__Cross-Lingual_Transfer.txt 0.9678884768646412
Paper: CL4AC:_A_Contrastive_Loss_for_Audio_Captioning.txt 0.9678707568770886
Paper: Unsupervised_Pre-Training_on_Patient_Population_Graphs_for_Patient-Level
__Predictions.txt 0.967743547016064
Paper: Secure_Watermark_for_Deep_Neural_Networks_with_Multi-task_Learning.txt 0.9676992740978759
Paper: Using_Natural_Language_Processing_to_Develop_an_Automated_Orthodontic
__Diagnostic_System.txt 0.9676763661425333
Paper: A_Review_of_Emerging_Research_Directions_in_Abstract_Visual_Reasoning.txt 0.9675168264005688
Paper: Toward_Improving_Attentive_Neural_Networks_in_Legal_Text_Processing.txt 0.9673985042139338
Paper: Visual_Re-ranking_with_Natural_Language_Understanding_for_Text_Spotting.txt 0.9671689065494158
Paper: Multi-Task_Trust_Transfer_for_Human-Robot_Interaction.txt 0.967046450497281
Paper: Masked Autoencoders Are Scalable Vision Learners.txt 0.9670393680752684
Paper: HealthPrompt:_A_Zero-shot_Learning_Paradigm_for_Clinical_Natural
__Language_Processing.txt 0.9670177951311582
Paper: Learning_Hierarchical_Information_Flow_with_Recurrent_Neural_Modules.txt 0.9670141949026319
Paper: VRGym:_A_Virtual_Testbed_for_Physical_and_Interactive_AI.txt 0.9669869786264105
Paper: Self-supervised_Auxiliary_Learning_for_Graph_Neural_Networks_via
__Meta-Learning.txt 0.9667034358817674
Paper: Decision Transformer- Reinforcement Learning via Sequence Modeling.txt 0.9666728045721696
Paper: Neural_network_gradient-based_learning_of_black-box_function_interfaces.txt 0.9665724671156128
Paper: De-rendering_3D_Objects_in_the_Wild.txt 0.9665409663522805
Paper: Discriminative_Neural_Topic_Models.txt 0.9664610776279319
Paper: Efficient_2.5D_Hand_Pose_Estimation_via_Auxiliary_Multi-Task_Training
__for_Embedded_Devices.txt 0.9663563505157008
Paper: V2W-BERT:_A_Framework_for_Effective_Hierarchical_Multiclass
__Classification_of_Software_Vulnerabilities.txt 0.9662437196070948
Paper: An_Exploration_of_Prompt_Tuning_on_Generative_Spoken_Language_Model_for
__Speech_Processing_Tasks.txt 0.9661420658810969
Paper: What_can_we_learn_from_Semantic_Tagging?.txt 0.9660660882479185
Paper: Human_Visual_Attention_Prediction_Boosts_Learning_&_Performance_of
__Autonomous_Driving_Agents.txt 0.96603847665228
Paper: EmbRace:_Accelerating_Sparse_Communication_for_Distributed_Training_of
__NLP_Neural_Networks.txt 0.9660265539922613
Paper: Data_Augmentation_for_Voice-Assistant_NLU_using_BERT-based
__Interchangeable_Rephrase.txt 0.9660248860712031
Paper: Generic_Neural_Architecture_Search_via_Regression.txt 0.9659427324180921
Paper: Generative_Prior_Knowledge_for_Discriminative_Classification.txt 0.9658969135067217
Paper: Masked-attention Mask Transformer for Universal Image Segmentation.txt 0.9658634202249904
Paper: Extracting Training Data from Large Language Models.txt 0.9657998652162479
Paper: Image_Quality_Assessment_Guided_Deep_Neural_Networks_Training.txt 0.9655778368970844
Paper: Multi-Agent Reinforcement Learning is
A Sequence Modeling Problem.txt 0.9655647318380731
Paper: Towards_Lifelong_Learning_of_End-to-end_ASR.txt 0.9654847087951252
Paper: Object-aware_Video-language_Pre-training_for_Retrieval.txt 0.965388497926978
Paper: IL-Net:_Using_Expert_Knowledge_to_Guide_the_Design_of_Furcated_Neural
__Networks.txt 0.965370867096292
Paper: Transfer_Learning_for_Improving_Results_on_Russian_Sentiment_Datasets.txt 0.9653020550466771
Paper: Self-supervised_U-net_for_few-shot_learning_of_object_segmentation_in
__microscopy_images.txt 0.9652550873770402
Paper: Deductive_Association_Networks.txt 0.9651849133695054
Paper: Natural_language_understanding_for_task_oriented_dialog_in_the
__biomedical_domain_in_a_low_resources_context.txt 0.965152630798705
```

Figure 15. Top 100 Similar Documents for Uni-Perceiver-MoE: Learning Sparse Generalist Models with Conditional MoEs Using GloVe

As with the previous sections, we show the references in our paper that are part of the top fifty (50), and hundred (100) similar documents as predicted by our GloVe model.

In each figure, we show the reference and its similarity score to the input paper. Finally, we calculate the percentage of references accurately predicted.

```
[25]: models_performance("glove")

CALCULATING AI PAPERS
Paper: Conceptual 12M- Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts.txt 0.977224644447225
Paper: NeurIPS-2021-vatt-transformers-for-multimodal-self-supervised-learning-from-raw-video-audio-and-text-Paper.txt 0.9730943262988445
Paper: UNITER- UNiversal Image-TExt Representation Learning.txt 0.9730517041992804
Paper: Masked-attention Mask Transformer for Universal Image Segmentation.txt 0.9658634202249904
0.23529411764705882
```

Figure 16. GloVe Performance on Sample Document I

```
: models_performance("glove")

CALCULATING AI PAPERS
Paper: Conceptual 12M- Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts.txt 0.977224644447225
Paper: NeurIPS-2021-vatt-transformers-for-multimodal-self-supervised-learning-from-raw-video-audio-and-text-Paper.txt 0.9730943262988445
Paper: UNITER- UNiversal Image-TExt Representation Learning.txt 0.9730517041992804
0.17647058823529413
```

Figure 17. GloVe Performance on Sample Document II

As shown above, our GloVe model predicted 17.65% of the references as part of the top fifty (50), and 23.53% when considering the top hundred (100) similar documents. Below is a table of the results for the GloVe model, for all the test documents.

Table 14

GloVe Model Performance on Corpus

Paper	*Category*	*Score (Top 50)*	*Score (Top 100)*
Functional Code Building Genetic Programming	AI	12.5%	31.25%
TwiBot-22: Towards Graph-Based Twitter Bot Detection	AI	0%	5.56%
Jewelry Shop Conversational Chatbot	AI	0%	0%
Uni-Perceiver-MoE: Learning Sparse Generalist Models with Conditional MoEs	AI	17.65%	23.53%
Visualization in virtual reality: a systematic review	VR	5%	5%
Joint Compute-Caching-Communication Control for Online Data-Intensive Service Delivery	VR	36.84%	52.63%

Paper	*Category*	*Score (Top 50)*	*Score (Top 100)*
6G Survey on Challenges, Requirements, Applications, Key Enabling Technologies, Use Cases, AI integration issues and Security aspects	VR	6.67%	6.67%
Quantifying the Effects of Working in VR for One Week	VR	16.67%	44.44%
Neo-GNNs: Neighborhood Overlap-aware Graph Neural Networks for Link Prediction	NN	5%	5%
Learning Vehicle Trajectory Uncertainty	NN	5%	10%
Early Transferability of Adversarial Examples in Deep Neural Networks	NN	0%	0%
Face-Dubbing++: Lip-Synchronous, Voice Preserving Translation of Videos	NN	16.67%	16.67%
NLU for Game-based Learning in Real: Initial Evaluations	NLP	18.75%	18.75%

Paper	*Category*	*Score (Top 50)*	*Score (Top 100)*
Multi-Agent Reinforcement Learning is A Sequence Modeling Problem	NLP	11.76%	29.41%
Differentially Private Model Compression	NLP	6.25%	12.5%
Quantum Neural Network Classifiers: A Tutorial	NLP	0%	5.56%

Based on Table 14, the category with the highest average score is VR, with an average score of 16.30% for the top 50 papers and 27.19% for the top 100 papers. This is higher than the average scores for the oth.er categories, which are NLP (9.19% and 16.55% for the top 50 and top 100, respectively), AI (7.53% and 15.09% for the top 50 and top 100, respectively), and NN (6.67% and 7.91% for the top 50 and top 100 papers, respectively).

