SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY - INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics: Data Engineering and Analytics

Scientific Document Representation Learning

Divya Bansal

Acknowledgment

I would like to express my deep gratitude towards Professor Dr. Matthias Grabmair and the TUM Legal Tech Group for giving me the opportunity and the resources to pursue this thesis. I am profoundly thankful to my dedicated advisor Mr. Santosh Tokala, for his unwavering support and guidance throughout the entire process of completing my thesis. His expertise, patience, and encouragement have been crucial in helping me navigate the challenges and complexities of my research. Lastly, I would like to extend my appreciation to my family and friends for their constant motivation and support.

Abstract

Representation learning has evolved as a primary component for tackling various NLP challenges. In this work, we deal with representation learning for scientific documents which is imperative for applications such as document classification and recommendation. Today, transformer-based contextual models such as BERT have become ubiquitous for representing sentences and short documents. However, recently, substantial work has been done to enhance these representations specifically for scientific documents, which naturally come with an additional inter-document signal in the form of citations. Prior works such as SPECTER utilize this signal to learn scientific document representations with the help of contrastive learning by minimizing the distance between the contextual representations of the documents connected through citations (positives) and maximizing the distance between representations of disconnected documents (negatives). However, in this work, we posit that two documents which are similar in certain citation aspects can be completely dissimilar in other aspects and thus we can leverage this distinction for creating more granular, aspect-aware representations. We propose SciAspect, a model similar to SPECTER, but for choosing the positives and negatives we consider whether two documents are connected with the same citation intent or not. We also experimented with a hybrid model that tries to strike a balance between the representation capabilities of SPECTER and SciAspect. Furthermore, instead of a simple variant of contrastive loss i.e., triplet loss used in prior works, we employ a general contrastive loss that can learn from multiple positive and negative samples at once. The models are evaluated on the SciDocs, MDCR and SciRepEval evaluation benchmarks. We found that in some cases, our models perform better than or are comparable to the previous best models, suggesting that incorporating aspect information could improve representations.

Contents

1. Introduction

1.1. Motivation

Scientific articles store vast amounts of knowledge amassed through many decades of research. They serve as the primary means to disseminate new information, ideas, and research findings, thus playing a crucial role in the advancement of science and understanding in various fields of study. However, the sheer volume and velocity of research publications that is rapidly growing every year, makes it difficult for scientists to keep track of recent research and to navigate the research landscape for finding relevant information. To cope with this problem of information overload, many NLP methods are being employed that help automatically analyze and explore scientific literature. These include the classification of articles into different categories, extraction of key phrases or key aspects from articles, argumentation mining, article summarization, and citation recommendation.

Building an effective NLP system from scientific text requires a good representation of the text in the form of concise, machine-understandable numerical vectors (also known as embeddings). However, constructing such representations that can encode all the essential features and the underlying meaning of the text is challenging. Scientific documents are generally long, diverse, and contain complex domain-specific language, making the task non-trivial. A common approach to solve this is to learn representations trained on textual and metadata features of the document with an objective function depending on the task we are trying to solve. For instance, for learning embeddings from the semantic features of a document for the task of document classification, a set of labeled examples is used with a cross-entropy loss objective to train a model to learn from the examples by minimizing the objective function and predicting the class of unlabeled documents. For accomplishing this, high-quality manually annotated data is frequently required to provide examples of how the data should be classified. Obtaining such data, however, is expensive, time-consuming, and needs domain knowledge.

In recent years many techniques have been proposed to learn generalized, task-agnostic, universal representations for documents, in a self-supervised or semi-supervised fashion (i.e., without the need for labeled data), which can perform well on multiple tasks. These generalized embeddings can be easily indexed and later efficiently retrieved for different applications. The transformer architecture-based [1] large pretrained language models (PLMs) like BERT [2] and its variants [3–5] have shown state-of-the-art performance across many NLP tasks and have become a de facto method to encode textual data. PLMs learn a robust encoder on a large unlabeled corpus (such as the whole of Wikipedia), through carefully designed self-supervised pretraining tasks. With this, they are able to learn contextual representations of text. [6] proposed BERT models pretrained specifically on scientific data and showed improvement in the performance of the resulting embeddings as compared to BERT embeddings trained on general natural language.

However, these embeddings still suffer from information loss. Due to the quadratic complexity of self-attention when learning as explained in chapter 2, most of the PLMs can only handle short text (e.g., 512 tokens for BERT) which is not sufficient for handling generally long and complex scientific documents. So generally, only the title and abstract of a document, are used to encode documents which may not be very informative to distinguish between the documents. Even the models like Longformer [7] and BigBird [8], which have been proposed to efficiently handle long text sequences using sparse attention, are still not able to capture the nuances and high-level semantic structure of scientific documents well. Moreover, BERT-based embeddings have been shown to generate similar embeddings for different documents if they have multiple common words even when they have completely different semantic meanings [9].

To address this problem, recent scientific knowledge representation models [10–13] started including additional inter and intra-document relatedness signals, during training, to enhance the quality of the representations. For instance, the current state-of-the-art models SciNCL [10] and ASPIRE [12] which were preceded by SPECTER [11], all leverage citations and co-citations as an indicator for similarity. With the help of contrastive learning, they try to improve the semantic representations by pulling representations of similar data points (positives) closer together, while pushing representations of dissimilar documents (negatives) apart.

Although citation links are a good signal to integrate information from other documents and complement the content-based embeddings, in this work, we posit that citations have different motivations and should not be treated equally. Scientific documents often describe multi-faceted arguments and ideas and thus can be similar or dissimilar in many different aspects. So the axes to measure their similarity should be different. This reasoning follows from previous works such as [14, 15] which argue that traditional document similarity measures do not consider the aspect-level similarity of documents, thus, integrating potential risks that arise from implicit biases assumed by the models that rely on a single notion of similarity.

Hence, we hypothesize that capturing aspect information, while learning the representations themselves, will make them capable of matching specific aspects and they can better capture overall document relatedness. This, in turn, can improve performance on downstream tasks like citation recommendation. In this work, we consider the citation intents between two documents as a signal for capturing aspect-specific relatedness.

A scientific document needs to cite relevant and important previous work to help readers understand its background, context, and innovation. It may cite other documents in different aspects, for example, to highlight the importance of a problem or to compare against results provided by another method. We speculate that it would be better to learn a representation that is aware of these nuances and is able to distinguish between different aspects in the latent space.

Moreover, as discussed, previous approaches (SPECTER, SciNCL) treat all citations equally. However, a large proportion of citations have a relatively low impact on a citing paper [16]. Consequently, these low-impact citations when weighted equally, may add significant noise and reduce the local semantic discriminative potential of the derived document embeddings [17]. For instance, the background citations may be semantically very different from the citations that are cited for result comparison. Accounting for them multiple times in the training data may neglect signals that can be learned from other citation aspects. So, including positive and negative samples for a query from all citation intents when learning the representations may also improve performance.

To investigate this, we propose a model called **SciAspect** that learns more granular, aspect-aware representations of scientific documents. In line with previous work, we first extract citations and references of the same anchor papers used in SPECTER and SciNCL along with their citation intent information from the Semantic Scholar API[1]. Here, each citation or reference link is classified as **methodology, background, result, or unknown** depending on the citation context. A link can belong to multiple classes at once as there can be multiple reasons for citation. Then the positive and negative pairs are curated depending on whether the papers are connected to the query papers with the same citation intent or not. The representations are then learned in different representation spaces, by first projecting contextual SciBERT embeddings of the anchor, positive and negative papers into separate aspect spaces with the help of a feed forward network, and then optimizing these spaces using a contrastive loss function.

Furthermore, in contrast to all previous approaches for learning scientific document representations that optimize on a simpler variant of contrastive loss called triplet loss, which uses only one positive and one negative for a query document at a time, we experiment with a general contrastive loss that can use multiple positives and negatives for a query document at a time. As demonstrated by [18], this loss subsumes triplet loss and should perform at least as well as the triplet loss.

SciAspect learns finer-grained aspect-informed representations of scientific documents in three different aspect spaces. In addition to this, we also experiment with a hybrid model by optimizing for a weighted loss between a global contrastive loss, learned by modeling positives and negatives without aspect information (as in SPECTER), and a local contrastive loss learned by modeling positives and negatives with the aspect information (as in SciAspect). We call this model **SciAspectHybrid**. With this, we expect to combine the representation power of both the coarse, aspect-less embeddings and fine, aspect-aware embeddings. In congruence with previous research, we empirically evaluate our final representations on three standard benchmarks namely SciDocs[11], MDCR[19] and SciRepEval[20] that evaluate the classification, regression and recommendation powers of the learned embeddings.

[1] https://www.semanticscholar.org/product/api

1.2. Contribution

The main contributions of this work can be summarized as follows:

1. We re-train SPECTER but with considering in-citations as positives. We also train this model ensuring there is no leakage between the train and test data.
2. We propose the use of a general contrastive loss as the objective for learning scientific document representations (SDRs) as opposed to triplet loss used in previous works. To the best of our knowledge, this is the first work that uses such a loss in this domain.
3. We present a novel approach for sampling positives and negatives for learning SDRs via contrastive learning. This approach uses citation intents as the basis for discriminating representations. Our models are able to outperform previously proposed models in some cases, hence proving the validity of our approach.

1.3. Structural Outline

The rest of the thesis is divided into 7 chapters. Chapter 2 presents the theoretical background and clarification of terminology that is required for understanding the given work. Chapter 3 traces the evolution of the various approaches proposed in the past for representing scientific documents. In Chapter 4, we introduce the reader to our proposed models and implementation details. Following that, in Chapter 5 we explain our experiments. In chapters 6 and 7, the results and their detailed analysis is provided. Finally, Chapter 8 contains the concluding discussion and future directions of this research.

2. Theory

In this chapter, we explain the background knowledge and relevant terminology required to understand the thesis.

2.1. Representation Learning

One of the most important design choices for solving any task in machine learning or in computer science in general, depends upon how the information is represented. In computer science, the choice of data structure, the database management system or the database schema influences performance. Similarly, in the context of machine learning, the representation of the input data in a way that a machine can understand and learn insights from, ultimately influences the final performance. Ideally, a good representation would be one that can encode as much information about the input data as possible such that it performs well on the learning objective. However, most representation learning problems face a trade-off between preserving as much information about the input data as possible while tending to other constraints such as cost, size, generalizability etc.

Representation learning of documents is particularly interesting because it is concerned with meaningfully representing the contents of a document in the form of numeric low-dimensional vectors or embeddings. Ideally, a document representation learning model should learn embeddings such that related documents are represented closer together in the vector space and unrelated documents are placed far apart. The matter of relatedness is subjective and depends on the use case of the representations. Documents could be related if they are semantically similar or if they were written by the same author or in the same country and numerous other combinations. This closeness in relation is often indicated by the Euclidean distance between the vectors or by cosine similarity between the vectors. The learned embeddings are evaluated by their performance in the task they are designed to solve on certain metrics.

A traditional way for representing a document based on its textual contents is using its lexical information. Such methods encode statistical information pertaining to the words/phrases used, their frequencies in the document, their intra-document frequencies and document lengths. Some of the standard models are bag-of-words (BOW)[21] and term frequency-inverse document frequency (TF-IDF) [22]. However, embeddings from such models cannot correctly compare the documents that have different lengths, use different words to express the same meaning, or use similar words to express different meanings as they compare only on the exact vocabulary used.

To solve the above problem semantic models were introduced. In this these, vector representations are learned by training deep neural networks on large natural language datasets (like Wikipedia) such that they retain the underlying linguistic relationship between the words and are able to compare texts regardless of whether they have common words or not.

Semantic models were first introduced by word2vec [23] to learn word embeddings that capture the underlying relationship between words such as co-occurrences and contextual relationships. This was followed by other word embedding models such as GloVe [24], FastText [25]. These were then extended to learn sentence-level and document-level representations (E.g.: Doc2Vec, [26]). More recently, pre-trained language models (PLMs), based on transformers [1] e.g., OpenAI GPT [27] and BERT [2], which encode additional contextual information have helped significantly advance the state-of-the art for several major NLP benchmarks. These models develop a statistical understanding of the language that they have been trained in. Embeddings learned from these models can be directly consumed or the models can be fine-tuned for a task with smaller datasets and resources. This work uses these models and language modeling using these is explained in detail in the next section.

In the case of scientific documents, besides textual data, their citation networks can also be leveraged to learn embeddings. These embeddings are designed to capture the structural relationships between the papers in the citation graph, such as proximity, similarity, or connectivity. Some popular techniques to learn citation embeddings are:

- DeepWalk [28]: It leverages the Skip-gram model from Word2Vec to learn embeddings on a citation graph. It also utilizes random walks to explore the graph, treating them as sentences. The Skip-gram model is then applied to predict the context (neighboring nodes) given a target node. By optimizing this prediction task, DeepWalk learns embeddings that capture the semantic and structural information of the graph.

- LINE [29]: It learns network-only graph representations in two phases - first-order proximity and second-order proximity to preserve the local and global network structures.

- Node2Vec [30]: It learns embeddings by capturing both local and global structural information of the citation graph. It generates random walks on the graph and uses a combination of breadth-first and depth-first search strategies to explore the neighborhood of nodes. By optimizing the objective function to predict the likelihood of neighboring nodes, Node2Vec produces embeddings that preserve the structural properties of the graph.

- Graph Neural Networks: These are a class of neural network architectures like Graph Convolution Networks(GCN) [31] or Graph Attention Networks(GAT) [32] that can be used to learn embeddings on citation graphs. They operate on the graph structure and utilize message passing between nodes to update the representations of each node based on its neighbors. By aggregating information from neighboring nodes iteratively, they produce embeddings that capture both local and global information of the citation graph.

2.2. Language Modelling

2.2.1. Pretraining

In NLP, pretraining refers to the process of learning some sort of representation of meaning for words or sentences by processing very large amounts of text [33]. The learned model is called a pretrained language model (PLM). Generally, a language model is trained by predicting the next word or phrase in a sentence based on the context it has seen so far. By doing this repeatedly on a massive scale, the model is trained to learn the statistical patterns and relationships within the language.

2.2.2. Finetuning

After pretraing, the knowledge captured in the pretrained language models can be used as it is or the parameters of the pretrained model can be adjusted further based on a specific task, domain or dataset. This process is called fine-tuning and by doing this the model can specialize, adapt and perform better on that particular task, domain or dataset. Fine-tuning is beneficial because it reduces the need for training large models from scratch. The pretrained model acts as a starting point, providing a solid foundation of language understanding, and the fine-tuning process tailors it to the specific task or domain at hand.

2.2.3. Transformers

Today, the most common architectural mechanism for language modeling is using transformers [1]. Transformers are able to effectively capture the context and relationships between words in a sentence. They are able to map sequences of input embeddings $(x_1, ..., x_n)$ to sequences of more contextually aware output embeddings $(y_1, ..., y_n)$ of the same length.

The core idea behind transformers is the notion of attention which is inspired by the way humans selectively focus on salient information. It allows a model to selectively attend to different parts of the input sequence when processing each element. Attention mechanisms have proven to be highly effective in capturing the dependencies and relationships within a sentence, making transformers superior in modeling long-range dependencies compared to their predecessors. Transformers are made up of stacks of transformer blocks, each of which is a multi-layer neural network made by combining self-attention linear layers, feed-forward networks, and self-attention layers.

Self-attention refers to the attention mechanism applied to the input sequence itself. Using it the model learns to weigh the relationship between each word in an input sequence to other words in the input sequence. To compute self-attention, each word in the sequence generates its query(q), key(k), and value(v) vectors by linearly projecting the original word embedding. This way the input embedding plays three different roles during the course of the attention process:

1. The **query** vector acts as the current focus of attention when being compared to all of the other preceding inputs.
2. The **key** vector acts as a preceding input being compared to the current focus of attention.
3. The **value** vector is used to compute the output for the current focus of attention.

These vectors are then used to calculate a similarity score between each word and every other word in the sequence and transform the embeddings according to the scores.

First, the $\boldsymbol{q}$, $\boldsymbol{k}$, and $\boldsymbol{v}$ vectors of all words in the sequence are put together in the matrices Q, K, and V respectively. These matrices are created by projecting the embedding of the input sequence $(\boldsymbol{x_1, ..., x_n})$ by multiplying with three weight matrices $\boldsymbol{W^Q}$, $\boldsymbol{W^K}$, $\boldsymbol{W^V}$ which are initialized and learned during the training process.

$$\boldsymbol{q}_i = \boldsymbol{W^Q x}_i; \ \boldsymbol{k}_i = \boldsymbol{W^K x}_i; \ \boldsymbol{v}_i = \boldsymbol{W^V x}_i \tag{2.1}$$

Then a self-attention score (Equation 2.2) is calculated by performing a dot product between the query and key matrices (divided by square root of the dimension of the k vector to get more stable gradients). The resulting scores are then passed through a SoftMax function to obtain attention weights. This ensures that the scores are normalized so that they are all positive and add up to 1. The weights indicate the importance of each key vector to the query vector and are used to weigh the corresponding value vectors. The weighted value vectors are summed up to produce the final attention output.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2.2}$$

The self-attention mechanism is applied multiple times in parallel, known as attention heads, to perform form MultiHead attention that captures diverse contextual information at different levels of granularity.

$$\begin{aligned} MultiHead(Q, K, V) &= Concat(head_1, ..., head_h)W^O \\ head_i &= Attention(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \tag{2.3}$$

Transformer-based models can process text in parallel which means that at any time the model is able to look at any word in the sentence it is processing. To be able to process long text sequences of different lengths in parallel while keeping track of the exact word order in the sequence, transformers encode the position of the word directly into the embedding. This is a "marker" that lets attention layers in the model identify where the word or text sequence they're looking at was located.

Figure 2.1 shows the full transformer architecture along with its components. The left block of the architecture is called an encoder and the right part is a decoder. The encoder takes the input sequence and creates its contextual representation (which is also called context) and the decoder takes this contextual representation as input and generates an output sequence. In recent years, various encoder-only or decoder-only architectures have been developed for creating large language models. Encoder-only architectures are more suitable for tasks that require contextual understanding, fine-grained semantic analysis, and adapting to specific downstream tasks through fine-tuning. Decoder-only models, on the other hand, are well-suited for generative tasks that involve generating text and maintaining coherence in a given context. In this thesis, we want to only encode scientific documents to generate SDRs, so we only include encoders-only models and next we briefly describe the most prominent encoder BERT.

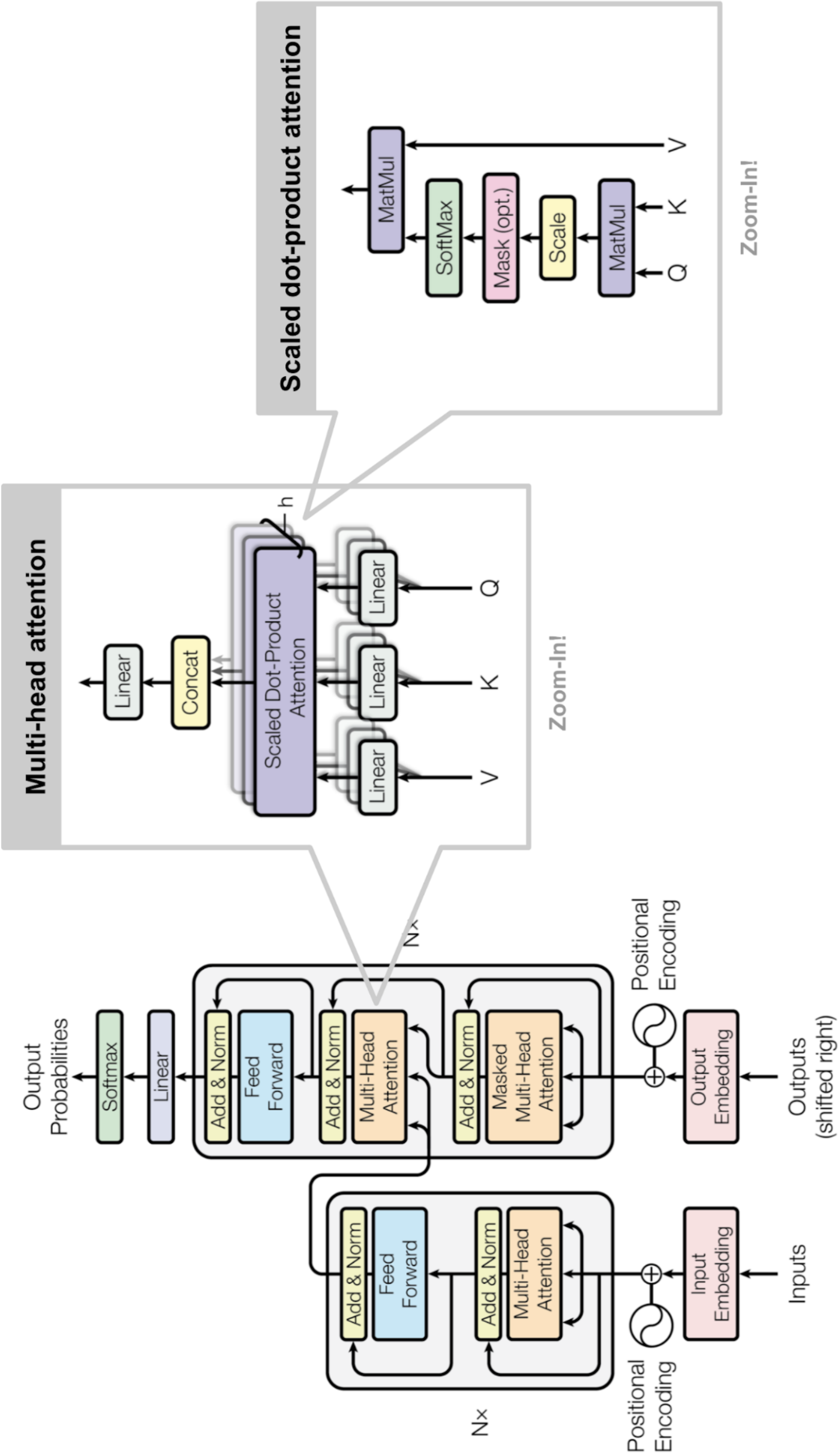


Figure 2.1.: The Transformer Architecture. (Source:https://neptune.ai/blog/bert-and-the-transformer-architecture)

2.2.4. BERT

Overview

BERT (Bidirectional Encoder Representations from Transformers) [2], a renowned transformer-based encoder model, extends the concept of self-attention to create a powerful pretrained language representation. BERT employs a bidirectional approach by utilizing both left-to-right and right-to-left contexts during training, enabling it to capture a comprehensive understanding of a word's context. It is pretrained on a large corpus of unlabelled text data, allowing it to learn general language patterns and semantics.