5.2 Enhanced Model Evaluations

Here, we implement a new algorithm for calculating the similarity of the documents by splitting the documents into smaller chunks, comparing the smaller chunks, and assigning the maximum similarity between the chunks as the similarity between the two documents. This implements the formula:

sim(doc1, doc2) =max(sim(doc1_1,doc2_1), sim(doc1_1,doc2_2),...,(sim(doc1_n, doc2_n))

where doc1_1 and doc2_1 are chunks of document 1 and document 2. As shown in the example below, we see an improvement in the performance of the models when subsections of the documents are used to determine the similarity score.

Taking one of our input documents as (***Functional Code Building Genetic Programing***) as X, and one if its referenced papers (***Lexicase Selection of Specialists***) as Y, we compare the similarity score between both papers using TF-IDF when comparing the whole document, and TF-IDF when we compare subsections of the papers. In the first attempt, the TF-IDF model was executed with stopwords removal. The resulting similarity score was 0.7872. In order of ranking, it was the fifth most similar paper in the available references for the input paper X. Finally, we compared both papers using their subsections. Each paper was divided into 15 subsections. Other sizes attempted were ten(10), twenty(20), and twenty-five (25). We found the similarity score between the two papers, X and Y, to have increased from 0.7872 to 0.9681. We then proceeded to inspect the areas of both documents that were marked as most similar. First, section 14 on both documents were the most similar. Giving us 0.9681. Inspecting the cleaned version of the papers gives the section(s) as follows:

From Y:

"e thomas helmuth lee spector comparison lin ear genome representations software synthesis genetic programming theory practice xvii wolfgang banzhaf erik goodman leigh sheneman leonardo trujillo bill worzel eds springer east lansing mi usa https doi org doi edward pantridge lee spector code building genetic programming proceedings genetic evolutionary computation conference riccardo poli william b langdon nicholas freitag mcphee field guide genetic programming published via http lulu com freely avail able http www gp field guide org uk with

contributions j r koza fran ois pottier type inference presence subtyping theory practice research report rr inria https hal inria fr inria john alan robinson machine oriented logic based resolution principle journal acm jacm geoffrey seward smith polymorphic type inference languages overloading subtyping ph d dissertation usa umi order no gax dominik sobania generalizability programs synthesized grammar guided genetic programming eurogp proceedings th european conference genetic programming lncs vol ting hu nuno lourenco eric medvet eds springer verlag virtual event https doi org doi dominik sobania dirk schweim franz rothlauf recent develop ments program synthesis evolutionary algorithms arxiv preprint arxiv lee spector jon klein andmaartenkeijzer push execution stack evolution control https doi org https doi org inco http www jstor org stable https doi org https doi org https doi org https doi org https doi org https doi org https doi org isal a https doi org https doi org doi tevc https doi org https doi org https doi org https doi org doi https hal inria fr inria https doi org doi https doi org abstract introduction code building genetic programming tools type theory types unification functional code building gp genomes compilation ast evolution simplification experimental design comparison methods results example solution programs discussion future work conclusion acknowledgments re"

and from paper X:

"lee spector jon klein maarten keijzer push execution stack evolution control gecco proceedings conference genetic evolutionary computation vol acm press washington dc usa https doi org lee spector william la cava saul shanabrook thomas helmuth edward pantridge relaxations lexicase parent selection ingenetic programming theory practice xv wolfgang banzhaf randal s olson william tozier rick riolo eds springer international publishing cham lee spector alan robinson genetic programming autocon structive evolution push programming language genetic program ming evolvable machines march https doi org a https doi org https doi org https doi org https doi org https doi org https doi org https doi org https doi org https doi org https doi org tevc https web cs umass edu publication docs um cs phd pdf https web cs umass edu publication docs um cs phd pdf https doi org https doi org http www springer com us book https doi org https doi org https doi org ecal a https doi org ecal a https doi org https doi org https doi org http arxiv org abs http arxiv org abs http arxiv org abs https doi org https doi org https doi org https doi org a https doi org a erratum notice publication came attention errors data presented figure errors corrected figure pdf corrections influence discussion presented text therefore text changed originally published incorrect version figure found below string lengths backwards syllables vector average x word lines last index of zero mirror image negative to zero replace space with newline percent training cases used selection e n si ty abstract introduction background lexicase selection specialists genetic programming experimental design benchmark problems push pushgp specialists tournament selection specialists lexicase selection importance selecting specialists conclusions acknowledgments refe".

The sections from the cleaned version as shown above do not give convincing context into why they were the most similar sections. A look into the sections of the documents in their original state (uncleaned) revealed that this was the citation section on both papers.

Another example of a similar performance is when we consider the paper(X): ***Quantum Neural Network Classifiers: A Tutorial in natural language processing***. This paper, when compared to others using TF-IDF has a 0% similarity match with any of the papers referenced in the corpus. However, when compared using subsections of the document, we find a 0.9339 match with paper Y (***A rigorous and robust quantum speed-up in supervised machine learning***). From the unclean version of the documents, the matching subsections were section 13 from X, and section 2 from Y. The text for each of the subsections are shown below.

From paper X:

```
"6] X.-Z. Luo, J.-G. Liu, P. Zhang and L. Wang, Yao.jl: Extensible,
Efficient Framework for Quantum
Algorithm Design, Quantum 4, 341 (2020), doi:10.22331/q-2020-10-11-341.
[57] J. Bezanson, A. Edelman, S. Karpinski and V. B. Shah, Julia: A Fresh
Approach to Numerical
Computing, SIAM Rev. 59(1), 65 (2017), doi:10.1137/141000671.

[58] M. Broughton, G. Verdon, T. McCourt, A. J. Martinez, J. H. Yoo, S.
V. Isakov, P. Massey,
M. Y. Niu, R. Halavati, E. Peters, M. Leib, A. Skolik et al., TensorFlow
Quantum: A Software
Framework for Quantum Machine Learning, URL
https://arxiv.org/abs/2003.02989 (2020).
22
https://doi.org/10.22331/q-2021-09-09-539
https://arxiv.org/abs/2103.16774
https://doi.org/10.1103/PhysRevA.103.032430
https://arxiv.org/abs/2106.03880
https://doi.org/10.1103/PhysRevResearch.3.L032049
https://doi.org/10.22331/q-2021-03-29-422
https://doi.org/10.1103/PRXQuantum.2.040321
https://doi.org/10.1103/PhysRevLett.128.080506
https://arxiv.org/abs/2007.12369
https://doi.org/10.1103/PRXQuantum.2.040309
https://doi.org/10.22331/q-2018-08-06-79
https://doi.org/10.1103/RevModPhys.94.015004
https://doi.org/10.22331/q-2020-10-11-341
```

https://doi.org/10.1137/141000671
https://arxiv.org/abs/2003.02989
REFERENCES Submission
[59] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, M. S. Alam, S. Ahmed, J. M. Arrazola, C. Blank,
A. Delgado, S. Jahangiri, K. McKiernan, J. J. Meyer et al., PennyLane: Automatic differ-
entiation of hybrid quantum-classical computations, URL https://arxiv.org/abs/1811.04968
(2020).
[60] N. Killoran, J. Izaac, N. Quesada, V. Bergholm, M. Amy and C. Weedbrook, Strawberry
Fields: A Software Platform for Photonic Quantum Computing, Quantum 3, 129 (2019),
doi:10.22331/q-2019-03-11-129.
[61] G. Aleksandrowicz, T. Alexander, P. Barkoutsos, L. Bello, Y. Ben-Haim, D. Bucher, F. J.
Cabrera-Hernández, J. Carballo-Franquis, A. Chen, C.-F. Chen, J. M. Chow, A. D. Córcoles-
Gonzales et al., Qiskit: An Open-source Framework for Quantum Computing, Zenodo,
doi:10.5281/zenodo.2562111 (2019).
[62] K. Svore, A. Geller, M. Troyer, J. Azariah, C. Granade, B. Heim, V. Kliuchnikov, M. Mykhailova,
A. Paz and M. Roetteler, Q#: Enabling Scalable Quantum Computing and Development with a
High-level DSL, In Proceedings of the Real World Domain Specific Languages Workshop 2018,
RWDSL2018, pp. 1–10. Association for Computing Machinery, New York, NY, USA, ISBN
978-1-4503-6355-6, doi:10.1145/3183895.3183901 (2018).
[63] F. Zhang, C. Huang, M. Newman, J. Cai, H. Yu, Z. Tian, B. Yuan, H. Xu, J. Wu, X. Gao,
J. Chen, M. Szegedy et al., Alibaba Cloud Quantum Development Platform: Large-Scale Clas-
sical Simulation of Quantum Circuits, URL https://arxiv.org/abs/1907.11217 (2019).
[64] C. Huang, M. Szegedy, F. Zhang, X. Gao, J. Chen and Y. Shi, Alibaba Cloud Quantum De-
velopment Platform: Applications to Quantum Algorithm Design, URL https://arxiv.org/abs/
1909.02559 (2019).