During pretraining, BERT masks certain words in the input sequence and trains the model to predict those masked words based on the surrounding context. This process instills BERT with the ability to contextualize words effectively. Additionally, BERT utilizes another pretraining objective called next sentence prediction, where it learns to predict whether two sentences in a pair are consecutive or not, further enhancing its understanding of sentence-level relationships.

After pretraining, BERT can be fine-tuned on specific downstream tasks by adding task-specific layers and training the model on labeled data. This fine-tuning process leverages the pretrained language representations of BERT to achieve state-of-the-art performance on a wide range of NLP tasks.

Architecture

BERT was released as two variants, $BERT_{BASE}$ and $BERT_{LARGE}$. $BERT_{BASE}$ has 12 layers in the Encoder stack while $BERT_{LARGE}$ has 24 layers in the Encoder stack. These are more than the Transformer architecture described in the original paper (6 encoder layers). BERT architectures (BASE and LARGE) also have larger feed-forward networks (768 and 1024 hidden units respectively), and more attention heads (12 and 16 respectively) than the Transformer architecture suggested in the original paper. It contains 512 hidden units and 8 attention heads. $BERT_{BASE}$ contains 110M parameters while $BERT_{LARGE}$ has 340M parameters. In this work we use the $BERT_{BASE}$ variant for all tasks.

Training BERT

BERT takes the [CLS] token as the first input token followed by a sequence of words as input. If a sequence contains two distinct sentences/entities, they are separated by the [SEP] token. A [SEP] token is also added at the end of the input sequence to indicate the end. The tokens are then converted to embeddings. The input embeddings are a combination of 3 types of embeddings:

1. **Position Embeddings:** BERT learns and uses positional embeddings to express the position of words in a sequence.

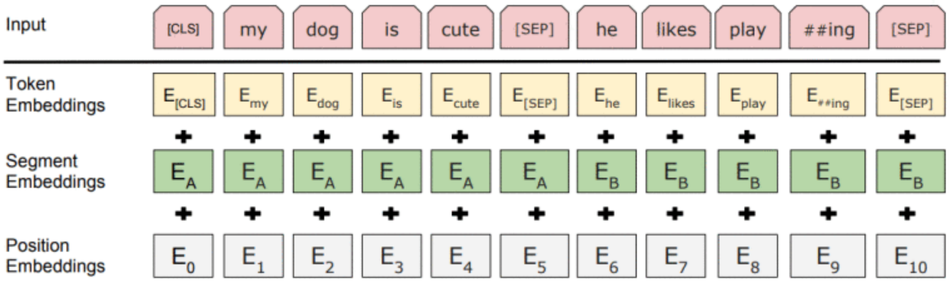


Figure 2.2.: BERT input layer

2. **Segment Embeddings:** BERT can also take sentence pairs as inputs for tasks like next sentence prediction and question-answering. That is why it learns a unique embedding for the first and the second sentences to help the model distinguish between them. In Figure 2.2, all the tokens marked as E_A belong to sentence A and the ones marked as E_B belong to sentence B.

3. **Token Embeddings:** These are the embeddings learned for the input token according to the used vocabulary.

It then passes the input to the above layers. Each layer applies self-attention and passes the result through a feed-forward network after then it hands off to the next encoder. The model outputs a vector of hidden size (768 for $BERT_{BASE}$). If we want to output a classifier from this model we can take the output corresponding to the [CLS] token.

Even though BERT has surpassed many NLP benchmarks, it has certain limitations pertaining to encoding long documents such as scientific documents. It tokenizes input texts into fixed-length sequences. The maximum token limit (e.g., 512 tokens for BERT-base) restricts the model's ability to handle longer texts. When a text exceeds this limit, it needs to be divided into smaller segments, leading to potential information loss and disconnections between segments. Moreover, BERT's memory or compute requirements increase quadratically with the input length(due to quadratic self-attention complexity), making it infeasible to process extremely long texts. Moreover, for very long texts, the context window for the attention process may not cover the entire document, resulting in incomplete understanding. As a result, BERT may struggle to establish coherent relationships between different sections or paragraphs, leading to sub-optimal performance on tasks that require a deep understanding of the overall text. As a result many new techniques have been proposed to improve embeddings of long documents.

2.3. Contrastive Learning

Contrastive learning has become an increasingly popular and effective representation learning approach. It aims to learn low-dimensional representations of data by contrasting between similar and dissimilar samples. The basic contrastive learning framework consists of selecting a data sample, called an "anchor" or "query"; a set of data samples similar to the anchor, called "positives"; and a set of data samples dissimilar to the anchor, called "negatives". These are then passed through a representation model with the objective to minimize the distance between the anchor and positive samples and maximize the distance between the anchor and negative samples, in the representation space. As depicted in Figure 2.3.

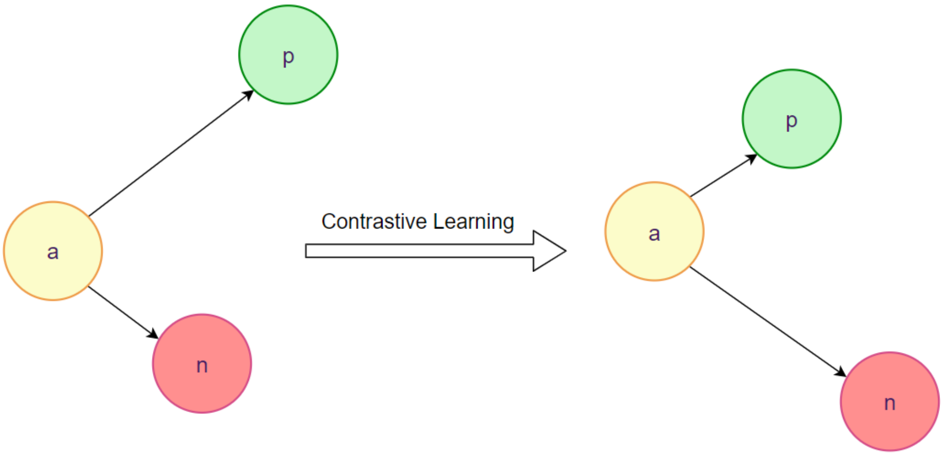


Figure 2.3.: Depiction of Contrastive Learning. When a model is trained via a contrastive loss function the representations that are labelled as positives (p) are pulled closer to the anchor (a) while the negative (n) representations are pushed apart.

The selection of positive and negative samples is essential for learning a good model. These can be determined, for instance, by whether the samples have the same class label or come from the same distribution as the anchor point.

Apart from random easy negatives, hard negatives i.e., negative samples that are difficult to distinguish from the positive samples are important to force a model to learn features that more effectively differentiate between similar and dissimilar data points. This improves the robustness of the model and makes it better at generalizing to new data [34].

Different contrastive learning approaches are characterized by the choice of their loss function. A contrastive loss is low if positive samples are encoded to similar (closer) representations and negative examples are encoded to dissimilar (farther) representations. Three variants of the contrastive loss that are referred to in our work are described below and the notation used is as follows:

The anchor is denoted as $\boldsymbol{a}$, its corresponding positive as $\boldsymbol{p}$ and negative as $\boldsymbol{n}$. The set of positive and negative points w.r.t an anchor is denoted as $\boldsymbol{P}(\boldsymbol{a})$ and $\boldsymbol{N}(\boldsymbol{a})$ respectively. The function to measure a distance is $d(.,.)$ and to measure similarity as $sim(.,.)$. m is the margin hyperparameter for triplet loss while τ is the temperature hyperparameter for other contrastive losses.

Triplet Loss

Triplet Loss [35] is simple variant of contrastive loss which consists of one positive and one negative sample per anchor. The loss function, calculated as Equation 2.4, is minimized by pushing $d(\boldsymbol{a}, \boldsymbol{p})$ towards 0 and $d(\boldsymbol{a}, \boldsymbol{n})$ to be greater than $d(\boldsymbol{a}, \boldsymbol{p}) + m$. This encourages that dissimilar pairs be distant from any similar pairs by at least a certain margin value.

$$\mathcal{L}_{Triplet} = max(0, d(\boldsymbol{a}, \boldsymbol{p}) - d(\boldsymbol{a}, \boldsymbol{n}) + m) \tag{2.4}$$

N-pairs Loss

N-pairs Loss [36] considers one positive and many negative samples for an anchor and is calculated as:

$$\mathcal{L}_{N-pairs} = -\log \frac{\exp(sim(\boldsymbol{a}, \boldsymbol{p})/\tau)}{\sum_{\boldsymbol{p} \in P(a)} \exp(sim(\boldsymbol{a}, \boldsymbol{p})/\tau) + \sum_{\boldsymbol{n} \in N(a)} \exp(sim(\boldsymbol{a}, \boldsymbol{n})/\tau)} \tag{2.5}$$

SupConLoss

The Triplet Loss and N-Pairs loss were extended to consider multiple positives and multiple negative samples for a given anchor point [18]. This loss obtained state-of-the-art performance for the task of image classification. The loss is given as:

$$\mathcal{L}_{CL} = -\frac{1}{|P(a)|} \sum_{\boldsymbol{p} \in P(a)} \log \frac{\exp(sim(\boldsymbol{a}, \boldsymbol{p})/\tau)}{\sum_{\boldsymbol{p} \in P(a)} \exp(sim(\boldsymbol{a}, \boldsymbol{p})/\tau) + \sum_{\boldsymbol{n} \in N(a)} \exp(sim(\boldsymbol{a}, \boldsymbol{n})/\tau)} \tag{2.6}$$

2.4. Scientific data terminology

Scientific Document Representation learning is a way of learning representations of scientific documents using unsupervised and semi-supervised methods. We often have very large amounts of unlabeled training data and relatively little labeled training data. Training with supervised learning techniques on the labeled subset often results in severe overfitting. Semi-supervised learning offers the chance to resolve this overfitting problem by also learning from the unlabeled data. Specifically, we can learn good representations for the unlabeled data, and then use these representations to solve the supervised learning task.

A scientific paper typically contains a **title** and an **abstract** to communicate a succinct and comprehensive summary of the paper. They help researchers to filter out articles as per their requirements. After titles, abstracts are the most read section for scientific document discovery. They act as an important decision tool to distinguish papers from one another.

Moreover, many times they are the only openly available information about a paper as the full content is closed for access due to copyright restrictions. Hence, they act as a primary tool for learning the representation of a research paper and have been used in many works in this domain as shown in later chapters. In this work, we also use them for characterizing a paper.

Research papers are connected among one another through citations. In this work we use data from the Semantic Scholar [37] corpus to get the title and abstract along with the citations and references of research papers to construct a citation graph of the papers. Semantic Scholar is a free and open-access database supported by the Allen Institute for AI which is accessible via an API. It provides free, AI-driven search and discovery tools and indexes over 200 million academic papers. For each of these papers, it provides metadata information including the paper title, abstract, authors, citations, references, publication venues, year etc.

We adopt the terminology used by the Semantic Scholar API to define the citation relations. The outgoing citations of a paper i.e., the papers cited by the paper (a list of sources that are mentioned in the paper's bibliography) are called **references**. The incoming citations of a paper i.e., the papers that cite the paper (papers in whose bibliography the paper appears in) are called **citations**.

Along with the information about the citations and references of a paper, we also get the information about the intention of a particular citation from Semantic Scholar. We use these to classify finer-grained aspects of the paper relations. The citation intents can be classified into three categories: **background**, **methodology** and **result**. These are summarized in Table 2.1. A citation relationship can belong to one or more of the three intentions. An example in Figure 2.4 depicts the citation intent relations of the SPECTER paper with some other papers that cite it or are referenced by it.

Intent Category	Description
Background information	The citation states, mentions, or points to the background information giving more context about a problem, concept, approach, topic, or importance of the problem in the field
Method	Making use of a method, tool, approach or dataset
Result comparison	Comparison of the paper's results/findings with the results/findings of other work

Table 2.1.: The definition of the citation intent categories taken from SciCite [38] that are used to characterize our aspects

It is important to note that these intent classifications were not manually curated but were automatically generated by a classification model proposed by Cohan et al. [38]. It is a multitask model that incorporates structural information of scientific papers into citations for automatic classification of citation intents. The model takes citation contexts, which are text spans in a citing paper describing a referenced work, as input. These are encoded as a concatenation of the GloVe and ELMo embeddings of their words which are passed through

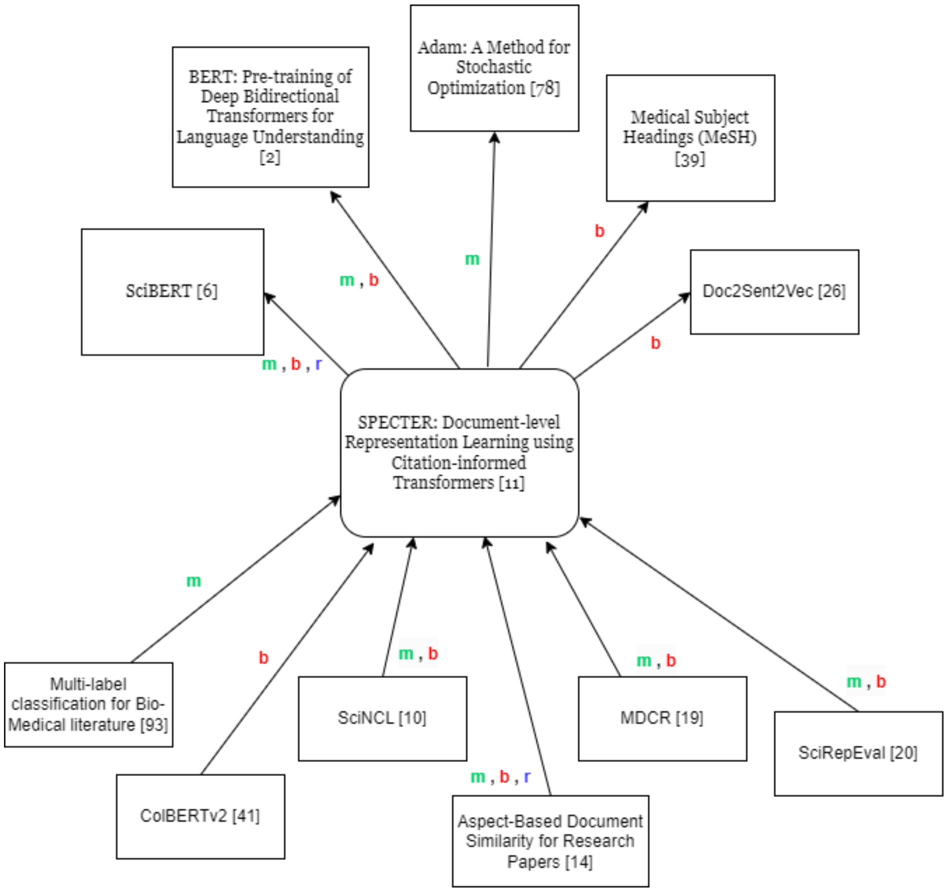

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [2]
Adam: A Method for Stochastic Optimization [78]
Medical Subject Headings (MeSH) [39]
SciBERT [6]
Doc2Sent2Vec [26]
m , b
m
b
b
m , b , r
SPECTER: Document-level Representation Learning using Citation-informed Transformers [11]
m
b
m , b
m , b
m , b
Multi-label classification for Bio-Medical literature [93]
SciNCL [10]
MDCR [19]
SciRepEval [20]
ColBERTv2 [41]
m , b , r
Aspect-Based Document Similarity for Research Papers [14]

Figure 2.4.: Example of citation intent relations of the SPECTER paper as given by Semantic Scholar. ’m’ represents methodology citations, ’b’ represents background citations, and ’r’ represents result citations.

a BiLSTM network to get a unified embedding. An MLP is trained to predict citation intents on these embeddings with an additional objective of two auxiliary tasks:

1. predicting the section title in which the citation occurs
2. predicting whether a sentence needs a citation.

The accuracy of these models is reported as 67.9% F1 on the ACLARC citations benchmark and 84% F1 on the SciCite benchmark.

Hence, this dataset is not gold-standard and the shortcomings and biases of this model will also be propagated in our models. Also, there are many citation relations with no citation intent data given because of no access to the full text of the papers. We classify these relations as **miscellaneous**.

2.5. Evaluation Benchmarks

A machine learning model is as good as its features. To test how good the embeddings learned by our models are, we use them as features to check how well they capture the aspects, the domain, the meaning etc. In this section we describe the evaluation benchmarks previously used to evaluate scientific document representations that were also used in our work.

2.5.1. SciDocs

SciDocs [11] is a standard evaluation framework to measure the effectiveness of scientific paper embeddings. It consists of a suite of seven different tasks and 223,932 documents that test the embeddings for their ability to be used for document classification, citation prediction and recommendation.

Document Classification

It is important for a good scientific document embedding to be able to capture the class of the document. For this, two tasks are considered:

1. Medical Subject Heading Classification (MeSH) [39]: Given a set of 23K academic medical papers, the task is to predict the disease classes that the paper belongs to, from a total of 11 categories.
2. Paper Topic Classification (MAG) [40]: This task aims to predict the main scientific field or topic of the given 25k documents. Each document belongs to one of 19 different topic classes which were retrieved from the Microsoft Academic Graph.

Citation Prediction

In this test, we assess how well the embeddings can capture the relatedness between two documents. This is done using two tasks:

1. Direct citation prediction: In this task, the model is asked to predict which papers are cited by a given query paper from a given set of candidate papers. The evaluation dataset includes approximately 30K total papers from a held-out pool of papers, consisting of 1K query papers and a candidate set of up to 5 cited papers and 25 (randomly selected) uncited papers. The task is to rank the cited papers higher than the uncited papers. For each embedding method, we require only comparing the L2 distance between the raw embeddings of the query and the candidates, without any additional trainable parameters.

2. Co-citation prediction: Here, the goal is to predict whether a given paper is so highly related to a candidate paper that they are likely to be co-cited. The dataset consisting of 30K total papers was constructed like the direct citations task with the ground truth label being whether the two documents were cited together or not.

User Activity

In the above tasks, citations act as the ground truth to indicate whether two documents are related or not. However, all related documents are rarely cited. Another classical way to capture relatedness is via user activity. Multiple users retrieving the same documents suggests relatedness between the documents. To capture this, the logs of user sessions from a major academic search engine are obtained and two tasks are defined:

1. Co-Views: In this task, the academic papers that are frequently viewed by users in the same browsing session are considered as related. The dataset consists of approximately 30K papers. A test set of 1K random papers consisting of up to 5 frequently co-viewed papers and 25 randomly selected papers is used. The embedding model is expected to rank the co-viewed papers higher than the random papers by comparing the L2 distances of raw embeddings.