[65] D. Nguyen, D. Mikushin and Y. Man-Hong, HiQ-ProjectQ: Towards user-friendly
and high-performance quantum computing on GPUs, In 2021 Design, Automa-
tion Test in Europe Conference Exhibition (DATE), pp. 1056–1061. Grenoble, France,
doi:10.23919/DATE51398.2021.9474170 (2021).
[66] A. S. Green, P. L. Lumsdaine, N. J. Ross, P. Selinger and B. Valiron, Quipper: A scal-
able quantum programming language, In Proceedings of the 34th ACM SIGPLAN Con-
ference on Programming Language Design and Implementation, PLDI '13, pp. 333–342.
Association for Computing Machinery, New York, NY, USA, ISBN 978-1-4503-2014-6,
doi:10.1145/2491956.2462177 (2013).

[67] A. JavadiAbhari, S. Patil, D. Kudrow, J. Heckey, A. Lvov, F. T. Chong and M. Martonosi,
ScaffCC: Scalable compilation and analysis of quantum programs, Parallel Computing 45,
2 (2015), doi:10.1016/j.parco.2014.12.001.
[68] N. Khammassi, I. Ashraf, X. Fu, C. Almudever and K. Bertels, QX: A high-performance quan-
tum computer simulation platform, In Design, Automation Test in Europe Conference Exhibition
(DATE), 2017, pp. 464–469. Lausanne, Switzerland, doi:10.23919/DATE.2017.7927034
(2017).
[69] J. R. Johansson, P. D. Nation and F. Nori, QuTiP: An open-source Python framework for the dy-
namics of open quantum systems, Computer Physics Communications 183(8), 1760 (2012),
doi:10.1016/j.cpc.2012.02.021.
23
https://arxiv.org/abs/1811.04968
https://doi.org/10.22331/q-2019-03-11-129
https://doi.org/10.5281/zenodo.2562111
https://doi.org/10.1145/3183895.3183901
https://arxiv.org/abs/1907.11217
https://arxiv.org/abs/1909.02559
https://arxiv.org/abs/1909.02559
https://doi.org/10.23919/DATE51398.2021.9474170
https://doi.org/10.1145/2491956.2462177
https://doi.org/10.1016/j.parco.2014.12.001
https://doi.org/10.23919/DATE.2017.7927034
https://doi"

And from paper Y:

"performance of SVM-QKE remains robust with additive noise in the kernel. In the following we prove noise robustness by introducing two additional results. First, we show that the dual SVM program (Eq. (5)) is robust, i.e., when the kernel used in (5) has a small additive perturbation, then the solution returned by the program also has a small perturbation. This follows from strong convexity of (5) and standard perturbation analysis of positive definite quadratic programs [46]. This result implies that the hyperplane w' obtained by the noisy kernel is close to the noiseless solution w with high probability. Second, we show that when w' is close to w, the linear classifier obtained by w' has high accuracy. This seemingly simple statement is not trivial, as the sign function is sensitive
to noise. That is, if ⟨φ(x), w⟩ is very close to 0, then a small perturbation in w could change its sign. We provide a solution to this problem by proving a stronger generalization bound. We show
7 that if a hyperplane w has a large margin on the training set, then not only does ⟨φ(x), w⟩ have the correct sign, it is also bounded away from 0 with high probability. Therefore, when the noisy solution w' is close to w, ⟨φ(x), w'⟩ also has the correct sign with high probability. Combining these two results with the proof sketch, we have the full proof of Theorem 2.
Conclusions and outlook We show that learning with quantum feature maps provides a way to harness the computational

power of quantum mechanics in machine learning problems. This idea leads to a simple quantum machine learning algorithm that makes no additional assumptions on data access and has rigorous and robust performance guarantees. While the learning problem we have presented here that demonstrates an exponential quantum speed-up is not practically motivated, our result sets a positive theoretical foundation for the search of practical quantum advantage in machine learning.

An important future direction is to construct quantum feature maps that can be applied to practical machine learning problems that are classically challenging. The results we have established here can be useful for the theoretical analysis of such proposals.

An important advantage of the SVM-QKE algorithm, which only uses quantum computers to estimate kernel entries, is that error-mitigation techniques can be applied [47–49] when the feature map circuit is sufficiently shallow. Our robustness analysis gives hope that an error-mitigated quantum feature map can still maintain its computational power. Finding quantum feature maps that are sufficiently powerful and shallow is therefore the stepping stone towards obtaining a quantum advantage in machine learning on near-term devices.

ACKNOWLEDGMENTS

We thank Sergey Bravyi and Robin Kothari for helpful comments and discussions. Y.L. was supported by Vannevar Bush faculty fellowship N00014-17-1-3025 and DOE QSA grant #FP00010905. Part of this work was done when Y.L. was a research intern at IBM. S.A. and K.T. acknowledge support from the MIT-IBM Watson AI Lab under the project Machine Learning in Hilbert Space, the IBM Research Frontiers Institute and the ARO Grant W911NF-20-1-0014.

[1] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Nature 549, 195 (2017).
[2] S. Arunachalam and R. de Wolf, SIGACT News 48, 41–67 (2017).
[3] V. Dunjko and H. J. Briegel, Reports on Progress in Physics 81, 074001 (2018).
[4] C. Ciliberto, M. Herbster, A. D. Ialongo, M. Pontil, A. Rocchetto, S. Severini, and L. Wossnig, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 474, 20170551 (2018).
[5] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, Rev. Mod. Phys. 91, 045002 (2019).
[6] A. W. Harrow, A. Hassidim, and S. Lloyd, Phys. Rev. Lett. 103, 150502 (2009).
[7] N. Wiebe, D. Braun, and S. Lloyd, Phys. Rev. Lett. 109, 050505 (2012).
[8] S. Lloyd, M. Mohseni, and P. Rebentrost, Quantum algorithms for supervised and unsupervised machine learning (2013), arXiv:1307.0411 [quant-ph].

[9] S. Lloyd, M. Mohseni, and P. Rebentrost, Nature Physics 10, 631 (2014).
[10] P. Rebentrost, M. Mohseni, and S. Lloyd, Phys. Rev. Lett. 113, 130503 (2014).
[11] S. Lloyd, S. Garnerone, and P. Zanardi, Quantum algorithms for topological and geometric analysis of big data (2014), arXiv:1408.3106 [quant-ph].
[12] I. Cong and L. Duan, New Journal of Physics 18, 073011 (2016).
[13] I. Kerenidis and A. Prakash, Quantum recommendation systems (2016), arXiv:1603.08675 [quant-ph].
https://doi.org/10.1038/nature23474
https://doi.org/10.1145/3106700.3106710
https://doi.org/10.1088/1361-6633/aab406
https://doi.org/10.1098/rspa.2017.0551
https://doi.org/10.1098/rspa.2017.0551
https://doi.org/10.1103/RevModPhys.91.045002
https://doi.org/10.1103/PhysRevLett.103.150502
https://doi.org/10.1103/PhysRevLett.109.050505
https://arxiv.org/abs/1307.0411
https://doi.org/10.1038/nphys3029
https://doi.org/10.1103/PhysRevLett.113.130503
https://arxiv.org/abs/1408.3106
https://doi.org/10.1088/1367-2630/18/7/073011
https://arxiv.org/abs/1603.08675
8
[14] F. G. S. L. Brandão, A. Kalev, T. Li, C. Y.-Y. Lin, K. M. Svore, and X. Wu, in 46th International Colloquium on Automata, Languages, and Programming (ICALP 2019), Leibniz International Proceedings in Informatics (LIPIcs), Vol. 132 (2019) pp. 27:1–27:14.
[15] P. Rebentrost, A. Steffens, I. Marvian, and S. Lloyd, Phys. Rev. A 97, 012327 (2018).
[16] Z. Zhao, J. K. Fitzsimons, and J. F. Fitzsimons, Phys. Rev. A 99, 052331 (2019).
[17] S. Aaronson, Nature Physics 11, 291 (2015).
[18] E. Tang, in Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing , STOC (2019) p. 217–228.
[19] E. Tang, Quantum-inspired classical algorithms for principal component analysis and supervised clustering (2018), arXiv:1811.00414 [cs.DS].
[20] A. Gilyén, S. Lloyd, and E. Tang, Quantum-inspired low-rank stochastic regression with logarithmic dependence on the dimension (2018), arXiv:1811.04909 [cs.DS].
[21] N.-H. Chia, H.-H. Lin, and C. Wang, Quantum-inspired sublinear classical algorithms for solving low-rank linear systems (2018), arXiv:1811.04852 [cs.DS].