2. Co-Reads: In this task, the academic papers for which a user clicks to access the PDF of a paper from the paper description page in the same browsing session are considered to be related. Accordingly, a dataset also of size 30K is curated in the similar way as above.

Recommendation

In the recommendation task, the ability of an embedding to be used in a recommendation system is assessed. Based on embedding cosine similarity, the recommendation system should be able to rank a given list of papers for recommendation to a query paper. A dataset of user clickthrough rates is used for this task which consists of 6K clickthrough events from a public scholarly search engine. The examples are temporally partitioned into train (4K examples), validation (1K), and test (1K) sets.

2.5.2. MDCR

Medic et al. [19] argued that the recommendation tasks in SciDocs are too narrow to evaluate the effectiveness of an embedding. They criticize that its relatively small, randomly generated candidate pool for the recommendation tasks is not representative enough of a realistic recommendation system with millions of papers in the candidate pool. They also note that the majority of SciDocs queries (over 70%) come from a single domain (computer science), making it a predominantly computer science-oriented benchmark. Hence, they propose a new benchmark called Multi-Domain Citation Recommendation dataset (MDCR) to evaluate representations on large, challenging, and multi-domain candidate pools. In contrast to SciDocs (which contains 1k queries that are paired with a candidate pool of size 30, with 5 positive candidates and 25 randomly generated negative candidates), MDCR contains 200 queries for 19 scientific domains, where each query is paired with a set of 60 negative candidates and 5 cited articles as positive candidates. In total it consists of a total of 200,033 documents.

For each query the negative candidates were selected such that:

1. they were difficult candidates for the models BM25, SPECTER and SciNCL. This means that the models ranked them highly even though they were not cited by the query. (Model-Based Selection)
2. they were in the neighbourhood of the query in the citation graph but were not directly cited by the query (Graph neighbors-based selection)
3. they were among the most cited articles in the scientific field of the query (Citation count-based selection)
4. random selection

The benchmark was used for evaluating 5 different semantic transformer-based SDR models and the simple lexical BM25 (Robertson and Walker, 1994) model (the results are reported in Table 6.8). Contrary to expectations, BM25 outperformed transformer-based models in most fields. Among the transformer-based models, SciNCL performed the best.

2.5.3. SciRepEval

Recently, Singh et al. [20] introduced SciRepEval, a new, bigger, and more diverse benchmark dataset for evaluating scientific document representations. They point out that four of the tasks in SciDocs are highly correlated (greater than 0.99) and are therefore insufficient in evaluating the generalizability of the embedding models. In contrast, SciRepEval contains 25 challenging and realistic tasks including SciDocs. The tasks are categorized into four formats: classification, regression, proximity-based ranking, and ad-hoc search.

They demonstrate that state-of-the-art models like SPECTER and SciNCL struggle to generalize across task formats, and that simple multi-task training fails to improve them. To address this problem, they propose a new approach that instead of condensing all relevant information of a document into a single embedding, learns four different embeddings per document, each tailored to the different formats. The resulting embeddings outperformed the single embeddings by up to 1.5 absolute points. To achieve this, they fine-tune the existing models with task-format-specific control codes and adapters in a multi-task setting.

Contrary to SciDocs, which contains very few training samples for each task and therefore suboptimal for evaluation, in SciRepEval, eight of the largest tasks across the four formats are used during training, while the rest of out-of-train tasks are reserved for evaluation. This enables the study of multi-task approaches, rather than relying solely on the citation signal. The training datasets in SciRepEval have at least 100,000 instances and also have a large-scale representation in multiple domains.

The formats and associated tasks are described below:

Ad-Hoc Search

It is a critical mechanism for paper discovery in practice. In ad-hoc search tasks (QRY), we are given a textual query and the task is to rank a set of candidate papers by relatedness to the query. SciRepEval uses multiple real-world data sets for training and evaluation on this task. It is evaluated by ranking the candidate papers by increasing the Euclidean distance between the query embedding and the candidate paper embeddings using pytrec_eval. The datasets used are summarised below:

- Used only for evaluation:
  - TREC-CoVID [42]: a biomedical challenge task that ranks papers from CORD-19 dataset in response to textual search queries.
  - Feeds dataset: taken from a scholarly paper recommendation system, where the user-specified feed name is considered as the topic query, and the goal is to rank the papers the user has annotated as relevant to the feed above those annotated as irrelevant.
- Also used for training:
  - Search: the dataset contains more than 700,000 click-through events from a scholarly search engine

Proximity

Similar to ad-hoc search, proximity tasks (PRX) involve ranking a set of candidate papers by their relatedness to a query, except the query in this case is not textual but instead a paper. Proximity-based tasks form a basis for paper-based retrieval and recommendation, and for estimating paper similarity for use in applications like author disambiguation. The datasets used are summarised below:

- Used only for evaluation:

  - Paper-Reviewer Matching: Candidate reviewers are ranked by expert annotators based on the similarity of their papers to the query paper to be reviewed.
  - SciDocs Citation and user Activity Prediction Tasks
  - Feeds: uses the recommendation system as above, but instead of a textual query, one or multiple relevant papers serve as queries.

- Also used for training:
  - Same Author Prediction: Papers are ranked according to if they are likely to have the same author as the query paper.
  - Citation Prediction: Papers are ranked according to their likelihood of being cited by the query paper.
  - Influential citation prediction: Papers cited by the query paper are ranked according to their influence. They define a paper as influential if it is cited in the text of a single paper more than four times.

Classification

In classification tasks (CLF), a paper is taken as input and the output is a topical category, which is a foundational task for document organization and discovery. Support vector classifiers are used to evaluate these models. Also, to better understand how embeddings perform in data-scarce regimes, manually annotated gold-labeled few-shot versions of the classification datasets are used. The datasets are summarised below:

- Used only for evaluation:
  - SciDocs MAG and MeSH classification
  - Biomimicry: to predict whether a paper is about biomimicry or not [43]
  - DRSM: to predict the 'Disease Research State' category for medical papers [44]
- Also used for training:
  - MeSH Descriptors: to predict the hierarchical categories of biomedical publications
  - Field of Studies classification (Not included in this work due to technical issues with the provided dataset)

Regression

The goal of regression tasks (RGN) is to predict a continuous quantity for a given paper.

- Used only for evaluation:
  - Tweet Mentions: to predict the number of tweet mentions
  - Peer Review Ratings: to predict Peer Review Score
  - Maximum h-Index of authors: to predict the maximum h-Index of the author

- Also used for training:
  - Citation Count Prediction: to predict the number of citations
  - Publication Year: to predict the year of publication

2.6. Evaluation metrics

Here we present the metrics used in the thesis to evaluate the above classification and recommendation tasks.

Classification

In a classification task, we predict the class of a data point and compare it with the actual class of the point. To quantify the classification accuracy the precision, recall and F_1 metrics are used.

- Precision quantifies the number of positive class predictions that actually belong to the positive class.
- Recall quantifies the number of positive class predictions made out of all positive examples in the dataset.

The F_1 score is the harmonic mean of precision and recall and is a way to balance the concerns of the two to get a single metric. The formulae for calculating these are:

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

where TP means True Positives i.e., the number of items correctly predicted as belonging to the actual class; FP means false positives i.e., the number of items incorrectly predicted as belonging to the class; FN means false negatives i.e., the number of the items which were not labelled as belonging to the positive class but should have been.

Recommendation

The recommendation tasks to evaluate our representation models work as follows:

For a given query document, a set of candidate papers are ranked for recommendation based on a similarity score between the query and candidate pairs. To evaluate the quality of these recommendations and rankings, the metrics used are:

1. Precision@k: It indicates the percentage of correct recommendations in the top k recommended items.

2. Recall@k: It indicates the percentage of overall correct recommendations that are included in the top k recommendations.

3. Mean Average Precision (MAP): It indicates how many relevant documents are ranked high in the recommendations. This metric sets a threshold for similarity and gives a binary relevance score for considering whether a document is relevant or not. It is useful in situations when ranking all relevant items high is important.

4. Normalized Discounted Cumulative Gain (nDCG): It indicates how many highly relevant documents were recommended by the algorithm. Instead of binary relevance, this considers the degree of relevance, i.e., how similar the query and candidate are, to compare the model's recommendations. In comparison with MAP, it penalizes lower placement of relevant items more. According to this metric highly relevant items should come before medium relevant items, which should come before non-relevant items.

We guide the reader to [45] for further information on these metrics.

3. Related Work

In this chapter, we discuss some of the prominent preceding works in this domain. These were referred to for inspiration, background knowledge, and methodology for this thesis.

Given the influx of information in the scientific world many contributions have been made to map scientific documents to useful vector representations. These vector representations are typically derived with the objective that their similarity in the embedding space reflects their semantic or functional similarity. Depending on the use case of the representations: if semantic similarity is desired, the document representations are learned from information on the contents of the documents; if proximity in terms of neighborhood in a citation network is desired, they are learned from the citation network; if both are desired, a hybrid combination of the two is used. Additionally, aspect-based embeddings were also proposed by some works to capture the different aspects of a document.

3.1. Text-based

The text in scientific documents is different from the general text in terms of structure and they often contain special scientific terms and other jargon. This means that the models and tokenizers used for embedding general documents might be inefficient. So, the embeddings are learned from the topics, textual content, and/or the metadata information of the documents.

Early methods utilized manually curated rules and keywords, bag-of-words, and TF-IDF vectors to represent the text in scientific documents such as [46]. Topic models like LDA which represent a document as a mixture of topics, where each topic is a probability distribution over words, were also used [47, 48].

Then the course of work moved to more semantic models using word, paragraph, and document embeddings such as [49–51]. Instead of using them out-of-the-box, these were also specialized to learn domain-specific embeddings for scientific data [52–55].

After the success of transformer-based PLMs like BERT which are able to capture useful contextual information of a document, they became the de facto model for representing textual data. However, PLMs are trained on general domain corpora such as news articles and Wikipedia and are not familiar with the in-domain vocabulary and nuances of scientific documents. Several methods have been adopted to adapt PLMs to the scientific corpora with the goal of enhancing performance on tasks requiring domain knowledge.

To inculcate knowledge from different scientific domains models like: BioBERT [56] - a variant of RoBERTa for the biomedical literature, PubMedBERT [57] a BERT variant trained on PubMed abstracts, ClinicalBERT [58] - trained on clinical notes, MatSciBERT [59] - trained on material science literature, etc., were introduced.

Apart from specific domains, general models for scientific documents were also introduced. One such widely used model, SciBERT [6] is a pretrained language model based on BERT, that was trained on a huge corpus of 1.14M scientific papers. This corpus consisted of 18% papers from the computer science domain and 82% from the broad biomedical domain. A new vocabulary SCIVOCAB was constructed based on the frequently occurring words in the given scientific corpus. It only overlapped with the BERT's original vocabulary by 42%, indicating substantial differences in frequently used words between scientific and general domain texts. Both cased and uncased versions of the vocabularies were produced, and it was observed that the uncased models performed better than the cased models for most tasks.

OAG-BERT [60] introduced another variant pretrained on the Open Academic Graph where along with textual data (title and abstract), they incorporated other entities including paper, author, concept, venue, and affiliation. With this, they were able to produce better results on tasks like author name disambiguation and venue/field of study prediction compared to SciBERT.

Following SciBERT, S2ORC-SciBERT [61] was introduced that was trained on the S2ORC corpus which consisted of 81.1M academic papers resulting in a corpus of 16.4B tokens, nearly five times larger than the corpus for SciBERT. The corpus also corpus consists of a more balanced distribution of papers across diverse academic disciplines such that biomedical (42.7%) and computer science (7.2%) papers only comprise half the corpus. Unlike SciBERT, they also identified figure captions, table text and captions, headers, footers, footnotes, and ill-formed paragraphs (such as those containing too many symbols) and excluded them from the pretraining corpus. The Jaccard index between the S2ORC-SciBERT and SciBERT vocabularies is 0.536. They showed comparable results to SciBERT in a range of downstream tasks.

One approach that has recently gained attention is the combination of PLMs with contrastive finetuning to improve the semantic textual similarity between document representations. These contrastive methods learn to distinguish between pairs of similar and dissimilar texts. For instance, Tan et al. [13] proposed a coupled text pair embedding (CTPE) model which learns SDRs by segmenting a document into two parts (such as the title and abstract) to form a coupled text pair, embedding them with word and sentence embedding methods, and then using contrastive learning (triplet loss) to bring embeddings of text pairs from the same document closer and different documents apart.

For tasks where reasoning and a deeper understanding of text and language generation are needed like question answering and citation prediction, recently a decoder-only large language model Galactica [62] was also introduced which was trained on a large and highly curated corpus of 48 million papers, textbooks, and lecture notes, millions of compounds and proteins, scientific websites, encyclopedias, etc.

3.2. Citation-based

In the case of scientific documents, embeddings can also be generated based on the valuable structure present in the citation network. Citations play a crucial role in academia as they help position a new publication. They assist researchers to navigate through research work and find new information. Consequently, representing documents with network embedding algorithms has been a popular strategy. These embeddings capture the latent features of papers based on their citation patterns.

Common methods for graph embedding such as those described in chapter 2 have also been utilized for representing citation graph embeddings. Additionally, [63] and [64] proposed random walk-based methods to represent scientific documents for the task of Scientific impact prediction. [65–67], proposed a GCN-based approach for citation recommendation. Node2vec algorithms were also employed to generate citation embeddings [10, 17, 68].

3.3. Hybrid

While text-based embeddings are useful for capturing the contents of a document, they don't capture its relationship with other documents, thus not leveraging the additional information that naturally comes with scientific documents. So, for e.g., for recommendation tasks, considering only the content restricts recommending articles that are useful for the user but are not textually similar to the query such as articles from another domain. Similarly, by not leveraging the contents of a document, purely citation-based approaches suffer from information loss. This is because citations have a bias towards specific articles based on an author's preferences and knowledge. All the references of an article may not be very informative, and some important related articles may not be referenced such as contemporaneous works.

To combine the power of both contextual and citation knowledge, hybrid embeddings are suggested and are applicable in most use cases. With these, the documents are characterized both by their position in a citation graph and their textual content. The textual and citation representations can be learned separately and then combined together or they are learned together jointly.

Earlier, text embeddings were combined with citation embeddings to generate hybrid embeddings. In 2015, Yang et al. [69] proposed Text-Associated DeepWalk (TADW) that generates paper embeddings by factorizing the DeepWalk matrix with TF-IDF matrix. Similarly, Paper2Vec [70] and VOPrec [71] used paragraph vectors and random walk-based citation embedding methods to produce SDRs. In P2V [72] and LDE [73], a supervised model was proposed to learn embeddings from labeled linked documents. Three kinds of content relations were modeled for learning paper embeddings for linked documents: word–word–document, document-document, and document-label relations. They optimize the weighted average of three different objective functions to maximize co-occurrence in these relations using the methods described in [70].

More recently, contrastive learning has emerged as a popular method to generate SDRs with both intra and inter-document signals. Citeomatic [74] proposed a triplet loss based document embedding model, whereby positive samples are papers cited in the query. Easy negatives are random papers not cited by the query. Hard negatives are references of references – papers referenced in positive citations of the query but are not cited directly by it. They also used a second type of hard negatives, which are the nearest neighbors of a query (in terms of document embeddings of the papers from the previous epoch being close) that are not cited by it. They used bag-of-words for its textual features.

Along similar lines, Cohan et al. [11] presented SPECTER (Scientific Paper Embeddings using Citation-informed TransformERs), a model that uses contrastive learning to incorporate inter-document relatedness signal to improve contextual SciBERT embeddings. In this paper, titles and abstracts are concatenated and fed into SciBERT and the model is retrained with a triplet loss objective with positives again being papers cited in the query, easy negatives are random papers not cited by the query and hard negatives are references of references – papers not cited by the query paper but cited by the references of the query paper, refer Figure 3.1. The assumption is that these papers are somewhat related to the query paper, but typically less related than the cited papers. For a query 5 positives, 3 easy negatives, and 2 hard negatives were sampled. 684k triplets for 146k query documents and 145k triplets for 32k query documents were used for training and validation respectively. An average performance of 80.0 was reported SciDOCS benchmark, which outperformed SciBERT and other baselines. An online A/B testing was also performed on a production recommender system that uses textual similarity measures to rank relevant documents and SPECTER embeddings improved clickthrough rates by 46.5%. Apart from that they performed several ablation studies and found reduced performance from adding additional fields such as venues, and authors; encoding title and abstract separately and concatenating; removing hard-negatives; training only on titles; using BERT-Large (general domain) instead of domain-specific SciBERT. A task-specific fine-tuned version of SciBERT was also found to be worse than SPECTER.

Building up on SPECTER, Ostendorff et al. [10] released SciNCL in which a different approach was chosen for representing positives and negatives. They argued that discrete citation relations to generate contrast samples enforce a hard cutoff for similarity and propagate human biases of which papers are similar. This is because scientific papers can be similar even without a direct citation link between them, which is essentially the core problem of finding relevant research. Hence, the representation should be able to link semantically similar papers even if they are not connected through citations. Moreover, they point out that SPECTER defines papers cited by the query as positives, while papers citing the query could be treated as negatives. For e.g., in the Figure 3.1, paper P1 was not specifically marked to not be considered as an easy negative. This means that positive and negative learning information collides between citation directions, which [75] have shown to deteriorate performance. Lastly, they found that 40.5% of SciDocs' papers leak into the SPECTER's training data. To overcome these limitations, they employed controlled nearest neighbour sampling over citation graph embeddings for sampling positives and negatives for contrastive learning. Given a query paper in a citation graph embedding space, 5 positive papers are sampled from the close neighbourhood around the query embedding, 3 easy negatives which are very distant from

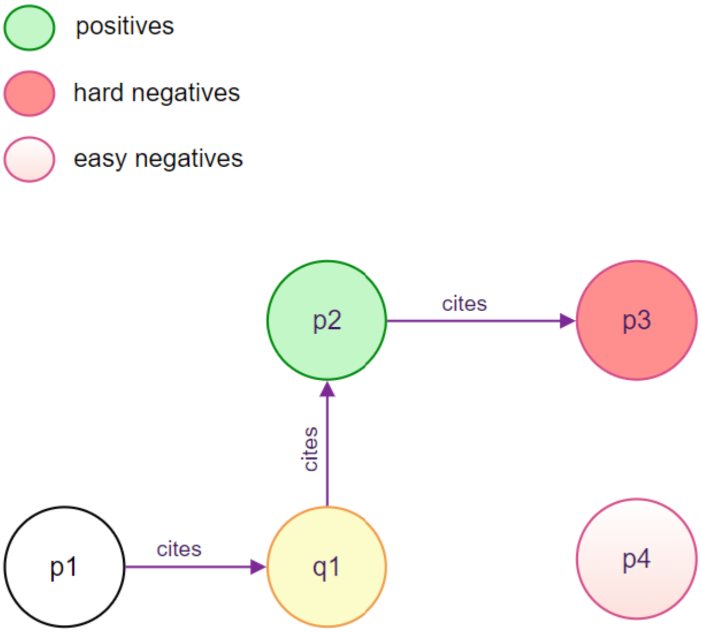


Figure 3.1.: The positive and negative sampling strategy as defined by SPECTER

the query, 2 hard negatives such that they are close to potential positives but still farther from easy negatives by a certain margin such that they do not collide with positives Figure 3.2. They also remove leaked training data from SPECTER's training set and substitute it from the S2ORC corpus [61] to train their models. They document state-of-the-art performance on SciDocs beating SPECTER in 9 out of 12 metrics.

Another work, [76] also used contrastive learning to encourage document pairs that are semantically relevant to be closer and structurally unrelated to be far apart in the representation space. Furthermore, the separation margins between the documents were varied flexibly to encode the heterogeneity in relationship strengths. The structural similarity was determined by whether the two documents share the same topic code or are connected through citations or not. This was represented as an adjacency matrix and similar documents were found using the page rank algorithm. The document embeddings were learned using a quintuplet loss with four input branches: the anchor, the structurally similar documents, the structurally dissimilar documents, and the semantically similar documents.

3.4. Aspect-based

New measures to deal with aspect-based document similarity for research papers have also gained popularity. These capture different arguments, sub-topics, citation context, and reasoning, etc. in the representations for a more fine-grained document similarity comparison.