[22] C. Ding, T.-Y. Bao, and H.-L. Huang, Quantum-inspired support vector machine (2019), arXiv:1906.08902 [cs.LG].
[23] N.-H. Chia, A. Gilyén, T. Li, H.-H. Lin, E. Tang, and C. Wang, in Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing , STOC (2020) p. 387–400.
[24] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Physical Review A 98, 032309 (2018).

[25] E. Farhi and H. Neven, arXiv preprint arXiv:1802.06002 (2018).
[26] E. Grant, M. Benedetti, S. Cao, A. Hallam, J. Lockhart, V. Stojevic, A. G. Green, and S. Severini, npj Quantum Information 4, 1 (2018).
[27] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, Physical Review A 101, 032308 (2020).
[28] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, Quantum Science and Technology 4, 043001 (2019).
[29] V. Havĺıček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, Nature 567, 209 (2019).
[30] M. Schuld and N. Killoran, Phys. Rev. Lett. 122, 040504 (2019).
[31] B. E. Boser, I. M. Guyon, and V. N. Vapnik, in Proceedings of the Fifth Annual Workshop on Computational Learning Theory , COLT (1992) p. 144–152.
[32] V. Vapnik, The nature of statistical learning theory (Springer science & business media, 2013).
[33] M. Anthony and P. L. Bartlett, Combinatorics, Probability and Computing 9, 213–225 (2000).
[34] J. Shawe-Taylor, P. L. Bartlett, R. C. Williamson, and M. Anthony, IEEE Transactions on Information Theory 44, 1926 (1998).
[35] P. Bartlett and J. Shawe-Taylor, Generalization performance of support vector machines and other pattern classifiers, in Advances in Kernel Methods: Support Vector Learning (MIT Press, Cambridge, MA, USA, 1999) p. 43–54.
[36] J. Shawe-Taylor and N. Cristianini, IEEE Transactions on Information Theory 48, 2721 (2002).
[37] M. J. Kearns, The computational complexity of machine learning (MIT press, 1990).
[38] R. A. Servedio and S. J. Gortler, SIAM J. Comput. 33, 1067–1092 (2004).
[39] R. Sweke, J.-P. Seifert, D. Hangleiter, and J. Eisert, On the quantum versus classical learnability of discrete distributions (2020), arXiv:2007.14451 [quant-ph].
[40] X. Gao, Z.-Y. Zhang, and L.-M. Duan, Science Advances 4, 10.1126/sciadv.aat9004 (2018).
[41] M. J. Kearns and U. V. Vazirani, An introduction to computational learning theory (MIT press, 1994).
[42] P. W. Shor, SIAM Journal on Computing 26, 1484 (1997).
[43] M. Blum and S. Micali, SIAM J. Comput. 13, 850–864 (1984).
[44] D. Aharonov and A. Ta-Shma, SIAM Journal on Computing 37, 47 (2007).
[45] P. L. Bartlett and P. M. Long, Journal of Computer and System Sciences 56, 174 (1998).
[46] J. W. Daniel, Mathematical Programming 5, 41 (1973).
[47] K. Temme, S. Bravyi, and J. M. Gambetta, Phys. Rev. Lett. 119, 180509 (2017).
[48] Y. Li and S. C. Benjamin, Phys. Rev. X 7, 021050 (2017).
[49] A. Kandala, K. Temme, A. D. Córcoles, A. Mezzacapo, J. M. Chow, and J. M. Gambetta, Nature 567, 491 (2019).
[50] M. Mosca and C. Zalka, International Journal of Quantum Information 02, 91 (2004).
[51] C. J. Burges, Data Mining and Knowledge Discovery 2, 121 (1998).
[52] A. J. Smola and B. Schölkopf, Statistics and Computing 14, 199 (2004).

```
https://doi.org/10.4230/LIPIcs.ICALP.2019.27
https://doi.org/10.4230/LIPIcs.ICALP.2019.27
https://doi.org/10.1103/PhysRevA.97.012327
https://doi.org/10.1103/PhysRevA.99.052331
https://doi.org/10.1038/nphys3272
https://doi.org/10.1145/3313276.3316310
https://arxiv.org/abs/1811.00414
https://arxiv.org/abs/1811.04909
https://arxiv.org/abs/1811.04852
https://arxiv.org/abs/1906.08902
https://doi.org/10.1145/3357713.3384314
https://doi.org/10.1145/3357713.3384314
https://doi.org/10.1088/2058-9565/ab4eb5
https://doi.org/10.1038/s41586-019-0980-2
https://doi.org/10.1103/PhysRevLett.122.040504
https://doi.org/10.1145/130385.130401
https://doi.org/10.1145/130385.130401
https://doi.org/10.1017/S0963548300004247
https://doi.org/10.1137/S0097539704412910
https://arxiv.org/abs/2007.14451
https://doi.org/10.1126/sciadv.aat9004
https://doi.org/10.1137/S0097539795293172
https://doi.org/10.1137/0213053
https://doi.org/10.1137/060648829
https://doi.or"
```

Based on the similarities of the documents above and their similar sections, we propose that the words "**http, https, arxiv, org, abs, and doi**" should be added to the stop words list in any future experiments. While these words are repeated across the corpus, they do not add any extra context to the papers and as such should be applied accordingly. Given the time constraint in completing these experiments, updating the stop words list and re-running the experiments would require extra weeks of experimentation. Below we take a deeper look at the performance of the different models when we compare the papers in subsections.

5.2.1 TF-IDF

Figure 5 shows the references in the paper "Uni-Perceiver-MoE: Learning Sparse Generalist Models with Conditional MoEs". After running the TF-IDF model, the top hundred (100) similar papers to the input paper are shown below:

```
[46]: models_performance("tfidf_2")

CALCULATING AI PAPERS
Paper: Language Models are Few-Shot Learners.txt 0.983631819237239
Paper: ST-MoE:_Designing_Stable_and_Transferable_Sparse_Expert_Models.txt 0.9816920453444842
Paper: XFBoost:_Improving_Text_Generation_with_Controllable_Decoders.txt 0.9702242941646023
Paper: Zero-Shot Text-to-Image Generation.txt 0.966795788979384
Paper: Improving_Search_through_A3C_Reinforcement_Learning_based_Conversational
__Agent.txt 0.9633402344989505
Paper: Deep_Extrapolation_for_Attribute-Enhanced_Generation.txt 0.9631680145510517
Paper: Bayesian_Neural_Networks_at_Scale:_A_Performance_Analysis_and_Pruning
__Study.txt 0.9599467004052123
Paper: Improving_Compositionality_of_Neural_Networks_by_Decoding
__Representations_to_Inputs.txt 0.9577902887738597
Paper: Robust_Generalization_of_Quadratic_Neural_Networks_via_Function
__Identification.txt 0.9575376582007576
Paper: Engineering_flexible_machine_learning_systems_by_traversing_functionally
__invariant_paths_in_weight_space.txt 0.9571988666012089
Paper: Decision Transformer- Reinforcement Learning via Sequence Modeling.txt 0.9563369972711565
Paper: Locally_Sparse_Neural_Networks_for_Tabular_Biomedical_Data.txt 0.9549987436978679
Paper: Variational_Autoencoder_with_Disentanglement_Priors_for_Low-Resource
__Task-Specific_Natural_Language_Generation.txt 0.9545170571547754
Paper: ANNA:_Enhanced_Language_Representation_for_Question_Answering.txt 0.9512862589073308
Paper: lamBERT:_Language_and_Action_Learning_Using_Multimodal_BERT.txt 0.9509133254836409
Paper: The_Pitfalls_of_Simplicity_Bias_in_Neural_Networks.txt 0.949951940358343
Paper: Implicit_Policy_for_Reinforcement_Learning.txt 0.9490786558253564
Paper: Embodied Multimodal Multitask Learning.txt 0.9482961775661005
Paper: Layer Normalization.txt 0.9480370848133277
Paper: Knowledge_Transfer_by_Discriminative_Pre-training_for_Academic
__Performance_Prediction.txt 0.9478920305955914
Paper: Microsoft COCO Captions- Data Collection and Evaluation Server.txt 0.9453937796600029
Paper: Dynamic_Inference_with_Neural_Interpreters.txt 0.9449572029629227
Paper: Comparing_Generative_Adversarial_Network_Techniques_for_Image_Creation
__and_Modification.txt 0.9437602990435665
Paper: Machine_Reading_Comprehension:_a_Literature_Review.txt 0.9432507988401356
Paper: i-Mix:_A_Domain-Agnostic_Strategy_for_Contrastive_Representation