Earlier approaches derive aspect-based document similarity by splitting documents into aspect-specific segments and computing a segment-level similarity. E.g. in [77] abstracts of papers are segmented into four classes depending on their research aspects: background, purpose, mechanism, and findings. Next, they represent a paper with four vectors, each derived from the corresponding segment's content. Computing the cosine similarity between the segment vectors allows the retrieval of similar papers for a specific aspect. [78] apply this approach to biomedical research papers. However, splitting documents into segments breaks the document coherence and can hurt performance as the individual segments can retain insufficient context from the whole document to produce meaningful representations.

Ostendorff et al. [14] extended traditional document similarity to aspect-based document similarity using a pairwise multi-label, multi-class classification of related documents to classify: under which section heading of one of the papers, the other document is cited. This acts as an aspect-level signal for distinguishing between different citation relations. However, pairwise similarity is not scalable, and getting fixed, condensed embeddings that can be indexed and retrieved in at least O(n) is desired for documents specially for tasks like recommendation and retrieval.

Other methods have tried to ingrain the aspect information into the embeddings themselves. For instance, [17] proposed to learn hybrid embeddings of papers by incorporating both contextualized text and citation network position information weighted based on citation

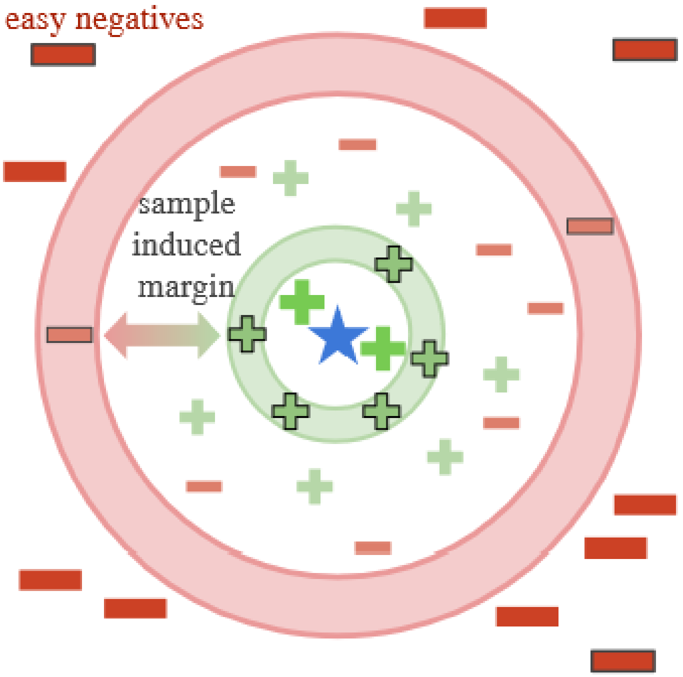


Figure 1: Starting from a query paper ★ in a citation graph embedding space. Hard positives ✚ are citation graph embeddings that are sampled from a similar (close) context of ★, but are not so close that their gradients collapse easily. Hard (to classify) negatives ▭ (red band) are close to positives (green band) up to a *sampling induced margin*. Easy negatives ▬ are very dissimilar (distant) from the query paper ★.

Figure 3.2.: The positive and negative sampling strategy as defined by SciNCL

Model	Positive selection strategy	Negative selection strategy
SPECTER	direct citation	citation-of-citation
SciNCL	citation embedding based neighbors	distant nodes in the citation embedding graph
ASPIRE	co-citation	random

Table 3.1.: Summary of the sampling strategy used for selecting the positive and negative candidates by previous works

intent. For this, SCIBERT embeddings of paper abstracts were concatenated with three different node2vec embeddings that are individually trained for three different citation intents i.e., background, method, and result. In another line of work [15], instead of learning single vector representations, multiple specialized aspect-specific representations were learned using computer science papers where the aspect spaces were represented as the task, dataset, and method of the papers. Shesher et al. [12] introduced ASPIRE to learn aspect-based document embeddings with contrastive learning. Instead of citations, they use co-citation as a signal for relatedness and the citing sentences as aspects that describe how the co-cited papers are related (e.g., similar methods or findings, related challenges or directions, etc.). A triplet loss objective is used where the positives are papers co-cited together in the same sentence and negatives are random papers other than positives. The distance between the positives and negatives is determined by how the sentences from the abstracts of the contrasting pairs align with the citing text (aspect). They showed improved performance compared to SPECTER and SciNCL for document similarity.

4. Methodology

4.1. Introduction

Our goal is to experiment with new ways to represent a given research paper P as a dense vector v that best represents the paper and can be effectively consumed in downstream tasks even without further fine-tuning. In this vector, we wish to capture both the contextual and citation information of the paper. The intuition behind this is that all papers that are connected through citations may not be very similar, apart from maybe quoting a fact, and two papers that may be very similar w.r.t. the contents of their title and abstract may not be connected through citations. Thus, we wish to find a good balance between the two. Additionally, we want to integrate aspect information into these embeddings. The underlying assumption here being that a paper shows different aspects when interacting with different neighbours and should own different embeddings accordingly. For learning such a representation model, we draw inspiration from SPECTER and build on top of that by experimenting with different model configurations.

To learn aspect-aware embeddings, we need to identify aspect-specific relations between the documents. Due to the constraints of such annotated data, we consider citation intent as the aspect term to align embeddings in their respective aspect spaces. As a result the embeddings generated would be such that they are closer for papers that are cited/referenced with the same intent than papers cited with a different aspect. The hypothesis here is that for tasks like recommendation and classification we want the papers connected directly with the same citation intents to have closer representations.

In the first experiment, we replicate the methodology of SPECTER but unlike SPECTER, which uses only references as positives, our data contains both citations and references as positives. We also remove any data points that overlap with the test sets. Then in experiment 2, we train the above SPECTER(Undirected) with a general contrastive loss function instead of a triplet loss. In experiment 3, we introduce our model SciAspect to learn aspect-aware embeddings and try different variations to improve it. In experiment 4, we train a hybrid model called SciAspectHybrid, which combines the best models found in experiment 2 and experiment 3. We anticipate that this model will learn embeddings that are aware of both its aspect-specific local neighborhood as well as its general global neighborhood. Similar to this, in Experiment 4 we try another way to preserve the local neighbourhood of the anchor and incorporate the aspect-aware neighbourhood knowledge. This is another variant of SciAspect in which instead of considering the immediate neighbours of an anchor document connected with a different aspect as negative, we consider the reference-of-reference of the anchor in another aspect as negative. The experiments are explained in detail in chapter 5.

4.2. Experimental Setup

Like SPECTER, we take the title and abstract of a paper as a proxy for the paper itself. This way we can efficiently store the data and their combined length (in most cases) is suitable to feed to a BERT-based encoder (< 512 tokens). Each paper is given as input to the encoder as a concatenation of its title and abstract, which are converted to a sequence of tokens, separated by the [SEP] token. Then the pooled representation corresponding to the [CLS] vector of the last layer of the encoder is obtained as the contextual embedding for the paper. Given a paper's title and abstract, this is achieved as:

$$z_i = \text{SciBERT}([\text{CLS}]\ \text{title tokens}(i)\ [\text{SEP}]\ \text{abstract tokens}(i))[\text{SEP}]$$

At the time of training, we take a combination of anchor papers and their corresponding positive and negative papers as input in a batch, tokenize and encode them with the SciBERT model and pass the resulting contextual embeddings to a contrastive loss function. In addition, for the aspect-based models, after the SciBERT encoder, a separate linear layer is added for each aspect to project the contextual embeddings onto different aspect spaces and then passed to the loss function. With each training iteration, the input batch is passed through the model and parameters of the SciBERT layer (and the projection layer in case of aspect-based models) are updated through backpropagation in order to minimize the objective loss function. Through this we encourage the model to learn to generate representations by maximizing agreement between the anchor and positive pairs (and minimizing agreement the anchor and negative pairs) in the latent space via the contrastive loss.

At the time of inference, the title and abstract of a paper are passed through only the encoder layer of the learned model and its pooled output representation is the new representation of the paper. In this case, the model does not need any citation information about the input paper and thus can be used to generate embeddings for new, yet to be cited papers.

4.3. Dataset Overview

To be consistent with previous work, we used the same anchor documents used by SciNCL (i.e., SPECTER without leakage) to train and validate our models. We then queried the Semantic Scholar API to retrieve the title and abstract for each of the training (total 132k) and validation documents (total 19k), along with their citations and references and their corresponding intents. Subsequently, for each reference we retrieved their references along with their intent. Hence, generating a local citation graph for each anchor paper as shown below.

$$\text{Citation} \rightarrow \text{Anchor paper} \rightarrow \text{Reference} \rightarrow \text{Reference}$$

This choice of data representation was carefully taken, considering several factors. Having a graph of all documents, as in SciNCL, and learning citation embeddings for determining the positives and negatives from the respective neighborhoods of the anchor paper has proven to be the state-of-the-art approach, however, this method is very expensive both memory-wise and computationally. On top of that, querying the Semantic Scholar API for getting the citation intent information for all the connections in the graph is not feasible. So, we stick to the dataset curation, as in SPECTER where for each anchor paper, its references are considered as positive while the reference of references are considered negative. However, we also add the citations of an anchor paper as positives. This is done because considering citations avoids the possibility of collisions in SPECTER as described in [10]. Also, considering just references and reference-of-references limited the number of connected papers, many of which did not have any associated intent information, and thus by adding direct citations along with their intents, we were able to add more possibilities for finding positives and negatives for a given query paper. Another important advantage of using this training data is that it is free of leakage, as it was made sure that there is no overlap between the training documents and the documents used to test on both our evaluation benchmarks (as described in [10] & [19]).

Out of the 136,820 anchor documents published by SciNCL for training, 4376 documents were eliminated as they did not have an entry in the Semantic Scholar API. Also, 25% of the documents (33279) did not have any abstract. On analyzing the query documents, we found that more than 75% of them had title + abstract total token length less than 512 making them suitable for feeding to a BERT model. In addition, we note that 68% of documents were from the medicine field, 39% from biology and 37% from computer science, highlighting that the training data is highly skewed in representing data from various fields of studies[1]. Statistical summaries about these documents can be found in Appendix A. For aspect-specific data, all query ids with at least one related paper in each aspect (i.e., methodology, background, and result) were chosen to form the dataset.

Then with different mining strategies, as explained in the experiments below, a JSONL file is generated with each line containing the paper ids, titles and abstracts of an anchor document and its corresponding positive and negative documents.

4.4. Training Procedure

During training, the anchor document and its positive and negative documents are retrieved from the dataset in batches. They are then tokenized and initialized using the pre-trained tokenizer and encoder SciBERT (scibert-scivocab-uncased) [2] from HuggingFace's Transformers library [79]. This encoder consists of 12 Transformer layers and 109 million parameters. The pooled [CLS] representations of size 768 each, are then passed through the projection layers(if applicable) and then the contrastive loss is calculated. For experiment 1, an effective batch size of 32 was used (same as SPECTER) while the effective batch size was 8 for the remaining experiments.

For guiding the training, we use the AdamW optimiser [80] with $\epsilon = 1e^{-8}$ and an initial learning rate of $2e^{-5}$. The learning rate then follows a cosine schedule with a 100 warmup steps. All the models were trained for 2 epochs; each training epoch takes approximately 1-2

[1] many documents belong to multiple fields

[2] https://huggingface.co/allenai/scibert_scivocab_uncased

days to complete on the full dataset. We used Pytorch distributed training on 2 NVIDIA A40 GPUs using the 'ddp' distributed backend. The Weights and Biases API[3] was used to log and monitor the general and aspect-specific training and validation losses.

4.5. Evaluation

As discussed earlier, we evaluate our models on three benchmarks namely SciDocs, MDCR and SciRepEval. Each document in the these is represented with a title and abstract, which we pass through the encoder layer of our learned models to generate the embeddings for evaluation. As the goal is to develop generalized, broadly adaptable embeddings, it is important to note that instead of fine-tuning our models on the test tasks, we simply pass the embeddings as features for each task. Below, we describe the procedure for evaluating each task.

SciDocs

We used this dataset as a benchmark for distinguishing the performance in model variations in ablation studies and the best performing models were chosen for further evaluations. Evaluation on this benchmark was done using the implementation provided by [11]. For the classification tasks, a linear SVM is trained to classify the embedding vectors into different label categories. The C hyperparameter is tuned via a held-out validation set. For the user-activity and citation prediction tasks, a similarity score between the test query and the candidate documents (25 random negatives and 5 relevant/cited positives) is calculated. These scores are ordered from high to low similarity to generate a ranking for recommendation and the top 5 recommendations are compared with the positive candidates. This is then evaluated for the given metrics with the help of pytrec_eval [82], a tool for fast evaluation of a ranking method popularly used for evaluating the results of an information retrieval system. It indexes dense vector embeddings using a fast approximate nearest neighbor system, and at the time of retrieval, queries are encoded using the same model and similarity search is performed in the vector space. For the recommendation tasks, a feed forward ranking neural network is trained with the paper embeddings, papers' citations, titles, authors, publication dates and other metadata.

MDCR

In this benchmark 200 query documents are provided for each scientific domain along with the candidates carefully selected as described in chapter 2. We compute the Euclidean distance between the query and candidate paper embeddings to generate the final recommendation scores. Then the suitable evaluation metrics are computed based on these scores again using pytrec_eval.

[3] https://wandb.ai/

SciRepEval

In this dataset, the training of the classification, regression, ad-hoc and proximity models is done separately on specific datasets, as described in chapter 2 and the evaluation on these models is done using in-train test datasets and out-of-train test datasets. The classification problems are modelled using linear support vector classifiers. The regression models are modelled using linear support vector regression models. The proximity and ad-hoc tasks are modelled like above using pytrec_eval.

5. Experiments

5.1. Experiment 1 - SPECTER(Undirected)

First, we validate our dataset by replicating the SPECTER model but according to this dataset we also consider in-citations as positives and we ensure no data leakage into the test set. We call this model SPECTER(Undirected).

As in SPECTER, each training instance is a triplet of papers: a query paper a, a positive paper p and a negative paper n. For each query paper a total of 5 positives (direct connections of the anchor paper), 2 hard negatives (connection of connections of the anchor paper) and 3 easy negatives (random disconnected papers) are randomly sampled from the respective pool of candidates. The triplet loss objective as defined in Equation 2.4 with an L2 norm distance metric is used. The candidates that qualify for sampling in our case are as follows:

- Positive candidates:
    - All papers directly connected to the anchor paper i.e., references and citations
    - Excluded self references and citations
    - Oversampled if the required number of candidates is not found
    - Skip anchor papers altogether if no positive candidates
- Hard negative candidates:
    - All papers that are references of the anchors' referenced papers
    - Excluded papers that can also be directly connected to the anchor paper (avoiding citation loops)
- Easy negatives candidates:
    - All papers in the corpus excluding the anchor paper and papers connected to the anchor paper (i.e., positive candidates and hard negative candidates)

5.2. Experiment 2 - SPECTERCL: SPECTER(Undirected) but with a general contrastive loss

In early versions of loss functions for contrastive learning, only one positive and one negative sample is involved. The trend in recent training objectives is to include multiple positive and negative pairs in one batch. So, unlike SPECTER and other approaches that use triplet loss as

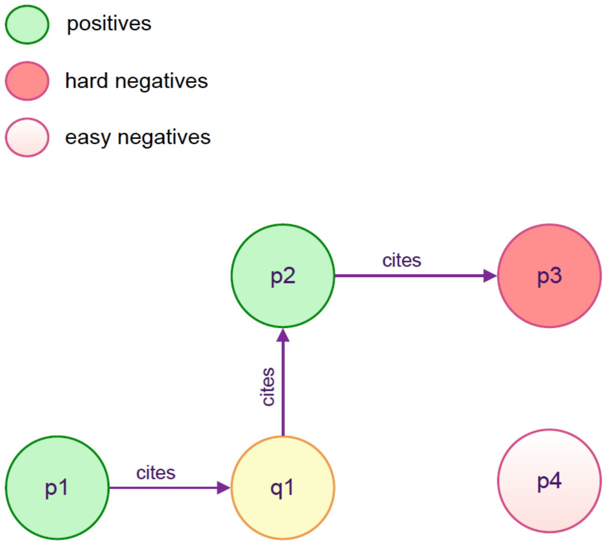


Figure 5.1.: Positive and negative mining strategy used in SPECTER(Undirected). Direct connections of the query are considered positive and reference-of-references are considered negatives. All other papers are considered easy negatives.

their training objective, in this work, we apply a contrastive loss inspired by [18] to train on multiple positive and negative instances for an anchor document at once. The loss function is given in Equation 2.6.

Now we train SPECTER (Undirected), taking this contrastive loss (CL) function as our objective. We call this model SPECTERCL. In this, we follow the same procedure as Experiment 1 but instead of a triplet, we combine the 5 positives and 5 negatives for each anchor as one training instance. $\tau = 0.05$ is chosen as the temperature parameter. Furthermore, two ablation studies are performed, to examine the behavior of our model and to choose the best model.

- **Experiment 2.1 - Effect of number of positives and negatives:** In this case study, we wish to know how varying the number of positives and negatives affects performance and which combination should finally be chosen to model further experiments.
  - **Experiment 2.1.1 - 1 positive, 1 negative:** [18] shows that the triplet loss is a special case of the contrastive loss and the latter subsumes the former. So in this experiment, we inspect if there is any significant performance difference between the triplet loss and the contrastive loss when passed only 1 positive and 1 negative for each anchor.
  - **Experiment 2.1.2 - 1 positive, 3 easy negatives:** Multiple works including SPECTER and SciNCL emphasize the importance of hard negatives when learning via contrastive learning. However, [18] suggests that the general loss is capable of performing implicit hard negative mining and can learn good representations even

without hard negatives. We evaluate this hypothesis by excluding hard negatives while training.

  - **Experiment 2.1.3 - different number of positives (1, 2, 4, 5):** This experiment studies the variation in performance with different number of positives i.e. 1(making the loss function equal to the N-Pair Loss Equation 2.5), 2 and 4.

- **Experiment 2.2 - Different temperature values (0.01, 0.05, 0.1, 0.5):** Temperature is an important hyperparameter in contrastive learning, that needs to be tuned for learning the best model. Smaller temperatures benefit training more than higher ones, but extremely low temperatures are harder to train due to numerical instability [18]. Here, the impact of changing temperature values is studied on the best model obtained above i.e., SPECTERCL with 1 positive and 5 negatives. We tested for the temperature values 0.01, 0.1 and 0.5.

5.3. Experiment 3 - SciAspect with L1 negatives

In this experiment, we examine a novel approach for representing the positives and negatives to learn scientific document representations via contrastive learning. We hypothesize that the positive and negative relations of an anchor document are different in different aspect spaces. A paper might cite another with different intentions and thus all cited documents should not be given equal weightage. The distances among the embeddings should be such that the model is able to distinguish between the different aspects/citation intents.

In SPECTER/SciNCL, all papers from the dataset are projected into the same embedding space. For training SciAspect, we project our anchor, positive and negative embeddings into 4 different aspect spaces - methodology, background, result, and miscellaneous/rest. These aspect spaces represent the citation intent in which the anchor and positives are similar, and the anchor and negatives are dissimilar. With the help of a contrastive loss objective, we wish to enforce that the anchors are closer to the positive papers and farther from the negative papers in the respective aspect spaces.

Each training/validation instance consists of an aspect category, an anchor document, at most 5 documents that relate to the anchor in the given aspect (positives), at most 10 documents related with the anchor but in a different aspect (hard negatives) and 3 random disconnected papers (easy negatives) refer Figure 5.2. These are then passed through the SciBERT encoder and projected into the different aspect spaces using a simple Linear projection head. The parameters of the model are then updated based on the contrastive loss function used in the previous experiment.

The strategy to define the pool of candidates from which our positives and negatives are randomly sampled for training is given as follows:

- Positive candidates:
  - All papers directly connected to the query paper in the given aspect

  - Excluded self references and citations
  - Skip query altogether if at least 1 positive candidate for each aspect was not found

- Negative candidates:
  - For the methodology, background, and result aspects:
    - All papers directly connected to the anchor paper but in a different aspect, excluding all anchor papers and positive candidates (as papers can be related in multiple aspects).
    - Easy negatives: like above experiments, it is a set of all papers in the corpus excluding the anchor paper and the papers connected to the anchor paper (i.e., positive candidates and the negative candidates chosen above)
  - For the miscellaneous aspect:
    - Since in this aspect, it is not certain how the documents cited by the anchor are connected to it and it could be any of the other aspects, these documents cannot be considered as negatives for other aspects. Also, the only certain negative documents here are the easy negatives.

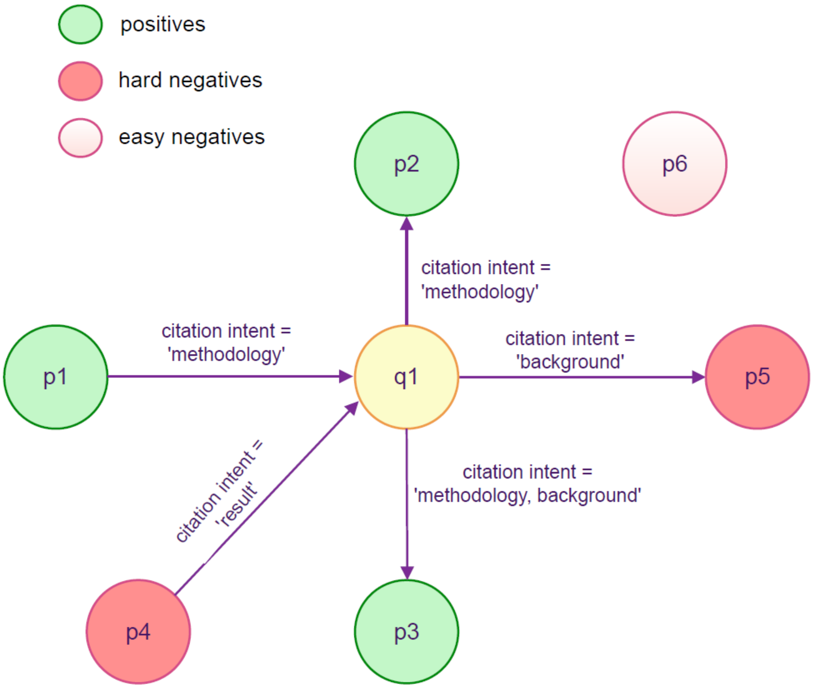

positives
hard negatives
easy negatives
p2
p6
citation intent = 'methodology'
citation intent = 'methodology'
citation intent = 'background'
p1
q1
p5
citation intent = 'result'
citation intent = 'methodology, background'
p4
p3

Figure 5.2.: Positive and negative mining strategy used in SciAspect: Example of positives and negatives for the **methodology** aspect.

Note that SciAspect only needs information about the local, one hop neighborhood of the anchor document (which we call Level 1 or L1) thus eliminating the need for looking at further indirect connections as in SPECTER and SciNCL. The high-level overview of our proposed model is shown in Figure 5.3 and described as follows:

$$\{q, p, n, i\} \rightarrow Enc(.) \rightarrow \{z_q, z_p, z_n\} \rightarrow Proj_i(.) \rightarrow z_q^i, z_p^i, z_n^i$$

where q is a query/anchor, p are the positives, n are the negatives, and i is an aspect category $\in \{methodology, positive, negative, miscellaneous\}$. $Enc(.)$ is an encoder, in our case SciBERT, that maps the input documents to contextual vectors. $Proj_i(.)$ is a Projection Layer that projects the encoded inputs to the aspect representation. The Linear Projection Layer maps a document embedding d to a

$$z_d^i = W_i d + b_i,$$

here W_i and b_i are the weight matrix and bias that are used to form the aspect space. According to this notation, we reframe Equation 2.6 as:

$$\mathcal{L}_{CL} = -\frac{1}{|P(q)|} \sum_{p \in P(q)} \log \frac{\exp(sim(z_q^i, z_p^i)/\tau)}{\sum_{p \in P(q)} \exp(sim(z_q^i, z_p^i)/\tau) + \sum_{n \in N(q)} \exp(sim(z_q^i, z_n^i)/\tau)} \tag{5.1}$$

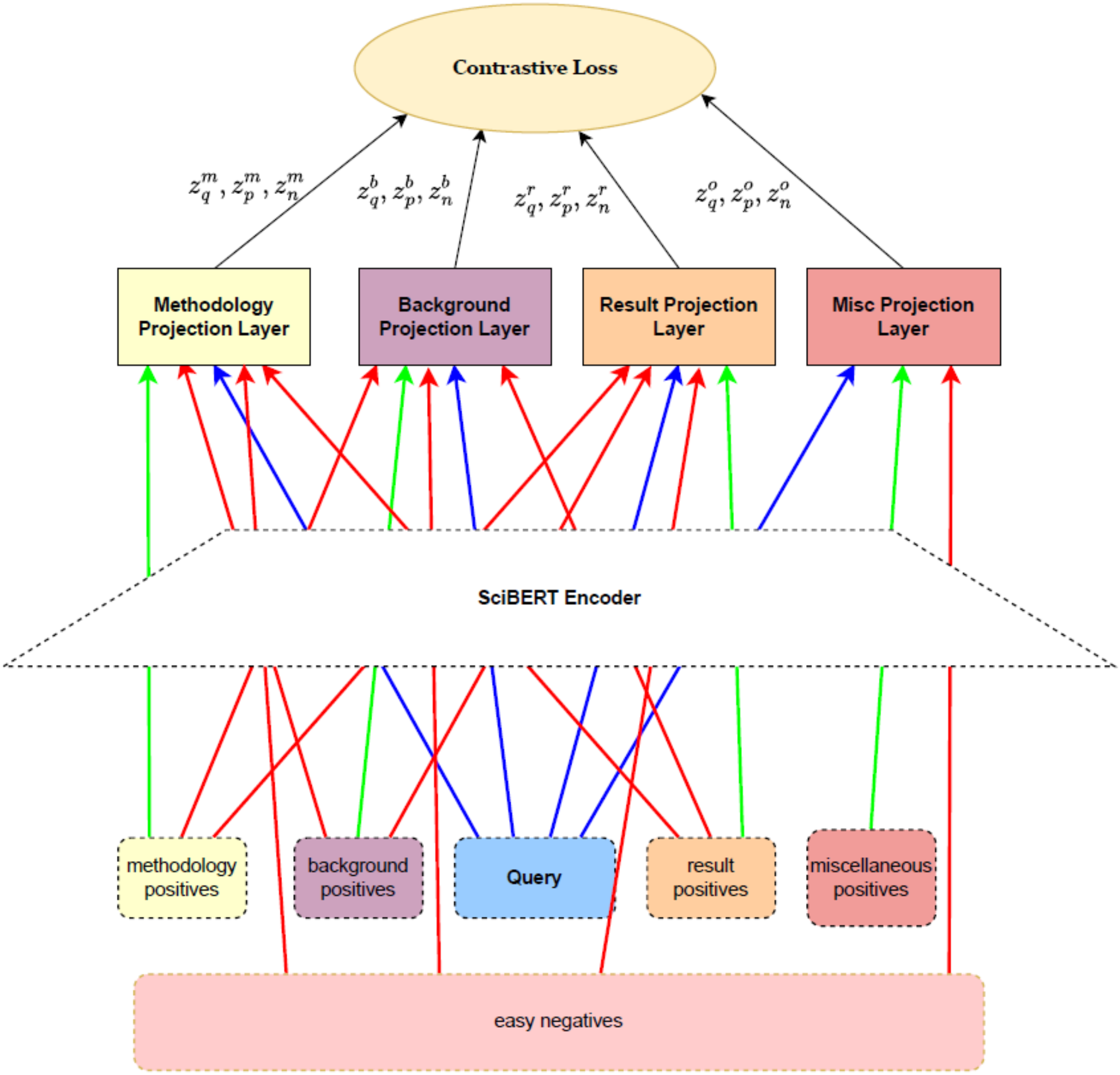


Figure 5.3.: Framework of SciAspect. Red arrows represent negatives. Green arrows represent positives.

Now we experiment with different settings to learn this model. Following Experiment 2, we keep the temperature τ fixed as 0.05.

- **Experiment 3.1 - *m* positive, *n* negatives:** This is the standard setting in which we pass all the obtained positives and negatives through the network.
- **Experiment 3.2 - 1 positive, *n* negatives:** From the previous experiments, this was found to be the optimal way to model the contrastive loss.
- **Experiment 3.3 - 1 positive, *n* negatives and no miscellaneous aspect:** Here, we study how the performance will change if we eliminate the miscellaneous aspect space as it does not have any hard negatives to learn from.
- **Experiment 3.4 - Aspect-specific embeddings:** Following [15], we also inspect the performance of the embeddings in the respective aspect-specific embedding spaces. This means instead of taking the trained BERT representation of the test papers, we also project them as done during the training time to the learnt aspect spaces and then evaluate the resulting embedding. We also try evaluating the concatenation of the embeddings generated from the three different aspect spaces. Following the results of the previous experiments, we compute the contrastive loss using 1 positive and n negatives. Also, we only consider the methodology, background and result aspects in this case. Aspect-specific Equation 5.2 and concatenated embeddings Equation 5.3 for a paper p are described below.

$$Embedding(aspect(p)) = proj\_aspect(Enc(p)) \tag{5.2}$$

$$\begin{aligned} Concat\_Embedding(p) = CONCAT(&Embedding(methodology(p)) \\ &+Embedding(background(p)) \\ &+Embedding(result(p))) \end{aligned} \tag{5.3}$$

- **Experiment 3.5 - SciAspect(Weighted):** In this experiment, we try to add more complexity to the SciAspect model architecture by adding an additional layer at the input that uses attention mechanism to weigh input embeddings for the different aspects before feeding into the model. We hypothesize that to decide the intention of a citation, getting final representations from the [CLS] tokens may not take finer sentence-level signals into account. Instead paying special attention to specific terms and sentences in an abstract might play a key role. For example, if it is a survey paper, it is likely to be cited with the intention of background information while if a paper is proposing a new method, then it will likely be cited for methodology. To capture this information more thoroughly, we pay attention to every sentence of the paper w.r.t the intention. So, instead of taking the final [CLS] embedding of a whole paper (title + abstract) at once, we average over the sentence-aware, aspect-specific embeddings of the paper.

We first take the SciBERT embeddings of each of the four aspects and pass them through a trainable linear layer, then taking these as queries, we calculate attention with respect to the sentence embeddings of a paper. This results in four different embeddings for each aspect, weighted by the sentences of the paper. These are then averaged to form the final embedding of the paper. This is done to obtain embeddings for all query, positive and negative papers which are then passed as input to be projected into different aspect spaces and contrastive learning is performed, similar to the above SciAspect experiment. The whole pipeline is illustrated in Figure 5.4

To split paper abstracts into sentences, we used an NLP library, 'scispacy' with the model 'en_core_sci_scibert' which is built specifically for processing scientific text (predominantly Biomedical text) [83].

Now to get individual representations of these sentences we were inspired by [84]. In vanilla BERT, the [CLS] is used as a symbol to aggregate features from one sentence or a pair of sentences. We modify the model by inserting multiple [CLS] symbols to get features for the sentences. We insert a [CLS] token before each sentence and a [SEP] token after each sentence.

$$\begin{aligned} z_i = SciBERT(&[CLS]title(i)[SEP]\ [CLS]abstract(i)_{Sentence1}[SEP]\ [CLS] \\ &abstract(i)_{Sentence2}[SEP]\ [CLS]\ldots abstract(i)_{Sentence_n}[SEP]) \end{aligned}$$

5.4. Experiment 4 - SciAspectHybrid

In experiment 2, the embeddings are learned such that documents connected through direct citations are pulled together and otherwise pushed apart. This is a global approach to represent documents in the embedding space. To refine this further, in experiment 3 we suggest a local approach in which we add further discrimination between the directly connected documents based on their citation intent. Now we experiment with a hybrid learning objective where the representations can learn from both the global and local similarity signal. This way we try to strike a good balance between being too narrow (the local approach) and too wide (the global approach).

To achieve this, the loss function is modelled as the weighted average of the global SPECTER-based contrastive loss and the local SciAspect-based contrastive loss, as given in the

$$\mathcal{L}_{hybrid} = \alpha * \mathcal{L}_{global} + (1 - \alpha) * \mathcal{L}_{local} \tag{5.4}$$

Here, α is an adjustable loss scaling factor to balance between the two losses. For this experiment we chose α as 0.7.

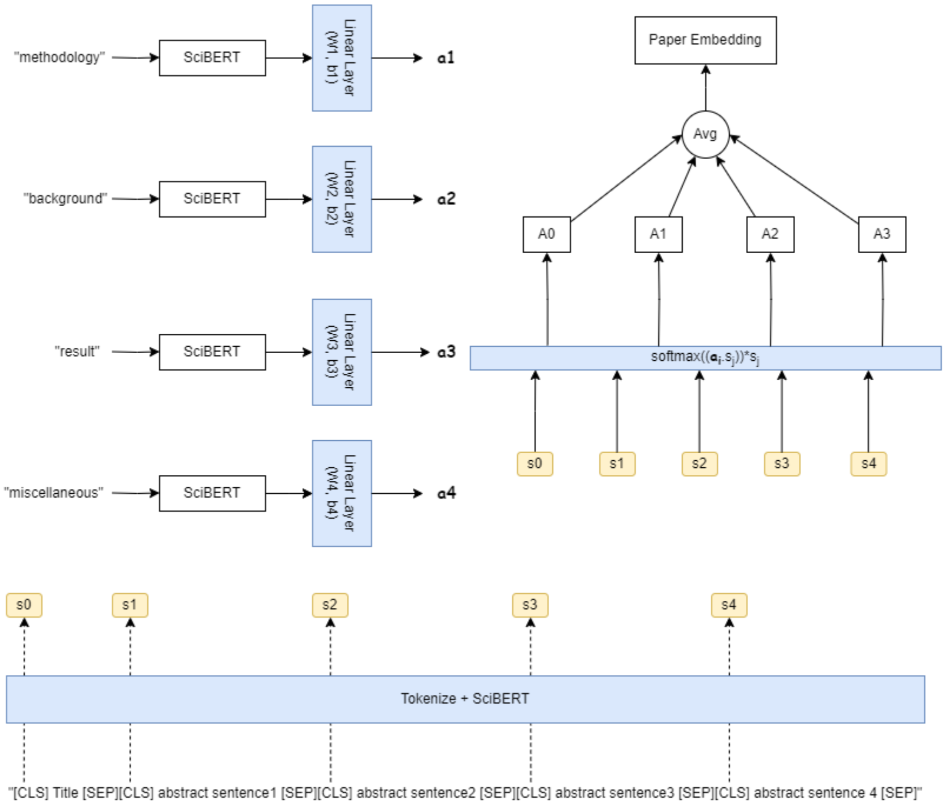


Figure 5.4.: SciAspect Weighted Architecture

Now during training, at each instance for an anchor document we pass the aspect-specific positives and negatives as well as the SPECTERCL like positives and negatives to the model, as illustrated in. As found in previous experiments the best modelling combination is to use 1 positive and n negatives. So, in a training/validation instance, for each anchor document we provide an aspect, a paper directly connected to the anchor with respect to the given aspect (positive), n papers that are directly connected with the anchor but in a different aspect (local negatives) and m papers that are indirectly connected with the anchor i.e., reference of references (global negatives). The temperature was set as 0.05. We call this model SciAspectHybrid and also try four different variations of it.

- **Experiment 4.1 - Aspect-specific embeddings:** Similar to Experiment 3.4, we analyze the hybrid of global embeddings and aspect-specific/concatenated embeddings.

- **Experiment 4.2 - Different weights(aplha):** We train and evaluate SciAspectHybrid with different alpha weights (0.5, 0.6, 0.8, 0.9) between global and local loss.

5.5. Experiment 5 - SciAspect with L2 negatives

In SciAspect, we treat directly connected papers in a different aspect as negative and try to push them away in the given aspect space. However, it may be a harsh objective which may confound the model in learning the local neighbourhood. We also lose the capability of SPECTER which encompasses knowledge of the global neighbourhood. To include that, in SciAspectHybrid, we train the model with both local negatives and global negatives. Another way to incorporate this information is to choose indirectly connected papers in a different aspect as negatives instead of directly connected papers in a different aspect as negatives Figure 5.5.

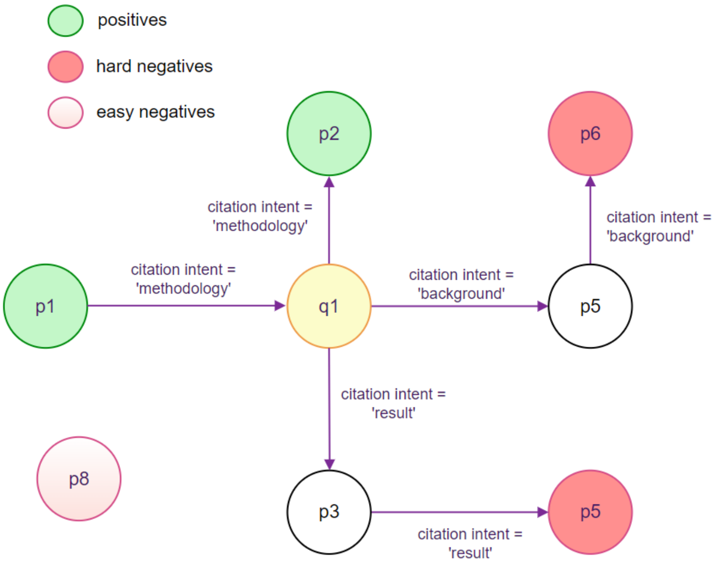

positives
hard negatives
easy negatives
p2
p6
citation intent = 'methodology'
citation intent = 'background'
p1
citation intent = 'methodology'
q1
citation intent = 'background'
p5
citation intent = 'result'
p8
p3
p5
citation intent = 'result'

Figure 5.5.: Positive and negative mining strategy used in SciAspect with L2 negatives: Example of positives and negatives for the **methodology** aspect.

We try different configurations for the same:

- **Experiment 5.1 - 1 positive, 3 easy negatives:** First, we test SciAspect with just 1 positive and 3 easy negatives. To reiterate, these are papers not connected with the anchor paper directly or through reference-of-references (level 2/L2).
- **Experiment 5.2 - 1 positive, 5 hard negatives:** Then we train SciAspect with 5 hard negatives, where hard negatives are papers not connected with the anchor paper through reference-of-references (level 2/L2) in the aspect that the positive is selected from.
- **Experiment 5.3 - 1 positive, 3 hard negatives and 3 easy negatives:** Finally, we pass 3 easy and 3 hard negatives for each positive into this variant of SciAspect.

Model	#positives, #negatives	Positive Candidates	Negative Candidates
SPECTER(Undirected)	1, 1	direct references	reference-of-reference
SPECTERCL	1, 5	direct references and citations	reference-of-reference
SciAspect	1, n	direct references and citations in the same aspect	reference in different aspect
SciAspectHybrid	1, 10 (5 local, 5 global)	direct references and citations in the same aspect	- For global loss: reference-of-reference - For local loss: reference in different aspect
SciAspectL2	1, 6	direct references and citations in the same aspect	reference-of-reference in different aspect

Table 5.1.: Proposed Models' Summary

The number of positives and negatives, and the positive and negative selection strategy for the best of the proposed models are summarised in Table 5.1.

6. Results

6.1. SciDocs

6.1.1. Baseline

Table 6.1 shows the results reported in [10] by evaluating SciDocs for different models including general contextual models [26, 85–87]; BERT; variants of BERT pretrained on scientific data [6]; fine-tuned variants of SciBERT [74, 88]; SPECTER and SciNCL (when trained by [10] on data free from leakage).

They also estimate Oracle SciDocs scores that represent an upper bound of the performance that can be achieved with a triplet margin loss and SciBERT encoder. It was obtained by training the same model as SPECTER/SciNCL except that its triples were generated from the labels of the validation and test set of SciDocs. For example, papers with the same MAG labels are positives and papers with different labels are negatives. Similarly, the ground truth for the other tasks is used, i.e., clicked recommendations are considered as positives etc. In total, this procedure creates 106K training triples for Oracle SciDocs.

Task→	Classification		User activity prediction				Citation prediction				Recomm.		Avg
Subtask→	MAG	MeSH	Co-View		Co-Read		Cite		Co-Cite				
Model/Metric↓	F1	F1	MAP	nDCG	MAP	nDCG	MAP	nDCG	MAP	nDCG	nDCG	P@1	
Oracle SciDocs	87.1	94.8	87.2	93.5	88.7	94.6	92.3	96.8	91.4	96.4	53.8	19.4	83.0
Doc2Vec	66.2	69.2	67.8	82.9	64.9	81.6	65.3	82.2	67.1	83.4	51.7	16.9	66.6
fastText-sum	78.1	84.1	76.5	87.9	75.3	87.4	74.6	88.1	77.8	89.6	52.5	18.0	74.1
ELMo	77.0	75.7	70.3	84.3	67.4	82.6	65.8	82.6	68.5	83.8	52.5	18.2	69.0
Citeomatic	67.1	75.7	81.1	90.2	80.5	90.2	86.3	94.1	84.4	92.8	52.5	17.3	76.0
SGC	76.8	82.7	77.2	88.0	75.7	87.5	91.6	96.2	84.1	92.5	52.7	18.2	76.9
BERT	79.9	74.3	59.9	78.3	57.1	76.4	54.3	75.1	57.9	77.3	52.1	18.1	63.4
SciBERT	79.7	80.7	50.7	73.1	47.7	71.1	48.3	71.7	49.7	72.6	52.1	17.9	59.6
BioBERT	77.2	73.0	53.3	74.0	50.6	72.2	45.5	69.0	49.4	71.8	52.0	17.9	58.8
CiteBERT	78.8	74.8	53.2	73.6	49.9	71.3	45.0	67.9	50.3	72.1	51.6	17.0	58.8
Random S2ORC	in data (w/o leakage):												
SPECTER	81.3	88.4	83.1	91.3	84.0	92.1	86.2	93.9	87.8	94.7	52.2	17.5	79.4
SciNCL	81.3	89.4	84.3	91.8	85.6	92.8	91.4	96.3	90.1	95.7	54.3	19.9	81.1

Table 6.1.: Values reported in [10] by evaluating SciDocs for different models.