__Learning.txt 0.9429921909900393
Paper: SGD_Learns_Over-parameterized_Networks_that_Provably_Generalize_on
__Linearly_Separable_Data.txt 0.9427296290909218
Paper: A_Survey_on_Contextual_Embeddings.txt 0.9426283738277964
Paper: Linking_Emergent_and_Natural_Languages_via_Corpus_Transfer.txt 0.9421899174612062
Paper: Object_Ordering_with_Bidirectional_Matchings_for_Visual_Reasoning.txt 0.9399416606825054
Paper: Step-unrolled_Denoising_Autoencoders_for_Text_Generation.txt 0.9398046907826184
Paper: Character-level_Convolutional_Network_for_Text_Classification_Applied_to
__Chinese_Corpus.txt 0.9398032787831421
Paper: Sparse_Meta_Networks_for_Sequential_Adaptation_and_its_Application_to
__Adaptive_Language_Modelling.txt 0.9385336742506258
Paper: Pairwise_Margin_Maximization_for_Deep_Neural_Networks.txt 0.9372608674774717
Paper: On_the_Adversarial_Robustness_of_Vision_Transformers.txt 0.936929503874554
Paper: PackIt:_A_Virtual_Environment_for_Geometric_Planning.txt 0.9363262944251848
Paper: Poison_Attacks_against_Text_Datasets_with_Conditional_Adversarially
__Regularized_Autoencoder.txt 0.9357124970654186
Paper: HealthPrompt:_A_Zero-shot_Learning_Paradigm_for_Clinical_Natural
__Language_Processing.txt 0.9347903283206462
Paper: WikiBERT_models:_deep_transfer_learning_for_many_languages.txt 0.9343228422646195
Paper: Memory_Augmented_Neural_Networks_with_Wormhole_Connections.txt 0.9333025107235916
```

```
Paper: Memory_Augmented_Neural_Networks_with_Wormhole_Connections.txt 0.9333025107235916
Paper: Temporal_Activity_Detection_in_Untrimmed_Videos_with_Recurrent_Neural
__Networks.txt 0.9321239186463562
Paper: CTAL:_Pre-training_Cross-modal_Transformer_for_Audio-and-Language
__Representations.txt 0.9319725233194085
Paper: Summarizing_a_virtual_robot's_past_actions_in_natural_language.txt 0.9319082455736886
Paper: Recovering_the_Lowest_Layer_of_Deep_Networks_with_High_Threshold
__Activations.txt 0.9315476742226093
Paper: Computational_principles_of_intelligence:_learning_and_reasoning_with
__neural_networks.txt 0.930347428068006
Paper: SimpleBooks:_Long-term_dependency_book_dataset_with_simplified_English
__vocabulary_for_word-level_language_modeling.txt 0.9303027591639041
Paper: Quantum_advantage_in_training_binary_neural_networks.txt 0.9296191910887425
Paper: ProphetNet-X:_Large-Scale_Pre-training_Models_for_English,_Chinese,
__Multi-lingual,_Dialog,_and_Code_Generation.txt 0.9289203474291318
Paper: RECKONition:_a_NLP-based_system_for_Industrial_Accidents_at_Work
__Prevention.txt 0.9288063003798664
Paper: Pointer_Value_Retrieval:_A_new_benchmark_for_understanding_the_limits_of
__neural_network_generalization.txt 0.9287041214871538
Paper: An_implementation_of_the_"Guess_who?"_game_using_CLIP.txt 0.9267713292710613
Paper: Local_SGD_Optimizes_Overparameterized_Neural_Networks_in_Polynomial_Time.txt 0.9266846899138402
Paper: Dynamic_Capacity_Networks.txt 0.9266513626310112
Paper: Conceptual 12M- Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts.txt 0.9251433034817508
Paper: Parametric_generation_of_conditional_geological_realizations_using
__generative_neural_networks.txt 0.9246209268029654
Paper: Challenges_in_Measuring_Bias_via_Open-Ended_Language_Generation.txt 0.9243098645688534
Paper: Generating_Natural_Language_Adversarial_Examples.txt 0.9240836637631649
Paper: Ouroboros:_On_Accelerating_Training_of_Transformer-Based_Language_Models.txt 0.9238003542827129
Paper: ProtoTransformer:_A_Meta-Learning_Approach_to_Providing_Student_Feedback.txt 0.9235546737886875
Paper: Livewired_Neural_Networks:_Making_Neurons_That_Fire_Together_Wire
__Together.txt 0.9229355409642017
Paper: Semantic-guided_Image_Virtual_Attribute_Learning_for_Noisy_Multi-label
__Chest_X-ray_Classification.txt 0.9207806540335184
Paper: On_the_link_between_conscious_function_and_general_intelligence_in
__humans_and_machines.txt 0.9205724201084028
Paper: N2N_Learning:_Network_to_Network_Compression_via_Policy_Gradient
__Reinforcement_Learning.txt 0.9194400517589486
Paper: Regularization_Effect_of_Fast_Gradient_Sign_Method_and_its
__Generalization.txt 0.9193108446272993
Paper: Merging_of_neural_networks.txt 0.9192315982046922
Paper: Comparing_Deep_Neural_Nets_with_UMAP_Tour.txt 0.9191089319302069
Paper: Towards_Robust_Neural_Networks_via_Close-loop_Control.txt 0.9186326925858973
Paper: Towards_Interpreting_Recurrent_Neural_Networks_through_Probabilistic
__Abstraction.txt 0.9161506057604383
Paper: Removing_Confounding_Factors_Associated_Weights_in_Deep_Neural_Networks
__Improves_the_Prediction_Accuracy_for_Healthcare_Applications.txt 0.9156250477407674
Paper: Training_robust_anomaly_detection_using_ML-Enhanced_simulations.txt 0.9153174071331522
Paper: Safe_Mutations_for_Deep_and_Recurrent_Neural_Networks_through_Output
__Gradients.txt 0.9153098623585542
Paper: The_Search_for_Sparse,_Robust_Neural_Networks.txt 0.9151474157259158
Paper: ViViT- A Video Vision Transformer.txt 0.9143485347802669
Paper: Generative_Adversarial_Residual_Pairwise_Networks_for_One_Shot_Learning.txt 0.9143483336232506
Paper: Modularized_Morphing_of_Neural_Networks.txt 0.9143056358062525
Paper: Why_Build_an_Assistant_in_Minecraft?.txt 0.9141952122156272
Paper: Entanglement_Entropy_of_Target_Functions_for_Image_Classification_and
__Convolutional_Neural_Network.txt 0.9140739451202173
Paper: Learning_epidemic_threshold_in_complex_networks_by_Convolutional_Neural
__Network.txt 0.9138199745524235
Paper: Can_Neural_Networks_Understand_Logical_Entailment?.txt 0.9135996256534148
Paper: Neural_Networks_in_Adversarial_Setting_and_Ill-Conditioned_Weight_Space.txt 0.9129780136795492
Paper: SparseDNN:_Fast_Sparse_Deep_Learning_Inference_on_CPUs.txt 0.9123885605754154
Paper: Stochastic_Neural_Networks_with_Infinite_Width_are_Deterministic.txt 0.911494847001695
Paper: Hardness_of_Learning_Neural_Networks_with_Natural_Weights.txt 0.9111182206852089
```

Figure 18. Top 100 Similar Documents for Uni-Perceiver-MoE: Learning Sparse Generalist Models with Conditional MoEs Using TF-IDF II

As with the previous sections, we show the references in our paper that are part of the top fifty (50), and hundred (100) similar documents as predicted by our TF-IDF model. In each figure, we show the reference and its similarity score to the input paper. Finally, we calculate the percentage of references accurately predicted.

```
[18]: models_performance("tfidf_2")

CALCULATING AI PAPERS
Paper: Embodied Multimodal Multitask Learning.txt 0.9482961775661005
Paper: Layer Normalization.txt 0.9480370848133277
Paper: Microsoft COCO Captions- Data Collection and Evaluation Server.txt 0.9453937796600029
Paper: Conceptual 12M- Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts.txt 0.9251433034817508
Paper: ViViT- A Video Vision Transformer.txt 0.9143485347802669
0.29411764705882354
```

Figure 19. TF-IDF II Performance on Sample Paper I

```
models_performance("tfidf_2")

CALCULATING AI PAPERS
Paper: Embodied Multimodal Multitask Learning.txt 0.9482961775661005
Paper: Layer Normalization.txt 0.9480370848133277
Paper: Microsoft COCO Captions- Data Collection and Evaluation Server.txt 0.9453937796600029
0.17647058823529413
```

Figure 20. TF-IDF II Performance on Sample Paper II

As shown above, our TF-IDF model predicted 17.65% of the references as part of the top fifty (50), and 29.41% when considering the top hundred(100) similar documents. Shown below is a table of all the documents and their related scores.

Table 15

TF-IDF II Model Performance on Corpus

Paper	*Category*	*Score (Top 50)*	*Score (Top 100)*
Functional Code Building Genetic Programming	AI	12.5%	12.5%
TwiBot-22: Towards Graph-Based Twitter Bot Detection	AI	0%	11.11%
Jewelry Shop Conversational Chatbot	AI	0%	0%
Uni-Perceiver-MoE: Learning Sparse Generalist Models with Conditional MoEs	AI	17.65%	29.41%
Visualization in virtual reality: a systematic review	VR	15%	45%
Joint Compute-Caching-Communication Control for Online Data-Intensive Service Delivery	VR	31.58%	47.37%
6G Survey on Challenges, Requirements, Applications, Key Enabling Technologies, Use Cases, AI integration issues and Security aspects	VR	6.67%	6.67%

Paper	*Category*	*Score (Top 50)*	*Score (Top 100)*
Quantifying the Effects of Working in VR for One Week	VR	22.22%	33.33%
Neo-GNNs: Neighborhood Overlap-aware Graph Neural Networks for Link Prediction	NN	10%	10%
Learning Vehicle Trajectory Uncertainty	NN	20%	20%
Early Transferability of Adversarial Examples in Deep Neural Networks	NN	29.41%	35.29%
Face-Dubbing++: Lip-Synchronous, Voice Preserving Translation of Videos	NN	27.78%	33.33%
NLU for Game-based Learning in Real: Initial Evaluations	NLP	6.25%	6.25%
Multi-Agent Reinforcement Learning is A Sequence Modeling Problem	NLP	35.29%	47.05%
Differentially Private Model Compression	NLP	0%	0%
Quantum Neural Network Classifiers: A Tutorial	NLP	5.56%	5.56%

Based on Table 15, the category with the highest average score is VR, with an average score of 18.89% for the top 50 papers and 33.10% for the top 100 papers. This is higher than the average scores for the other categories, which are NN (21.80% and 24.66% for the top 50 and top 100, respectively), NLP (11.78% and 14.71% for the top 50 and top 100, respectively), and AI (7.53% and 13.25% for the top 50 and top 100 papers, respectively).

5.2.2 BERT

Partial comparison using BERT was not possible due to resource constraints. The partial comparison model was only able to process the first five (5) documents against the corpus (9, 088 documents) in twenty-four (24) hours. While Google Colab was faster, the projected cost of running the model would've exceeded seven thousand US dollars ($7000).

5.2.3 Doc2Vec

Like the partial comparison for BERT, the doc2vec partial comparison model wasn't successful due to its processing speed. During experimentation, we were able to compare a hundred and fifteen (115) documents against the corpus in twenty-one days. Thus, we estimated that it would take approximately fifty (50) months to complete the experiment at its current pace. Like BERT, our Google Colab estimate for completing the comparison is approximately eight thousand US dollars ($8000).

5.2.4 Word2Vec

Shown below is a table of all test documents and their related scores when partially compared using Word2Vec.

Table 16

Word2Vec-II Model Performance on Corpus

Paper	*Category*	*Score (Top 50)*	*Score (Top 100)*
Functional Code Building Genetic Programming	AI	25%	25%
TwiBot-22: Towards Graph-Based Twitter Bot Detection	AI	0%	0%
Jewelry Shop Conversational Chatbot	AI	0%	0%
Uni-Perceiver-MoE: Learning Sparse Generalist Models with Conditional MoEs	AI	5.88%	23.5%
Visualization in virtual reality: a systematic review	VR	50%	75%
Joint Compute-Caching-Communication Control for Online Data-Intensive Service Delivery	VR	57.89%	63.16%

Paper	*Category*	*Score (Top 50)*	*Score (Top 100)*
6G Survey on Challenges, Requirements, Applications, Key Enabling Technologies, Use Cases, AI integration issues and Security aspects	VR	33.33%	40%
Quantifying the Effects of Working in VR for One Week	VR	27.78%	44.44%
Neo-GNNs: Neighborhood Overlap-aware Graph Neural Networks for Link Prediction	NN	10%	15%
Learning Vehicle Trajectory Uncertainty	NN	20%	20%
Early Transferability of Adversarial Examples in Deep Neural Networks	NN	41.17%	47.08%
Face-Dubbing++: Lip-Synchronous, Voice Preserving Translation of Videos	NN	33.33%	38.89%
NLU for Game-based Learning in Real: Initial Evaluations	NLP	6.25%	12.50%

Paper	*Category*	*Score (Top 50)*	*Score (Top 100)*
Multi-Agent Reinforcement Learning is A Sequence Modeling Problem	NLP	47.06%	52.94%
Differentially Private Model Compression	NLP	0%	0%
Quantum Neural Network Classifiers: A Tutorial	NLP	0%	0%

Based on Table 16, the category with the highest average score is VR, with an average score of 42.5% for the top 50 papers and 55.54% for the top 100 papers. This is higher than the average scores for the other categories, which are NN (26.125% and 30.24% for the top 50 and top 100, respectively), NLP (13.27% and 16.36% for the top 50 and top 100, respectively), and AI (7.72% and 12.125% for the top 50 and top 100 papers, respectively).

5.2.5 GloVe

Like Doc2Vec, and BERT, partial comparison for GloVe was also not successful due to limited resource and time constraints. Similarly, using Google Colab promised to complete the experiments, however. The cost of completing the experiment using Colab was estimated to be eight thousand US dollars ($8000).

Chapter 6

Analysis And Discussion

In this chapter, we will begin by analyzing the results of our experiment(s), and consequently our contribution to the current IR landscape.

6.1 Performance Analysis

Firstly, we will discuss the results of TF-IDF. Below is a table of the average performance of the two approaches on our test documents.

Table 17

TF-IDF and TF-IDF II Comparison

Category	*TF-IDF (50)*	*TF-IDF II (50)*	*TF-IDF (100)*	*TF-IDF II (100)*
VR	24.49%	18.89%	31.29%	33.10%
AI	15.17%	7.53%	20.99%	13.25%
NLP	9.01%	11.78%	15.07%	14.71%
NN	5.42%	21.80%	10.55%	24.66%

Although not significant, there is a slight increase in the average performance of TF-IDF II (when we compare documents in part). The most noticeable difference is in the performance of TF-IDF II on Neural Networks, a 300% increase for the top fifty (50) papers (from 5.42% to 21.80%) and over 130% performance improvement for the top hundred (100) papers (from 10.55% to 24.66%). Another point of interest is the negative performance of TF-IDF II on the Artificial Intelligence category. As shown by Table 10, there's a slight decrease in the number of references returned as part of similar documents across both experiment sizes. A 50.38% decrease when comparing the top fifty papers, and a 36.88% decrease when comparing the average of the top hundred papers. Based on their average performance, we can conclude that when comparing entire documents, TF-IDF performed better on the VR category. However, when comparing the documents in chunks, TF-IDF performed better on the NN category.

Next, we look at the performance of word2vec when the papers are compared in parts.

Table 18

Word2Vec and Word2Vec II Comparison

Category	*W2Vec (50)*	*W2Vec II (50)*	*W2Vec (100)*	*W2Vec II (100)*
AI	12.13%	7.72%	17.94%	12.125%
VR	20.32%	42.5%	24.69%	55.54%
NN	8.05%	26.125%	9.31%	30.24%
NLP	7.54%	13.27%	7.54%	16.36%

Based on Table 18, we can see that comparing the documents in chunks provides a higher performance by average across three (3) of the four (4) categories. The most visible difference is in VR and NN, with the performance improvement on NN going as high as 200% when considering the top hundred (100) documents. Like TF-IDF II, there is a slight decrease in the performance of Word2Vec II on Artificial Intelligence. We notice a 44.4% decrease in the top fifty (50) similar documents and a 32.39% decrease in the top one hundred (100) similar documents.

The consistency in negative performance of the models in Artificial Intelligence can be attributed to the overlapping content of the documents as most of the documents in the corpus are related to Artificial Intelligence. Due to the high accuracy in the relevant documents recommended by partial comparison, other documents in the corpus that are not selected as part of the references for the test documents are returned as the most similar documents.

6.2 Vector and Matrix Size Variations

The performance of each model varies depending on the vector size (max_features for TF-IDF) used to run the model. For models such as Doc2Vec and Word2Vec, the processing time varies directly proportional to the vector size i.e higher values of vector size would significantly increase the processing time of the models. In our initial experiment, TF-IDF was executed with a *max_features* of sixty-four (64). We further experimented with other values: one thousand (1,000), two thousand (2,000), five thousand (5000), ten thousand (10,000), and fifteen thousand (15,000). At 18,0000, the system goes out of memory due to the large matrix size. As shown in the attached charts, we can see that the performance grows as we go towards 10000 but flattens afterwards. At its peak, its performance is comparable to the recorded values for Doc2Vec (vector size 100). Below is a chart showing the performance of TF-IDF against their max_features on AI.