6.1.2. Main Results

Table 6.2 presents the main results corresponding to the SciDocs benchmark. SPECTERCL, the model similar to SPECTER in the design of positives and negatives with the added condition of considering in-citations as positives, and trained with a contrastive loss objective with multiple negative examples, outperformed the existing state-of-the-art SciNCL in 8 out of 12 metrics. The replicated version of SPECTER trained on our dataset performed the best in MeSH and Co-View tasks. The newly proposed SciAspect, SciAspectHybrid, SciAspectL2 outperformed existing benchmarks in classification tasks, with SciAspectHybrid being the best in MAG classification. SPECTERCL outperformed existing benchmarks in the recommendation tasks. For citation prediction, SciNCL still performed the best. A detailed discussion of these results is provided in the next chapter.

Task→	Classification		User activity prediction				Citation prediction				Recomm.		Avg
Subtask→	MAG	MeSH	Co-View		Co-Read		Cite		Co-Cite				
Model/Metric↓	F1	F1	MAP	nDCG	MAP	nDCG	MAP	nDCG	MAP	nDCG	nDCG	P@1	
SciBERT*	79.43	79.92	59.81	78.1	55.71	75.33	53.17	73.76	57.67	77.33	51.72	17.59	63.295
SPECTER*	81.3	88.4	83.1	91.3	84.0	92.1	86.2	93.9	87.8	94.7	52.2	17.5	79.4
SciNCL*	81.3	89.4	84.3	91.8	85.6	92.8	**91.4**	**96.3**	90.1	95.7	54.3	19.9	81.1
SPECTER(Undirected)	81.78	**89.84**	**84.66**	**92.01**	85.81	93.05	90.45	95.97	89.88	95.6	52.34	17.02	80.70
SPECTERCL (1p, 5n)	81.99	89.25	84.53	91.94	**86.14**	**93.2**	89.76	95.59	**90.26**	**95.82**	**54.66**	**20.56**	**81.14**
SciAspect	82.65	89.0	83.69	91.5	84.2	92.19	87.68	94.59	88.68	95.11	52.36	17.65	79.94
SciAspectHybrid	**82.72**	89.62	84.33	91.9	84.6	92.32	86.85	94.16	89.34	95.44	53.07	18.85	80.27
SciAspectL2	82.75	88.36	84.16	91.76	84.95	92.61	85.62	93.55	89.15	95.36	53.74	18.89	80.07

* The baseline scores taken from [10]

Table 6.2.: Results on SciDocs found by the best models in the experiments along with the baselines for comparison

6.1.3. Experiment Results

In this section, we present the results from the different experiments evaluated on SciDocs.

Experiment 2: SPECTER with contrastive loss

Table 6.3 summarizes the results obtained by the different variants for the model in experiment 2. We found that the contrastive learning model learned by passing 1 positive and 5 negatives performed the best. Removing hard negatives improved performance in classification while reduced performance in recommendation. Using just 1 positive and 1 negative as in a triplet, harmed performance. Finally, keeping the number of positives and negatives fixed at 1 and 5 and gradually increasing the temperature first increased performance and then decreased it, with a peak observed at the value 0.05.

Experiment 3: SciAspect

Table 6.4 summarizes the results obtained by the different variants for the model in experiment 3. Consistent with above observations, the SciAspect model trained with 1 positive and n negatives was found to be better than training with m positives. But like above, the model with more positives performed better than the model with 1 positive in classification tasks and worse in recommendation tasks. Moreover, removing the miscellaneous aspect, which does not add much valuable information because of the lack of hard negatives was found to perform better. The concatenated representations of the embeddings obtained by projecting to the aspect spaces were found to be overall worse than the embeddings obtained from just

Task→	Classification		User activity prediction				Citation prediction				Recomm.		Avg
Subtask→	MAG	MeSH	Co-View		Co-Read		Cite		Co-Cite				
Model/Metric↓	F1	F1	MAP	nDCG	MAP	nDCG	MAP	nDCG	MAP	nDCG	nDCG	P@1	
Different number of positives and negatives (t = 0.05)													
1 positive, 1 negative	81.81	88.19	83.88	91.66	84.37	92.18	88.74	95.03	88.94	95.16	54.33	19.95	80.35
1 positive, 5 negatives	81.99	89.25	**84.53**	**91.94**	**86.14**	**93.2**	**89.76**	**95.59**	**90.26**	**95.82**	**54.66**	**20.56**	**81.14**
1 positive, 3 easy negatives	82.68	**89.65**	84.27	91.86	84.87	92.5	88.03	94.75	89.54	95.44	53.94	19.22	80.56
2 positive, 5 negatives	82.51	88.62	83.52	91.48	83.7	91.9	88.15	94.82	88.39	94.85	52.6	17.7	79.85
4 positives, 5 negatives	82.38	88.37	83.65	91.54	83.55	91.82	88.31	94.86	88.2	94.77	52.13	16.93	79.71
5 positives, 5 negatives	**83.02**	89.32	83.45	91.49	83.61	91.86	87.96	94.68	88.11	94.73	52.27	17.61	79.84
Different temperatures (1 positive, 5 negatives)													
$\tau = 0.01$	81.84	88.86	84.1	91.69	84.82	92.52	87.38	94.41	89.15	95.33	54.03	19.57	80.31
$\tau = 0.05$	81.99	**89.25**	**84.53**	**91.94**	**86.14**	**93.2**	**89.76**	95.59	**90.26**	**95.82**	**54.66**	**20.56**	**81.14**
$\tau = 0.1$	**82.87**	89.02	84.43	91.92	85.03	92.68	89.7	**95.62**	89.33	95.44	51.93	17.45	80.45
$\tau = 0.5$	81.85	88.36	79.53	89.31	77.31	88.17	80.18	90.17	81.84	91.35	52.74	17.68	76.54

Table 6.3.: SciDocs results on SPECTERCL variants (Experiment 2)

the encoder model. Finally, contrary to expectations, SciAspectWeighted performed poorly than SciAspect.

Task→	Classification		User activity prediction				Citation prediction				Recomm.		Avg
Subtask→	MAG	MeSH	Co-View		Co-Read		Cite		Co-Cite				
Model/Metric↓	F1	F1	MAP	nDCG	MAP	nDCG	MAP	nDCG	MAP	nDCG	nDCG	P@1	
1 positive, n negatives (w/ misc aspect)	81.98	89.26	83.08	91.25	83.29	91.66	85.96	93.58	87.15	94.31	**52.9**	**18.64**	79.42
m positives, n negatives	**83.07**	**89.36**	82.35	90.8	82.4	91.24	84.5	92.91	86.2	93.79	52.38	17.81	78.9
1 positive, n negatives (w/o misc aspect)	82.65	89.0	83.69	91.5	**84.2**	**92.19**	**87.68**	**94.59**	88.68	**95.11**	52.36	17.65	**79.94**
Aspect-specific embeddings for SciAspect(1 positive, n negatives)(w/o misc aspect)													
SciAspect (methodology)	82.39	88.39	83.49	91.42	83.84	91.95	87.14	94.37	88.38	94.96	51.93	17.22	79.62
SciAspect (background)	82.48	88.59	83.65	91.47	84.04	92.11	87.45	94.45	88.63	95.1	52.48	17.45	79.86
SciAspect (result)	82.65	88.72	83.64	91.5	84.1	92.15	87.31	94.43	88.5	94.96	52.79	18.22	79.91
1 positive, n negatives (concat)	82.39	88.51	**83.7**	**91.54**	84.07	92.1	87.43	94.46	**88.69**	95.1	52.31	17.59	79.82
SciAspectWeighted (1 positive, n negatives)(w/o misc aspect)													
SciAspectWeighted	81.51	87.53	82.62	90.95	82.89	91.43	85.08	93.16	86.81	94.1	52.44	17.95	78.87

Table 6.4.: SciDocs results on SciAspect variants($\tau = 0.05$) (Experiment 3)

Experiment 4: SciAspectHybrid

Table 6.5 and Table 6.6 summarize the results obtained by the different variants for the model in experiment 4. Evaluating the SciAspectHybrid model in different aspect spaces, we found that the embeddings from the result space performed better than the embeddings from the background space, followed by the methodology space. A concatenated embedding of all three aspect spaces performed better than individual aspect spaces. However, they were still slightly worse than SciAspectHybrid itself.

When choosing for alpha, the base model with alpha = 0.7 was the best. We also observe that, increasing alpha from 0.5, decreased the performance on MAG classification while increased the performance in other tasks. The reason behind that is because SciAspect performs better in classification tasks, while SPECTERCL performs better in recommendation.

Experiment 5

SciDocs results for experiment 5 are reported in Table 6.7. We observe that SciAspect with 1 positive and 3 easy negatives performed better than SciAspect(all variants). While the overall

Task→	Classification		User activity prediction				Citation prediction				Recomm.		Avg
Subtask→	MAG	MeSH	Co-View		Co-Read		Cite		Co-Cite				
Model/Metric↓	F1	F1	MAP	nDCG	MAP	nDCG	MAP	nDCG	MAP	nDCG	nDCG	P@1	
base model	**82.72**	**89.62**	84.33	91.9	**84.6**	**92.32**	**86.85**	**94.16**	**89.34**	**95.44**	**53.07**	**18.85**	**80.27**
local loss w.r.t. methodology space	81.6	89.14	84.19	91.84	84.34	92.17	86.55	94.07	89.05	95.27	52.19	17.3	79.81
local loss w.r.t. background space	82.12	88.95	84.18	91.77	84.49	92.3	86.52	93.93	89.29	95.38	52.66	17.88	79.96
local loss w.r.t. result space	81.95	89.01	84.35	**91.96**	84.33	92.2	86.57	94.05	89.02	95.26	52.91	18.56	80.01
local loss w.r.t. concatenated embeddings	82.56	89.4	**84.4**	91.93	84.55	92.31	86.66	94.05	89.27	95.41	52.71	18.07	80.11

Table 6.5.: SciDocs results on SciAspectHybrid architecture variants($\tau = 0.05$, number of positives=1, number of negatives=m) (Experiment 4.1)

Task→	Classification		User activity prediction				Citation prediction				Recomm.		Avg
Subtask→	MAG	MeSH	Co-View		Co-Read		Cite		Co-Cite				
Model/Metric↓	F1	F1	MAP	nDCG	MAP	nDCG	MAP	nDCG	MAP	nDCG	nDCG	P@1	
alpha = 0.5	**82.91**	89.38	84.21	91.8	84.35	92.2	86.74	94.07	89.04	95.27	52.76	18.34	80.09
alpha = 0.6	82.76	89.54	84.28	91.83	84.48	92.25	**87.13**	**94.27**	89.23	95.38	52.79	18.14	80.17
alpha = 0.7 (base)	82.72	**89.62**	84.33	91.9	84.6	2.32	86.85	94.16	89.34	95.44	53.07	18.85	**80.27**
alpha = 0.8	82.65	89.51	84.37	91.93	84.72	92.41	86.74	94.09	89.36	95.44	52.95	18.4	80.21
alpha = 0.9	82.43	89.47	**84.44**	**91.97**	**84.83**	**92.48**	86.52	93.99	**89.39**	**95.45**	**53.2**	**18.86**	80.25

Table 6.6.: SciDocs results on SciAspectHybrid variants of parameter alpha($\tau = 0.05$, number of positives=1, number of negatives=m) (Experiment 4.2)

performance of SciAspect with 1 positive and 3 easy negatives was the best, the performance of SciAspectL2 with 1 positive, 3 hard negatives and 3 easy negatives was the better in 7 out of 12 metrics.

Task→	Classification		User activity prediction				Citation prediction				Recomm.		Avg
Subtask→	MAG	MeSH	Co-View		Co-Read		Cite		Co-Cite				
Model/Metric↓	F1	F1	MAP	nDCG	MAP	nDCG	MAP	nDCG	MAP	nDCG	nDCG	P@1	
SciAspect 1 positive 3 easy negatives	82.64	**89.05**	83.95	91.66	83.93	91.97	**86.77**	**94.09**	88.65	94.98	**53.9**	19.33	**80.08**
SciAspect L2 negatives (5 hard negatives)	82.22	87.42	82.43	90.78	83.54	91.97	80.58	90.93	86.88	94.26	**53.9**	**19.76**	78.72
SciAspect L2 negatives (3 hard negatives, 3 easy negatives)	**82.75**	88.36	**84.16**	**91.76**	**84.95**	**92.61**	85.62	93.55	**89.15**	**95.36**	53.74	18.89	80.07

Table 6.7.: SciDocs results on SciAspect Experiment 5 models with L2 negatives

Models	BM25		SCIBERT		SPECTER		SciNCL		ASPIRE-BM		ASPIRE-CS	
Fields	MAP	R @ 5	MAP	R @ 5	MAP	R @ 5	MAP	R @ 5	MAP	R @ 5	MAP	R @ 5
Art	**38.2**	**32.3**	22.4	16.6	34.1	28.8	34.7	29.2	34.0	27.7	34.1	28.0
Bio	38.3	33.6	20.4	14.0	34.6	30.0	36.8	32.3	**38.7**	**33.7**	35.7	29.9
Bus	28.1	22.5	19.1	13.1	27.5	21.8	28.5	**24.6**	28.5	23.4	**29.6**	23.1
Ch	**38.0**	**32.6**	20.0	13.7	33.7	29.3	36.5	31.5	36.5	31.0	34.1	28.3
CS	34.8	30.5	19.5	12.7	35.6	30.4	**37.2**	**32.5**	35.4	30.4	35.4	30.1
Eco	**30.5**	**26.0**	21.4	15.4	27.3	21.9	28.3	23.2	29.3	24.3	28.0	22.7
Eng	**34.6**	**29.3**	20.5	13.9	31.3	27.3	34.2	28.0	32.7	27.7	33.4	28.1
ES	**31.6**	**26.2**	21.3	15.1	30.1	24.2	31.5	25.5	30.8	24.7	29.9	23.7
Geog	**31.8**	**27.8**	21.9	16.7	26.4	22.2	29.5	23.8	30.3	26.0	28.4	22.2
Geol	**33.1**	**28.0**	19.5	13.9	24.8	20.1	25.7	19.9	28.5	23.5	25.8	21.4
His	**38.1**	**32.9**	20.8	15.2	27.1	20.6	30.9	23.9	31.0	24.2	28.5	22.1
MS	**36.1**	**30.7**	22.1	15.5	34.1	28.2	35.8	29.6	35.8	29.8	34.0	29.2
Mat	35.3	28.3	22.8	18.3	34.2	28.9	34.9	30.1	36.2	31.0	**36.9**	**32.2**
Med	38.6	32.5	22.0	16.4	41.4	36.3	42.7	36.5	**44.0**	**37.8**	41.7	36.7
Phi	**30.2**	**25.7**	19.2	13.3	27.1	21.1	29.9	23.5	28.7	24.1	29.1	23.3
Phy	**35.1**	**30.2**	23.9	18.1	30.8	26.3	34.5	30.0	32.9	27.7	32.9	28.7
PS	**28.6**	**23.1**	19.4	14.0	24.2	18.0	26.4	21.7	25.9	21.2	26.8	21.7
Psy	32.5	28.9	20.3	16.2	32.3	28.1	34.2	**30.5**	**34.3**	29.4	34.2	28.3
Soc	26.8	20.5	20.2	15.8	25.2	20.5	26.7	21.9	**27.3**	**22.2**	26.7	**22.2**
Avg	**33.7**	**28.5**	20.9	15.2	30.6	25.5	32.6	27.3	32.7	27.4	31.8	26.4

Table 6.8.: Results in terms of MAP and R@5 on MDCR. Values in bold indicate the best performing model for a combination of field and metric

6.2. MDCR

6.2.1. Baselines

Table 6.8 recalls the previous model performances on MDCR reported in [19]. We note that in the paper, the nDCG values for the baseline models are not reported. Also, the precise results on the given test set are only provided for the SciNCL model Table 6.9. These values were reconfirmed by evaluating the benchmark on SciNCL's HuggingFace model.

Field	MAP	nDCG	recall@5
Art	34.7136	60.3739	29.15
Biology	36.8358	62.7903	32.3
Business	28.4669	55.5329	24.6
Chemistry	36.5293	61.951	31.5
Computer Science	37.2056	62.7272	32.2
Economics	28.3171	55.395	23.2
Engineering	34.1759	60.4889	28.0
Environmental Science	31.4845	58.1616	25.5
Geography	29.4518	56.4011	23.8
Geology	25.72	52.8014	19.9
History	30.9442	57.1201	23.85
Materials Science	35.7855	61.9001	29.6
Mathematics	34.9152	60.9257	30.1
Medicine	42.6579	67.0481	36.5
Philosophy	29.8966	56.9015	23.45
Physics	34.499	60.3763	30.3
Political Science	26.3834	53.9156	21.7333
Psychology	34.1724	60.8867	30.5
Sociology	26.6838	54.0052	21.9
AVG	32.5704	58.9317	27.2675

Table 6.9.: Results obtained from evaluating the test set on SciNCL reported in MDCR's GitHub repository for the SciNCL model [1]

6.2.2. Experiment Results

Table 6.10 presents the results generated on evaluating the proposed models on MDCR. We take only SPECTERCL, SciAspect and SciAspectHybrid into consideration. Since, this is a purely recommendation task, we also evaluate it on SciAspect(concat) because we hypothesize that this model can do document comparison and recommendation in the aspect latent spaces. As stated above we can only compare our results on the MAP and Recall@5 metrics. And the true comparison on the same test set for all three metrics can only be done with the SciNCL model. We observe that our version of the SPECTER model trained via Contrastive Learning was able to outperform the best model BM25 w.r.t the MAP metric. It was also able to perform better than the next best SciNCL model in the all the other metrics. Coming to SciAspect, SciAspectHybrid and SciAspectHybrid(concat), we observe that although not better, the results for all metrics were comparable to SciNCL and were much better than SPECTER.

[1] https://github.com/zoranmedic/mdcr

Models	SciAspect			SciAspectHybrid			SciAspectHybrid(concat)			SPECTERCL		
Fields	MAP	nDCG	recall@5	MAP	nDCG	recall@5	MAP	nDCG	recall@5	MAP	nDCG	recall@5
Art	35.8884	61.8609	29.675	35.3471	61.0267	29.225	35.1077	60.7779	29.25	**39.5219**	**64.6546**	**33.175**
Biology	37.8107	**63.8141**	32.7	35.6078	61.7919	29.8	35.0275	61.3811	29.5	**38.3833**	63.7726	**33.6**
Business	29.6102	**56.5349**	**24.9**	**29.6716**	56.5343	24.3	29.1424	56.0929	23.7	29.614	56.3188	24.4
Chemistry	36.0408	61.9243	30.4	35.0916	61.1486	30.0	34.5446	60.8715	28.8	**37.4207**	**62.7871**	**31.9**
Computer Science	33.5129	59.9752	28.6	33.6166	60.0257	29.6	33.5718	59.9054	28.6	**35.0327**	**60.8233**	**29.8**
Economics	29.0597	55.992	23.9	29.9081	**57.0336**	24.5	29.312	56.3065	23.6	**29.9341**	56.8492	**24.6**
Engineering	31.7213	58.2002	26.4	31.9758	58.2666	**27.1**	32.2496	58.5496	26.3	**32.7904**	**59.269**	26.7
Environmental Science	30.5862	57.4912	24.0	30.4452	57.2448	24.9	30.1632	57.0597	24.1	**31.8268**	**58.2125**	**25.9**
Geography	29.6594	56.8324	23.8	29.0374	56.307	25.0	28.8802	56.1977	24.6	**31.676**	**58.5847**	**25.8**
Geology	25.7205	53.1501	21.4	26.0201	53.1658	21.2	25.8747	53.0521	21.8	**27.1988**	**54.4726**	**22.0**
History	32.0197	57.8836	25.1	32.134	58.0601	26.025	31.9018	57.8281	25.2	**33.6084**	**59.2065**	**26.95**
Materials Science	34.5561	60.6282	30.2	33.6648	60.1766	28.6	33.683	60.1373	29.6	**36.5694**	**62.4875**	**32.4**
Mathematics	35.2147	61.2538	29.2	34.8087	60.8715	28.4	34.8053	60.8985	29.3	**36.6159**	**62.4083**	**31.0**
Medicine	42.9074	67.3184	37.8	41.8509	66.1769	36.7	42.2568	66.7107	36.1	**44.1619**	**68.1375**	**38.3**
Philosophy	30.4059	57.4571	25.35	30.1626	57.1867	24.65	30.1662	57.275	24.65	**31.6667**	**58.4593**	**25.55**
Physics	**33.3889**	59.3969	**28.9**	33.2899	59.5166	28.7	33.2929	**59.5703**	28.4	32.6382	58.9497	27.7
Political Science	27.3483	54.2999	21.8333	27.646	54.6142	21.8667	27.2982	54.2845	21.6667	**29.193**	**56.1439**	**23.0333**
Psychology	34.2729	60.9107	29.7	**34.7621**	61.4245	**30.8**	34.3902	61.0973	30.1	34.6926	**61.5367**	30.1
Sociology	28.7196	55.9999	23.9	28.5986	55.969	22.5	28.5846	56.0021	22.6	**29.1627**	**56.3987**	**24.4**
AVG	32.5497	58.996	27.2504	32.2968	58.7653	27.0456	32.1186	58.6315	26.7298	**33.7741**	**59.9722**	**28.2794**

Table 6.10.: MDCR results on our proposed models

6.3. SciRepEval

Table 6.11 shows the results of the baseline model performances (taken from [89]) on SciRepEval benchmark along with the performance of our models. The best overall model performance and out-of-task performance was observed for SciAspectHybrid. SciAspectL2 was observed to be the best performer for In-task evaluation benchmarks followed closely by SciAspect and SciAspectHybrid. For most other regression tasks like review score, max h-index, and tweet mentions prediction and classification tasks, aspect-aware embeddings also performed better than aspect-unaware embeddings. In ranking tasks (PRX, QRY), the aspect-aware embeddings performed worse than aspect-unaware embeddings. With SciNCL still being the best in tasks like recommendation when searched with text query or query document/s and peer reviewer matching. SPECTERCL was found to be much worse than all other methods concerning regression tasks like review score, citation count, and publishing year prediction. Ultimately, all citation-informed embeddings were comparable in their overall performance.

Type	Task		Metric	SciBERT*	SPECTER (w/ leakage)*	SciNCL (w/ leakage)*	SPECTER(Undirected)	SPECTERCL	SciAspectHybrid	SciAspect	SciAspect(L2)
Out-of-Train	**Biomimicry [CLF]**	Complete	F1	73.37	72.87	69.74	72.27	73.05	74.52	72.29	**74.53**
		Few shot - 64 samples 50 runs	F1	37.26	39.62	40.14	42.11	42.06	**45.35**	45.13	44.04
		Few shot - 16 samples 100 runs	F1	16.00	19.50	21.26	22.12	23.02	25.62	**31.14**	19.85
			Wt. F1	50.00	51.22	50.22	52.19	52.79	55.00	**55.21**	53.24
	DRSM [CLF]	Complete	F1	76.84	**77.34**	74.73	74.71	75.27	75.37	76.35	76.11
		Few shot - 64 samples 50 runs	F1	56.31	61.06	61.24	62.04	63.88	64.02	**65.43**	63.50
		Few shot - 24 samples 100 runs	F1	46.05	48.88	49.68	52.61	54.09	55.43	**56.39**	53.78
			Wt. F1	64.01	66.16	65.10	66.02	67.13	67.55	**68.63**	67.37
	Feeds	Paper Query [PRX]	MAP	68.17	81.11	**81.16**	80.15	79.96	78.44	78.54	79.31
		Multi paper query [PRX]	MAP	65.44	74.35	**75.30**	73.44	73.87	72.83	72.52	72.85
		Title query [QRY]	MAP	66.42	**81.23**	80.72	78.65	79.80	76.52	76.96	76.39
	TREC CoVID [QRY]		nDCG	79.73	86.53	87.67	**89.00**	88.67	87.68	88.07	87.83
	Peer Reviewer Matching [PRX]	Hard decision	P@5	26.92	33.27	**34.21**	32.34	30.84	31.59	31.40	31.21
			P@10	24.30	25.51	25.42	25.42	25.05	**25.61**	25.23	25.05
		Soft decision	P@5	60.93	65.79	66.54	**66.92**	66.36	66.73	66.36	65.61
			P@10	54.58	**56.17**	55.42	55.98	55.61	**56.17**	55.79	55.89
			Avg	41.68	45.19	**45.40**	45.17	44.47	45.03	44.70	44.44
	Review Score [RGN]	ICLR 17-22	K Tau	20.26	17.35	18.87	20.60	14.07	**22.08**	21.57	21.40
	Max h-Index [RGN]		K Tau	6.81	10.04	11.30	13.97	11.74	12.82	13.67	**14.12**
	Tweet Mentions [RGN]		K Tau	22.18	24.19	25.78	21.31	21.37	**26.19**	20.50	24.21
In-Train	**MeSH [CLF]**		F1	76.71	85.46	86.17	**87.29**	86.09	85.91	85.80	85.41
	Same Author Prediction [PRX]		MAP	79.48	86.53	87.47	87.39	**87.91**	87.25	87.51	87.40
	Search [QRY/PRX]		nDCG	71.46	73.31	**73.54**	73.31	73.18	72.92	72.54	73.04
	Citation Context [PRX]		MAP	33.72	42.89	43.39	44.05	43.11	43.29	43.48	**43.63**
	Citation Count [RGN]		K Tau	**39.16**	33.21	34.61	36.16	23.70	36.99	36.98	37.10
	Publishing Year [RGN]		K Tau	27.71	25.96	29.00	28.40	15.88	30.01	**30.41**	30.34
SciDocs	**MAG [CLF]**		F1	79.54	79.40	81.11	81.55	81.97	82.50	82.09	**82.62**
	MeSH [CLF]		F1	79.84	87.70	89.00	**90.11**	89.41	89.67	89.08	88.45
	Co-View [PRX]		MAP	59.80	83.40	85.28	**84.63**	84.51	84.33	83.69	84.15
			nDCG	78.10	91.40	92.23	**91.99**	91.93	91.91	91.50	91.76
	Co-Read [PRX]		MAP	55.73	85.10	87.69	85.79	**86.15**	84.59	84.21	84.95
			nDCG	75.34	92.70	94.00	93.05	**93.20**	92.31	92.20	92.61
	Cite [PRX]		MAP	53.20	92.00	93.55	**90.36**	89.69	86.76	87.58	85.51
			nDCG	73.79	96.60	97.35	**95.93**	95.56	94.12	94.55	93.49
	Co-cite [PRX]		MAP	57.71	88.00	91.66	89.84	**90.27**	89.34	88.64	89.14
			nDCG	77.36	94.70	96.44	95.59	**95.83**	95.44	95.08	95.35
			CLF Avg	70.02	73.99	74.32	75.43	75.48	76.13	**76.16**	75.42
			REG Avg	23.22	22.15	23.91	24.09	17.35	**25.62**	24.63	25.43
		(Excluding SciDocs)	**PRX Avg**	59.99	67.23	**67.71**	67.25	67.08	66.63	66.55	66.78
			QRY Avg	72.54	80.36	**80.64**	80.32	80.55	79.04	79.19	79.09
			Out of task Avg	49.88	53.74	54.15	54.05	53.39	**54.41**	54.04	54.12
			In task Avg	54.71	57.89	59.03	59.43	54.98	59.40	59.45	**59.49**
			Scidocs Avg	69.04	89.10	90.83	89.88	**89.85**	89.10	88.86	88.80
			All avg	58.05	67.76	**68.82**	68.52	67.25	68.37	68.14	68.16
			Avg without SciDocs	51.59	55.20	55.87	55.95	53.95	**56.17**	55.95	56.01

Table 6.11.: SciRepEval Results

7. Discussion

This chapter will discuss the results and the potential sources of error.

7.1. Overview

With these experiments we learned that useful scientific document embeddings, in some cases even better than the state-of-the-art SciNCL, can be learned with just considering the close neighbourhood of the documents instead of the whole citation graph considered in SciNCL, thus saving on both memory and computational overhead incurred by generating citation embeddings for positive and negative mining.

When comparing with SPECTER, the model that we intended to improve, that only considers the close neighbourhood of the documents, we found that our replication of the SPECTER model, modified with a few changes suggested by SciNCL, performed significantly better than SPECTER itself in many tasks. In fact, all our models overall outperformed SPECTER. Next, we observe that all our proposed models also outperformed the reported baselines in the classification tasks. This indicates that considering both citations and references as positive improves performance.

Our SPECTER model trained via Contrastive Learning, was found to be the best performing model in the SciDocs and MDCR baselines. It was able to outperform the best model performances of SciNCL and BM25 on the recommendation tasks in terms of both SciDocs and MDCR. In SciRepEval, however, SPECTERCL had a comparably worse performance than other models, especially in regression tasks where the performance was significantly lower than all other models. Inspection of the reason for this and to understand more about the latent space is left for future work.

In classification tasks, our newly suggested models generally performed better than SPECTER because they were made to be more closer together with their immediate neighbours (both citations and references) as compared to SPECTER. Similarly, they performed better than SciNCL because the latent space in SciNCL also brings together relatively distant citation relations. Moreover, aspect-aware models outperformed the other aspect-unaware models on the task of classifying MAG Fields in case of SciDocs, and Biomimicry and DRSM classification tasks in SciRepEval. This can be reasoned by the fact that they are trained such that the documents that cite each other with the same intention often belong to the same field of topic.

In SciDocs, among the aspect-based models, in general, SciAspectHybrid performed better than SciAspectL2 which was better than SciAspect but worse than SPECTERCL. In SciRepEval, on the other hand, the best overall model performance and out-of-task performance were observed for SciAspectHybrid. This suggests that integrating both the global knowledge of citation distance as in SPECTER and the local knowledge of citation intent as in SciAspect helped learn good generalizable embeddings for scientific documents. SciAspectL2 was observed to be the best performer for In-task evaluation benchmarks in SciRepEval followed closely by SciAspect and SciAspectHybrid. This is because they performed much better in the task of publishing year prediction. The reason behind that could be that the papers directly connected through citations with the same intent are much more likely to cite recent methodology/background/result. For most other regression tasks like review score, max h-index, and tweet mentions prediction, aspect-aware embeddings also performed better than aspect-unaware embeddings.

Overall apart from classification tasks where SciAspect was better than SciAspectHybrid and SciAspectL2, we learned that adding directly connected papers with a different aspect as negative, harmed learning generalized embeddings. So, awareness of both local and global structures is important for the tasks.

In ranking tasks in SciDocs (user activity, citation prediction, and recommendation) and SciRepEval (PRX, QRY), the aspect-aware embeddings performed worse than aspect-unaware embeddings. With SciNCL still being the best in tasks like recommendation when searched with text query or query document/s and peer reviewer matching. This is because of its broader understanding of the distant citation connections. This validates the benefits of using SciNCL's choice of positives and negatives for learning. With regards to MDCR as well, our models are overall marginally worse than SciNCL but are still better in some domains like Art, Sociology, and Business. In some domains, the aspect-based models are even better than the state-of-the-art BM25.

7.2. Sources of Error

One of the given sources of errors in all the models considered is the limitations of the base SciBERT model itself. There would still be words/phrases in the training documents that are not present in the vocabulary of SciBERT and are not utilized for the semantic representations especially because the training set for SciBERT was skewed for certain domains (82% biomedical and 18% computer science). Moreover, only the title and abstract were used for training which are mere teasers of the whole document, and incorporating more information about the contents would help in better understanding and distinguishing the documents.

Given that, the main objective of injecting aspect-awareness into the SPECTER model was to improve its classification and recommendation capabilities. While we were able to achieve this in some sub tasks, our models did not show the desired improvement in recommendation, user activity prediction or citation prediction tasks. We try to analyze the potential sources of errors for the results.

First, we came to the conclusion that the evaluation datasets and models were typically small in scale and scope. While the evaluation tasks are appropriate to evaluate the general performance of the embeddings, they are not ideal for evaluating the true essence of the citation-aspect-awareness of the embeddings. For instance, in tasks where a query document is provided and the task is to recommend relevant documents to cite or predict whether two documents will be viewed or read together or have a citation link between one another, the information about citation intent does not necessarily add value, as for these tasks possibly all citation intents are required. It is the same for search with a textual query unless the intent is implicitly or explicitly mentioned in the query.

Irrespective of that, the additional aspect-related information should not have decreased performance on these tasks. One of the reasons for this is how the aspects are chosen. We considered citation intent as the aspect term to align embeddings in the respective aspect spaces. However, this may not be the ideal choice for configuring the vector space since these intents are still interconnected and more disconnected aspects could have improved the variance in performance. Moreover, as described, the information about the citation intent came from a classification model which was not completely accurate and their biases also propagated into our models. For instance, on inspecting the citation intents provided by Semantic Scholar for the SPECTER paper [11], we found that even though SciNCL and ASPIRE use SPECTER for background, methodology and most importantly result comparison, their citation intents were given as 'methodology' and 'background'. This is a potential source of error that deterred the desired learning of our models. In the future, better citation intent classification algorithms could improve the performance of our models.

8. Conclusion

Scientific document representation learning is a challenging but very rewarding problem. Finding a good solution for this problem has the potential to directly improve the quality of scientific production, as it impacts many downstream applications such as recommendation. In this work we provide a new perspective for learning scientific document representations. We challenge the existing approaches on both their choice of loss function and their strategy for defining positives and negatives when learning the representations through contrastive learning. We were able to improve the performance of the SPECTER model by posing its triplet loss function as a general contrastive or hybrid loss function and also by proposing newer methods for sampling positives and negatives for the loss. We found that integrating aspect-specific information into the general structural and semantic information can potentially improve model performance, especially in our case for classification problems. Our proposed models were comparable and even better than the state-of-the-art models in some scenarios when evaluated on the given benchmarks. Thus, validating the merit of our hypothesis. Overall, we were able to learn good generalizable embeddings that are comparable in performance with the existing approaches and can be productively consumed for downstream applications with minimal fine-tuning.

8.1. Future Work

The performance of our models inspires further study into aspect-based representation learning of scientific documents. When learning aspect-based positives and negatives, we only used the direct neighborhood of the anchor, and the positives and negatives were chosen if they were connected in the same aspect or not. Instead, we could get aspect-specific neighborhoods of the anchor document using a graph embedding method as proposed in the state-of-the-art SciNCL model. So, instead of one homogeneous citation graph we could look at different citation graphs in different aspect spaces to find out if we can further improve the representation provided by the proposed models. Graph Attention Transformers, which compute node embeddings by aggregating information from their neighbourhood, could also be explored to model this data.

Moreover, many more architectural choices can be explored. Instead of a linear projection for representing the aspect spaces a more complex neural network can be used. Additional contextual and metadata information can also be added instead of just title and abstract to gain more insights about the document. Careful hyperparameter tuning should also be performed. More avenues for representing positives and negatives could also be explored such as where author, venue, topic etc. are aspects. Citation intents itself are not limited to the three used in this work and can be extended further for e.g. whether the paper is citing to support or contrast a prior work or whether the authors cite to indicate how their work can be used later. Also, as we stated that the underlying citation intent classification model used to get the aspects is sub-optimal, we can also experiment with the newly introduced methods like [90] and [91] that showed better performance on the given benchmarks.

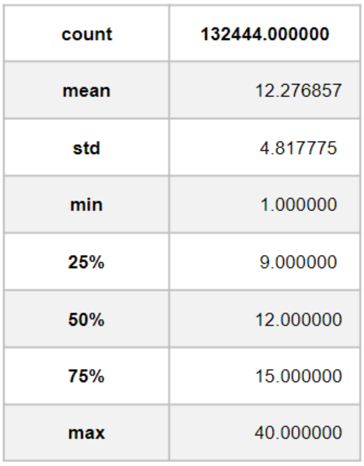
The evaluation scope of this work was limited to only a few tasks. A more comprehensive study of both the qualitative and quantitative aspects of the latent spaces learned by the models is required. In the future, we would also like to perform an intensive study to test our model on a more diverse range of downstream applications. For e.g., with the aspect-aware embeddings that ideally encode the citation information in the latent space, we would like to test how they would perform on finer-grained intent-based or context-based recommendation systems like [92] and [93] in which a citation context and its intent are considered for citation prediction or recommendation.

Hierarchical contrastive learning can also be explored to achieve even finer-grained representations by distinguishing the papers connected even by the same aspect. In hierarchical contrastive learning, a positive pair is constructed by pairing the anchor with data drawn from all levels in a hierarchy. The learning objective is to force positive pairs closer together, but the magnitude of the force is dependent on the common ancestry of the pair's labels [94]. To find fine-grained embeddings for the papers, one idea is to first use an appropriate topic model to extract important topics from possibly the whole text of the documents (these topics will be the same for different aspects), similarity in these aspects would act as the first level for determining positives and then use other aspects such as citation intents in the subsequent level to determine positives.

SciRepEval [20] suggested that instead of learning a general embedding for all tasks, for each of the classification, regression, ad-hoc and proximity task formats, multiple embeddings should be learned to offer best performance. Similarly, [95] suggested multi-objective learning of scientific document embeddings for the task of query-document (ad-hoc) and document-document (proximity) based information retrieval. We can also use these objectives to fine-tune and evaluate our models.