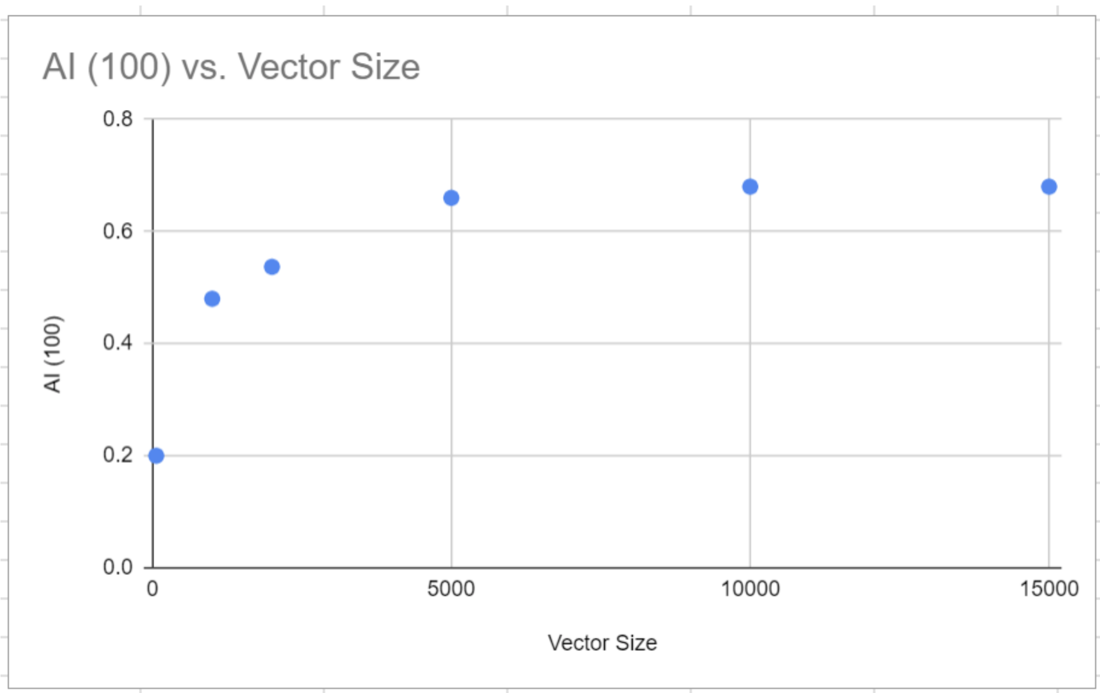

Figure 21. Performance of TF-IDF on AI Against Varying max_features I

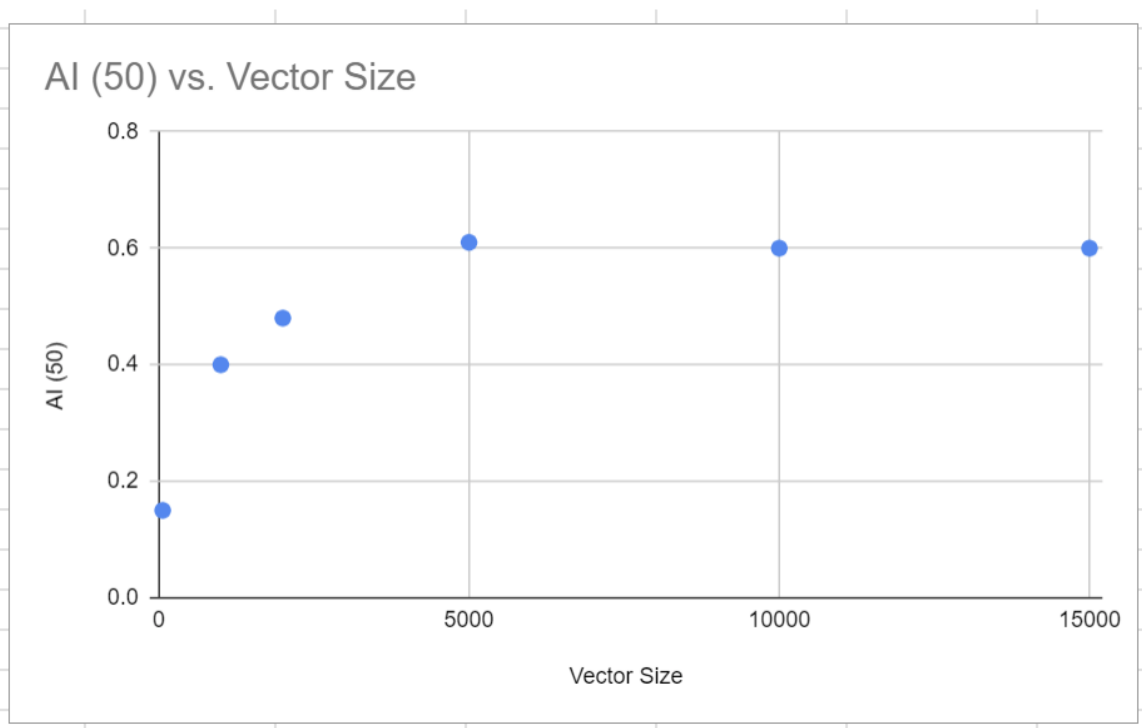

Figure 22. Performance of TF-IDF on AI Against Varying max_features II

Below are the vector sizes of the other models used in the experiment:

1. **Doc2Vec**: The vector_size for Doc2Vec determines the dimensionality of both word vectors(embeddings) and document vectors. For the initial experiment, we set this value to **100**. A higher value can increase the performance but also significantly increase the computational requirements.
2. **Word2Vec:** The vector_size for Word2Vec determines the dimensionality of each word vector. This value was 64 for the initial experiment.
3. **GloVe**: The vector size was 100.
4. **BERT:** The default size for BERT (128) was used. The transformer model used was **`sentence-transformers/all-mpnet-base-v2`**.

Based on the performance of TF-IDF with max_features >= 10,0000 being nearly as good as Doc2Vec (with 100 as the vector size) and Word2Vec (with 64 as vector size), we can imply that fine tuning Doc2Vec and Word2Vec with higher vector sizes will lead to an improved performance of those models, however, we lack the resources to do that currently. While we were able to run multiple experiments with TF-IDF using different values of max_features, we were unable to replicate the same feat with the other models due to their runtime. We intended to test several values, including thirty-two (32), sixty-four (64), one hundred and twenty-eight (128), and two hundred and fifty-six(256). However, due to the limitations of our current resources, conducting such experiments would exceed the time available for the current thesis.

6.3. The Contribution of This Work and How it Fits into The Current Information Retrieval Landscape

The Systematic Literature Review (SLR) (Feng et al. 2018), as discussed in Chapter 3, is a manual and labor-intensive process of compiling papers that are related to a specific topic. Our work improves this by attempting to automatically detect similar papers that should be referenced by an author when conducting literature reviews. Using the different machine learning models, we attempted to match a given paper with other similar papers in our corpus. By comparing the documents in whole, and in parts, we can conclude that comparing the documents in parts (subsections) more accurately identifies the similar (referenced) documents to a given paper. While Erekhinskaya et al. 2016 summarized the documents to automate literature reviews, our approach takes a step further by accurately selecting referenced papers in a medium sized corpus. We also show that the result of comparing two documents in whole and in parts can provide varying results in terms of similarity but more importantly, we enable researchers to search for similar documents without a structured Boolean query.

Chapter 7

Conclusion and Future Work

While comparing documents in parts proved to provide more accurate results, it is worthy to note that it is relatively slower than comparing the documents as single entities. Running on a 2.60Ghz CPU with four (4) cores, each of the original models generated a similarity matrix for the corpus within seventy-two (72) hours, asides from Doc2Vec which took another twenty-four (24) hours to complete. The processing time for comparing the documents in part was exponentially greater than comparing the documents as single entities. TF-IDF(II), when we compared the documents in part using TF-IDF, created the pairwise similarity matrix in approximately two weeks (2) on the same computer, while Word2Vec (II) did the same in three (3). Although these are long waiting times for the algorithms to execute, we show that the execution time can be reduced by using higher GPUs as provided by Google Colab.

Based on our experiments, Doc2Vec proved to be the most promising model for document similarity as it has the highest similarity score on the corpus. However, we also learned that it is the slowest model. Below are some of the other derived conclusions from the experiment:

- In a distinctive corpus, partial comparison is more accurate in selecting relevant similar documents.
- Partial comparison, while more accurate, is more intensive and requires higher processing power.
- Word2Vec and Doc2Vec are the most accurate models for recommending similar documents.

- TF-IDF is the least accurate when used in partial comparison but it is also the fastest.
- The most relevant part used by partial comparison to determine similar documents is their references.
- Similar documents can be suggested for an input document (text) without a query.
- Doc2Vec, GloVe, and BERT cannot be partially compared without significant computing resources beyond what's available in the current research environment.

For future work, the algorithm for comparing the documents in part can be improved to ensure that it is more efficient and performant on a larger corpus. The current implementation uses memoization and dynamic programming to avoid recalculating the similarity between two document pairs (i,j and j,i), however, we believe that the algorithm can be optimized for parallel execution or multi-threading. Another possible avenue for improvement is the application of large language models (LLM) to the tasks.

References

H. P. Luhn, "A Statistical Approach to Mechanized Encoding and Searching of Literary Information," in IBM Journal of Research and Development, vol. 1, no. 4, pp. 309-317, Oct. 1957, doi: 10.1147/rd.14.0309.

Vannevar Bush. As We May Think. Atlantic Monthly, 176:101–108, July 1945.

Christopher D. Manning, Schtze, H., P. R., (2009, April 7). *Introduction to information retrieval*.

Paul, C., Rettinger, A., Mogadala, A., Knoblock, C.A., Szekely, P. (2016). Efficient Graph-Based Document Similarity. In: Sack, H., Blomqvist, E., d'Aquin, M., Ghidini, C.,

Ponzetto, S., Lange, C. (eds) The Semantic Web. Latest Advances and New Domains. ESWC 2016. Lecture Notes in Computer Science(), vol 9678. Springer, Cham. https://doi.org/10.1007/978-3-319-34129-3_21

Elsayed, T., Lin, J., & Oard, D. W. (2008, June). Pairwise document similarity in large collections with mapreduce. In *Proceedings of ACL-08: HLT, short papers* (pp. 265-268).

Lee, M. D., Pincombe, B., & Welsh, M. (2005). An empirical evaluation of models of text document similarity. In *Proceedings of the annual meeting of the cognitive science society* (Vol. 27, No. 27).

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., and Harshman, R. A. (1990). Indexing by latent semantic analysis. Journal of the American Society of Information Science, 41(6):391–407.

Salton, G. (1989). Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Addison-Wesley, Boston, MA.

Damashek, M. (1995). Gauging similarity with n-grams: Language-independent categorization of text. Science, 267:843–848

Nelson Goodman. 1972. Seven strictures on similarity. Problems and Projects.

Daniel Bar, Torsten Zesch, and Iryna Gurevych. 2011. A Reflective View on Text Similarity. ¨ International Conference Recent Advances in Natural Language Processing (RANLP), pages 515–520..

Joeran Beel, Bela Gipp, Stefan Langer, and Corinna Breitinger. 2016. Research-paper recommender systems: a literature survey. International Journal on Digital Libraries, 17(4):305–338.

Ting-Hao ’Kenneth’ Huang, Chieh-Yang Huang, Chien-Kuang Cornelia Ding, Yen-Chia Hsu, and C. Lee Giles. 2020. CODA-19: Reliably Annotating Research Aspects on 10,000+ CORD-19 Abstracts Using a Non-Expert Crowd. arXiv:2005.02367

Yigal Attali and Jill Burstein. 2006. Automated essay scoring with e-rater v.2.0. Journal of Technology, Learning, and Assessment, 4(3).

G. Salton, A. Wong, and C.S. Yang, “A Vector Space Model for Automatic Indexing,” Comm. ACM, vol. 18, no. 11, pp. 613-620, 1975.

Li, B., & Han, L. (2013, October). Distance weighted cosine similarity measure for text classification. In *International conference on intelligent data engineering and automated learning* (pp. 611-618). Springer, Berlin, Heidelberg.

Qaiser, S., & Ali, R. (2018). Text mining: Use of TF-IDF to examine the relevance of words to documents. *International Journal of Computer Applications, 181*(1), 25–29. https://doi.org/10.5120/ijca2018917395

Hakim, A. A., Erwin, A., Eng, K. I., Galinium, M., & Muliady, W. (2015). “Automated document classification for news article in Bahasa Indonesia based on term frequency inverse document frequency (TF-IDF) approach,” 6th International Conference on Information Technology and Electrical Engineering: Leveraging Research and Technology, (ICITEE), 2014

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019, May 24). *Bert: Pre-training of deep bidirectional Transformers for language understanding*. arXiv.org. Retrieved November 16, 2022, from https://arxiv.org/abs/1810.04805

Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543).

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In Proceedings of the 31st International Conference on Machine Learning (ICML 2014), pages 1188–1196, Beijing, China.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In Proceedings of Workshop at the International Conference on Learning Representations, 2013, Scottsdale, USA.

Rumelhart, David E, Hinton, Geoffrey E, and Williams, Ronald J. Learning representations by back-propagating errors. Nature, 323(6088):533–536, 1986.

T. Mikolov, W.T. Yih, G. Zweig. Linguistic Regularities in Continuous Space Word Representations. NAACL HLT 2013.

Harris, Zellig. Distributional structure. Word, 1954.

Raulji, J. K., & Saini, J. R. (2016). Stop-word removal algorithm and its implementation for Sanskrit language. *International Journal of Computer Applications*, *150*(2), 15-17.

Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using Siamese Bert-Networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. https://doi.org/10.18653/v1/d19-1410

A. M. P. Braşoveanu and R. Andonie, "Visualizing Transformers for NLP: A Brief Survey," 2020 24th International Conference Information Visualisation (IV), 2020, pp. 270-279, doi: 10.1109/IV51561.2020.00051.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Carpineto, C., & Romano, G. (2012). A survey of automatic query expansion in information retrieval. ACM Computing Surveys. https://doi.org/10.1145/2071389.2071390

Crouch, C., McGill, M., Lesk, M., Jones, K. S., Fox, E. A., Harman, D., & Kraft, D. H. (1996). In Memorium: Gerald Salton, March 8, 1927-August 28, 1995. Journal of the American Society for Information Science, 47(2), 108–115.

https://doi.org/10.1002/(sici)1097-4571(199602)47:2<108::aid-asi2>3.0.co;2-2
Dubin, D. (2004). The most influential paper gerard salton never wrote. Library Trends.
Erekhinskaya, T., Balakrishna, M., Tatu, M., Werner, S., & Moldovan, D. (2016).

Knowledge extraction for literature review. In Proceedings of the ACM/IEEE Joint Conference on Digital Libraries. https://doi.org/10.1145/2910896.2925441

Feng, L., Chiam, Y. K., & Lo, S. K. (2018). Text-Mining Techniques and Tools for Systematic Literature Reviews: A Systematic Literature Review. In Proceedings - Asia-Pacific Software Engineering Conference, APSEC. https://doi.org/10.1109/APSEC.2017.10

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation. https://doi.org/10.1108/eb026526

Kelly, D., & Sugimoto, C. R. (2013). A systematic review of interactive information retrieval evaluation studies, 1967-2006. Journal of the American Society for Information Science and Technology. https://doi.org/10.1002/asi.22799

Luhn, H. P. (1958). The Automatic Creation of Literature Abstracts. IBM Journal of Research and Development, 159–165. https://doi.org/10.1147/rd.22.0159

Rocchio., J. J. (1965). Relevance Feedback in Information Retrieval, Report No. ISR-9 to the National Science Foundation.

Salton, G. (1971). The SMART Retrieval System Experiments in Automatic Text Processing. Prenticetice-Hall.

Sanderson, M., & Croft, W. B. (2012). The history of information retrieval research. In Proceedings of the IEEE. https://doi.org/10.1109/JPROC.2012.2189916

Turtle, H. (1994). Natural language vs. Boolean query evaluation: A comparison of retrieval performance. In Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 212–220).

Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, et al., "Roberta: A robustly optimized BERT pretraining approach", *CoRR*, vol. abs/1907.11692, 2019, [online] Available: http://arxiv.org/abs/1907.11692.

Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations", *8th International Conference on Learning Representations ICLR 2020*, April 26-30, 2020, [online] Available: https://openreview.net/forum?id=H1eA7AEtvS.

Appendix A

Arxiv File Download Code

Submitted with this thesis is a copy of the python used in downloading the documents from arxiv and converting the documents to PDF.

Appendix B

Document Similarity Code

Submitted with this thesis is a copy of the python used in the model experiments.

Appendix C

Model Reports Code

Submitted with this thesis is a copy of the python code used in calculating the similarity of documents using the data/result from Appendix B.