A. Appendix

A.1. Metadata Information about the dataset



count	132444.000000
mean	12.276857
std	4.817775
min	1.000000
25%	9.000000
50%	12.000000
75%	15.000000
max	40.000000

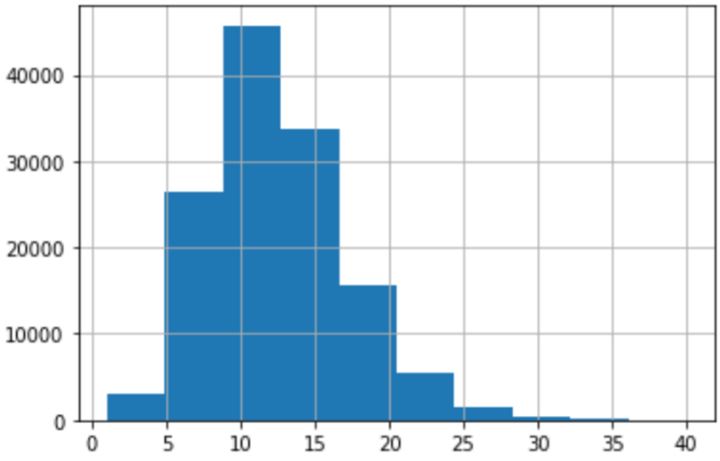
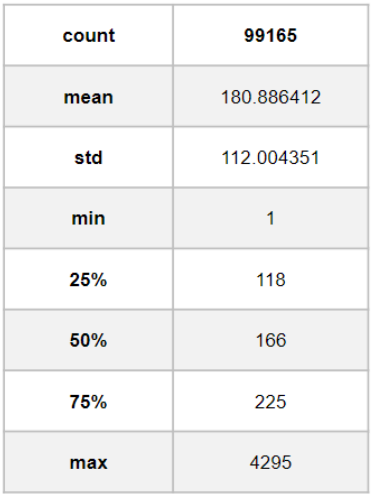


Figure A.1.: Title word count statistics of the query documents



count	99165
mean	180.886412
std	112.004351
min	1
25%	118
50%	166
75%	225
max	4295

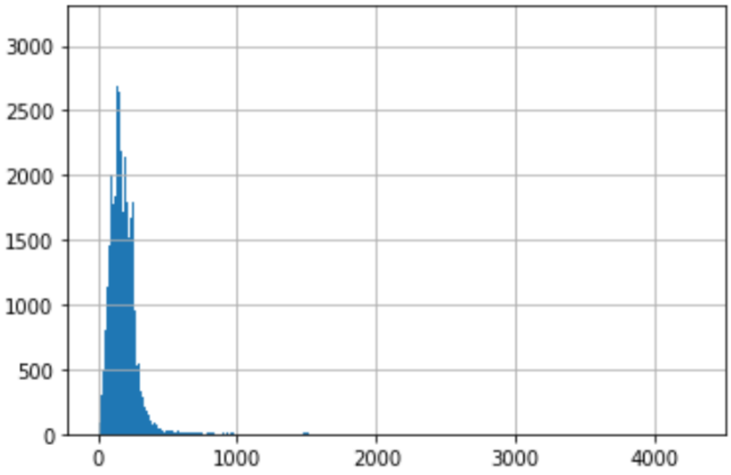
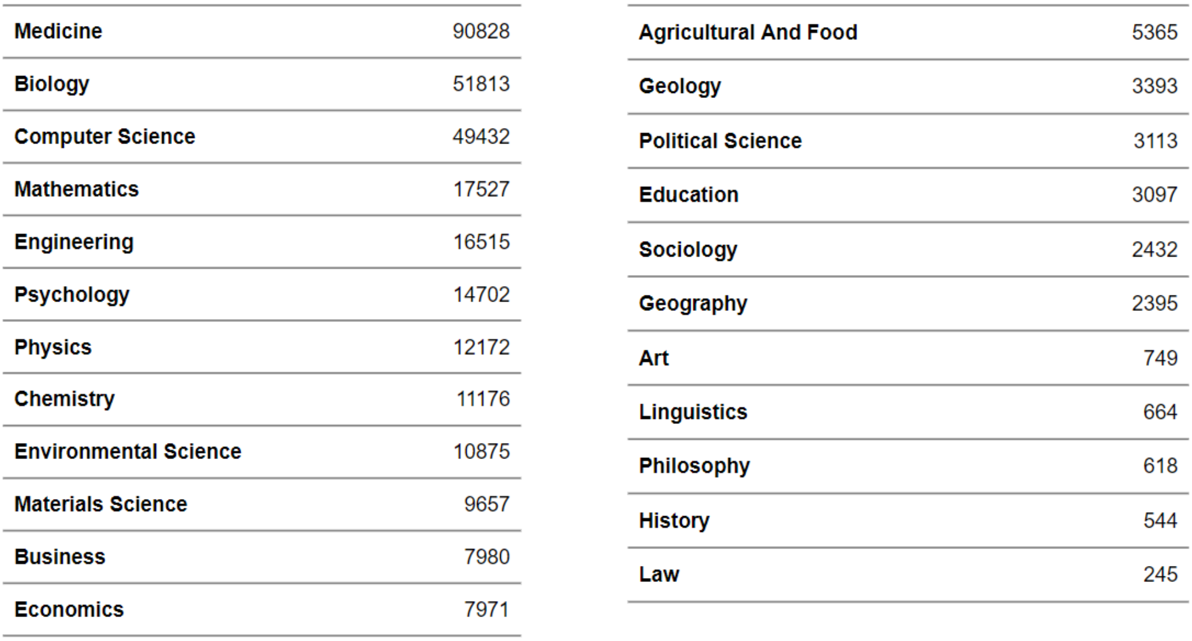
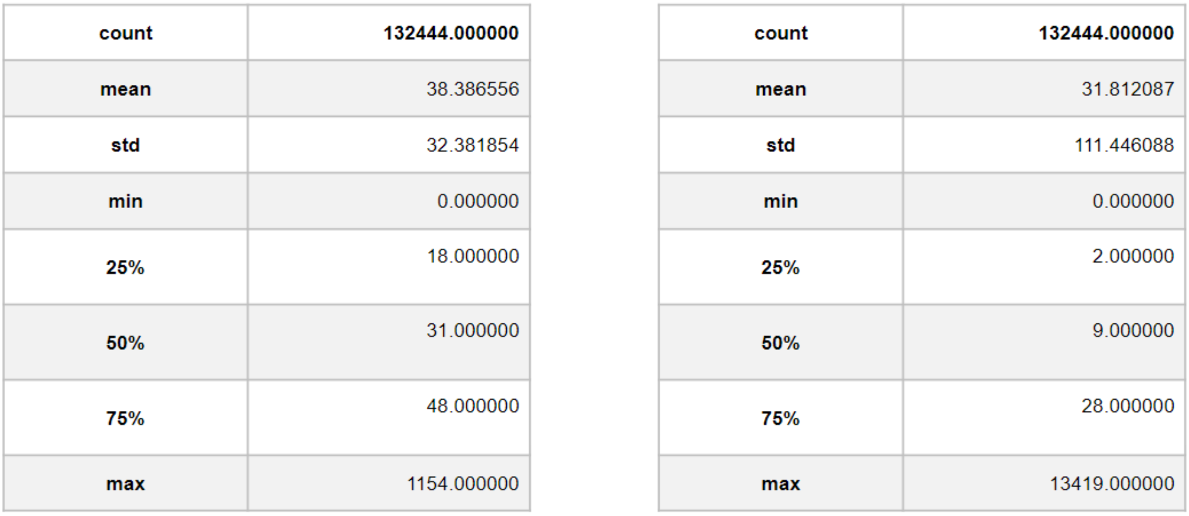


Figure A.2.: Abstract word count statistics of the query documents



Medicine	90828
Biology	51813
Computer Science	49432
Mathematics	17527
Engineering	16515
Psychology	14702
Physics	12172
Chemistry	11176
Environmental Science	10875
Materials Science	9657
Business	7980
Economics	7971
Agricultural And Food	5365
Geology	3393
Political Science	3113
Education	3097
Sociology	2432
Geography	2395
Art	749
Linguistics	664
Philosophy	618
History	544
Law	245

Figure A.3.: The distribution of the Fields of Study of the query documents



count	**132444.000000**
mean	38.386556
std	32.381854
min	0.000000
25%	18.000000
50%	31.000000
75%	48.000000
max	1154.000000

(a) Statistics about the number of references of the query documents

count	**132444.000000**
mean	31.812087
std	111.446088
min	0.000000
25%	2.000000
50%	9.000000
75%	28.000000
max	13419.000000

(b) Statistics about the number of citations to the query documents

Figure A.4.: Reference and citation count statistics

List of Figures

List of Tables

Bibliography

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. *Attention Is All You Need*. Dec. 5, 2017. DOI: `10.48550/arXiv.1706.03762`. arXiv: `1706.03762[cs]`.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. NAACL-HLT 2019. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: `10.18653/v1/N19-1423`.

[3] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut. *ALBERT: A Lite BERT for Self-supervised Learning of Language Representations*. Feb. 8, 2020. arXiv: `1909.11942[cs]`.

[4] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. July 26, 2019. DOI: `10.48550/arXiv.1907.11692`. arXiv: `1907.11692[cs]`.

[5] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. Feb. 29, 2020. DOI: `10.48550/arXiv.1910.01108`. arXiv: `1910.01108[cs]`.

[6] I. Beltagy, K. Lo, and A. Cohan. *SciBERT: A Pretrained Language Model for Scientific Text*. Sept. 10, 2019. DOI: `10.48550/arXiv.1903.10676`. arXiv: `1903.10676[cs]`.

[7] I. Beltagy, M. E. Peters, and A. Cohan. *Longformer: The Long-Document Transformer*. Dec. 2, 2020. DOI: `10.48550/arXiv.2004.05150`. arXiv: `2004.05150[cs]`.

[8] M. Zaheer, G. Guruganesh, A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, and A. Ahmed. *Big Bird: Transformers for Longer Sequences*. Jan. 8, 2021. DOI: `10.48550/arXiv.2007.14062`. arXiv: `2007.14062[cs,stat]`.

[9] S. Dehghan and M. F. Amasyali. "SupMPN: Supervised Multiple Positives and Negatives Contrastive Learning Model for Semantic Textual Similarity". In: *Applied Sciences* 12.19 (Jan. 2022). Number: 19 Publisher: Multidisciplinary Digital Publishing Institute, p. 9659. ISSN: 2076-3417. DOI: `10.3390/app12199659`.

[10] M. Ostendorff, N. Rethmeier, I. Augenstein, B. Gipp, and G. Rehm. *Neighborhood Contrastive Learning for Scientific Document Representations with Citation Embeddings*. arXiv:2202.06671. type: article. arXiv, Feb. 14, 2022. arXiv: `2202.06671[cs]`.

[11] A. Cohan, S. Feldman, I. Beltagy, D. Downey, and D. S. Weld. *SPECTER: Document-level Representation Learning using Citation-informed Transformers*. arXiv:2004.07180. type: article. arXiv, May 20, 2020. arXiv: 2004.07180[cs].

[12] S. Mysore, A. Cohan, and T. Hope. *Multi-Vector Models with Textual Guidance for Fine-Grained Scientific Document Similarity*. May 4, 2022. arXiv: 2111.08366[cs].

[13] S. Tan, S. Zhao, and Y. Zhang. "Coherence-Based Distributed Document Representation Learning for Scientific Documents". In: *ArXiv* (2022).

[14] M. Ostendorff, T. Ruas, T. Blume, B. Gipp, and G. Rehm. *Aspect-based Document Similarity for Research Papers*. arXiv:2010.06395. type: article. arXiv, Oct. 13, 2020. arXiv: 2010.06395[cs].

[15] M. Ostendorff, T. Blume, T. Ruas, B. Gipp, and G. Rehm. *Specialized Document Embeddings for Aspect-based Similarity of Research Papers*. arXiv:2203.14541. type: article. arXiv, Mar. 28, 2022. arXiv: 2203.14541[cs].

[16] S. Mammola, D. Fontaneto, A. Martínez, and F. Chichorro. "Impact of the reference list features on the number of citations". In: *Scientometrics* 126.1 (Jan. 1, 2021), pp. 785–799. ISSN: 1588-2861. DOI: 10.1007/s11192-020-03759-0.

[17] K. Henner. "Enriching Scientific Paper Embeddings with Citation Context". Accepted: 2020-02-04T19:28:20Z. Thesis. 2019.

[18] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan. *Supervised Contrastive Learning*. arXiv:2004.11362. type: article. arXiv, Mar. 10, 2021. arXiv: 2004.11362[cs,stat].

[19] Z. Medic and J. Šnajder. "Large-scale Evaluation of Transformer-based Article Encoders on the Task of Citation Recommendation". In: *undefined* (2022). DOI: 10.48550/arXiv.2209.05452.

[20] A. Singh, M. D'Arcy, A. Cohan, D. Downey, and S. Feldman. "SciRepEval: A Multi-Format Benchmark for Scientific Document Representations". In: (2022). Publisher: arXiv Version Number: 1. DOI: 10.48550/ARXIV.2211.13308.

[21] S. Wang and C. Manning. "Baselines and Bigrams: Simple, Good Sentiment and Topic Classification". In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. ACL 2012. Jeju Island, Korea: Association for Computational Linguistics, July 2012, pp. 90–94.

[22] S. Robertson and S. Walker. "Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval". In: Proceedings of the 17th ACM Conference on Research and Development in Information Retrieval (SIGIR), Dublin, Ireland. Jan. 1, 1994, pp. 232–241. ISBN: 978-3-540-19889-5. DOI: 10.1007/978-1-4471-2099-5_24.

[23] T. Mikolov, K. Chen, G. Corrado, and J. Dean. *Efficient Estimation of Word Representations in Vector Space*. Sept. 6, 2013. arXiv: 1301.3781[cs].

[24] J. Pennington, R. Socher, and C. Manning. "GloVe: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. EMNLP 2014. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. DOI: `10.3115/v1/D14-1162`.

[25] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. *Bag of Tricks for Efficient Text Classification*. Aug. 9, 2016. DOI: `10.48550/arXiv.1607.01759`. arXiv: `1607.01759[cs]`.

[26] G. J, M. Gupta, and V. Varma. "Doc2Sent2Vec: A Novel Two-Phase Approach for Learning Document Representation". In: *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. SIGIR '16. New York, NY, USA: Association for Computing Machinery, July 7, 2016, pp. 809–812. ISBN: 978-1-4503-4069-4. DOI: `10.1145/2911451.2914717`.

[27] A. Radford and K. Narasimhan. "Improving Language Understanding by Generative Pre-Training". In: 2018.

[28] B. Perozzi, R. Al-Rfou, and S. Skiena. "DeepWalk". In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, Aug. 2014. DOI: `10.1145/2623330.2623732`.

[29] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. "LINE". In: *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, May 2015. DOI: `10.1145/2736277.2741093`.

[30] A. Grover and J. Leskovec. *node2vec: Scalable Feature Learning for Networks*. July 3, 2016. DOI: `10.48550/arXiv.1607.00653`. arXiv: `1607.00653[cs,stat]`.

[31] T. N. Kipf and M. Welling. *Semi-Supervised Classification with Graph Convolutional Networks*. Feb. 22, 2017. arXiv: `1609.02907[cs,stat]`.

[32] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. *Graph Attention Networks*. 2018. arXiv: `1710.10903 [stat.ML]`.

[33] D. J. J. H. Martin. *Speech and Language Processing*. 2023. URL: `https://web.stanford.edu/~jurafsky/slp3/10.pdf`.

[34] T. Gao, X. Yao, and D. Chen. "SimCSE: Simple Contrastive Learning of Sentence Embeddings". In: (Apr. 18, 2021). DOI: `10.48550/arXiv.2104.08821`.

[35] F. Schroff, D. Kalenichenko, and J. Philbin. "FaceNet: A Unified Embedding for Face Recognition and Clustering". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015, pp. 815–823. DOI: `10.1109/CVPR.2015.7298682`. arXiv: `1503.03832[cs]`.

[36] K. Sohn. "Improved Deep Metric Learning with Multi-class N-pair Loss Objective". In: *Advances in Neural Information Processing Systems*. Vol. 29. Curran Associates, Inc., 2016.

[37] W. Ammar, D. Groeneveld, C. Bhagavatula, I. Beltagy, M. Crawford, D. Downey, J. Dunkelberger, A. Elgohary, S. Feldman, V. Ha, R. Kinney, S. Kohlmeier, K. Lo, T. Murray, H.-H. Ooi, M. Peters, J. Power, S. Skjonsberg, L. L. Wang, C. Wilhelm, Z. Yuan, M. van Zuylen, and O. Etzioni. "Construction of the Literature Graph in Semantic Scholar". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*. NAACL-HLT 2018. New Orleans - Louisiana: Association for Computational Linguistics, June 2018, pp. 84–91. DOI: 10.18653/v1/N18-3011.

[38] A. Cohan, W. Ammar, M. van Zuylen, and F. Cady. *Structural Scaffolds for Citation Intent Classification in Scientific Publications*. Sept. 30, 2019. arXiv: 1904.01608[cs].

[39] C. E. Lipscomb. "Medical Subject Headings (MeSH)." In: *Bulletin of the Medical Library Association* 88 3 (2000), pp. 265–6.

[40] A. Sinha, Z. Shen, Y. Song, H. Ma, D. Eide, B. P. Hsu, and K. Wang. "An Overview of Microsoft Academic Service (MAS) and Applications". In: *WWW (Companion Volume)*. ACM, 2015, pp. 243–246.

[41] K. Santhanam, O. Khattab, J. Saad-Falcon, C. Potts, and M. Zaharia. "ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction". In: *CoRR* abs/2112.01488 (2021).

[42] E. Voorhees, T. Alam, S. Bedrick, D. Demner-Fushman, W. R. Hersh, K. Lo, K. Roberts, I. Soboroff, and L. L. Wang. "TREC-COVID: constructing a pandemic information retrieval test collection". In: *ACM SIGIR Forum* 54.1 (June 2020), pp. 1–12. ISSN: 0163-5840. DOI: 10.1145/3451964.3451965.

[43] V. Shyam, L. Friend, B. Whiteaker, N. Bense, J. Dowdall, B. Boktor, M. Johny, I. Reyes, A. Naser, N. Sakhamuri, et al. "PeTaL (periodic table of life) and physiomimetics". In: *Designs* 3.3 (2019), p. 43.

[44] *Disease Research State Model (DSRM)*. https://github.com/chanzuckerberg/DRSM-corpus.

[45] Z. Medic and J. Snajder. "A Survey of Citation Recommendation Tasks and Methods". In: *CIT. Journal of Computing and Information Technology* 28.3 (July 12, 2021). Number: 3, pp. 183–205. ISSN: 1846-3908.

[46] F. Sebastiani. "Machine Learning in Automated Text Categorization". In: *ACM Computing Surveys* 34.1 (Mar. 2002), pp. 1–47. ISSN: 0360-0300, 1557-7341. DOI: 10.1145/505282.505283. arXiv: cs/0110053.

[47] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. "The Author-Topic Model for Authors and Documents". In: (2004).

[48] C. Wang and D. M. Blei. "Collaborative topic modeling for recommending scientific articles". In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. KDD '11. New York, NY, USA: Association for Computing Machinery, Aug. 21, 2011, pp. 448–456. ISBN: 978-1-4503-0813-7. DOI: 10.1145/2020408.2020480.

[49] S. Wang and R. Koopman. “Semantic embedding for information retrieval”. In: (2017).

[50] O. Gökçe, J. Prada, N. I. Nikolov, N. Gu, and R. H. Hahnloser. “Embedding-based Scientific Literature Discovery in a Text Editor Application”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Online: Association for Computational Linguistics, July 2020, pp. 320–326. DOI: 10.18653/v1/2020.acl-demos.36.

[51] H. J. Meijer, J. Truong, and R. Karimi. *Document Embedding for Scientific Articles: Efficacy of Word Embeddings vs TFIDF*. July 11, 2021. DOI: 10.48550/arXiv.2107.05151. arXiv: 2107.05151[cs].

[52] Y. Wang, S. Liu, N. Afzal, M. Rastegar-Mojarad, L. Wang, F. Shen, P. Kingsbury, and H. Liu. “A comparison of word embeddings for the biomedical natural language processing”. In: *Journal of Biomedical Informatics* 87 (Nov. 1, 2018), pp. 12–20. ISSN: 1532-0464. DOI: 10.1016/j.jbi.2018.09.008.

[53] Y. Zhang, Q. Chen, Z. Yang, H. Lin, and Z. Lu. “BioWordVec, improving biomedical word embeddings with subword information and MeSH”. In: *Scientific Data* 6.1 (May 10, 2019). Number: 1 Publisher: Nature Publishing Group, p. 52. ISSN: 2052-4463. DOI: 10.1038/s41597-019-0055-0.

[54] V. Tshitoyan, J. Dagdelen, L. Weston, A. Dunn, Z. Rong, O. Kononova, K. A. Persson, G. Ceder, and A. Jain. “Unsupervised word embeddings capture latent knowledge from materials science literature”. In: *Nature* 571.7763 (July 2019). Number: 7763 Publisher: Nature Publishing Group, pp. 95–98. ISSN: 1476-4687. DOI: 10.1038/s41586-019-1335-8.

[55] F. K. Khattak, S. Jeblee, C. Pou-Prom, M. Abdalla, C. Meaney, and F. Rudzicz. “A survey of word embeddings for clinical text”. In: *Journal of Biomedical Informatics*. Articles initially published in Journal of Biomedical Informatics: X 1-4, 2019 100 (Jan. 1, 2019), p. 100057. ISSN: 1532-0464. DOI: 10.1016/j.yjbinx.2019.100057.

[56] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang. “BioBERT: a pre-trained biomedical language representation model for biomedical text mining”. In: *Bioinformatics* 36.4 (Feb. 15, 2020), pp. 1234–1240. ISSN: 1367-4803, 1367-4811. DOI: 10.1093/bioinformatics/btz682. arXiv: 1901.08746[cs].

[57] Y. Gu, R. Tinn, H. Cheng, M. R. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, and H. Poon. “Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing”. In: *ACM Trans. Comput. Heal.* (2022). DOI: 10.1145/3458754.

[58] E. Alsentzer, J. R. Murphy, W. Boag, W.-H. Weng, D. Jin, T. Naumann, and M. B. A. McDermott. *Publicly Available Clinical BERT Embeddings*. June 20, 2019. DOI: 10.48550/arXiv.1904.03323. arXiv: 1904.03323[cs].

[59] T. Gupta, M. Zaki, N. M. A. Krishnan, and Mausam. *MatSciBERT: A Materials Domain Language Model for Text Mining and Information Extraction*. Sept. 30, 2021. DOI: 10.48550/arXiv.2109.15290. arXiv: 2109.15290[cond-mat].

[60] X. Liu, D. Yin, X. Zhang, K. Su, K. Wu, H. Yang, and J. Tang. *OAG-BERT: Pre-train Heterogeneous Entity-augmented Academic Language Models*. version: 2. Mar. 23, 2021. arXiv: `2103.02410[cs]`.

[61] K. Lo, L. L. Wang, M. Neumann, R. Kinney, and D. S. Weld. *S2ORC: The Semantic Scholar Open Research Corpus*. July 6, 2020. arXiv: `1911.02782[cs]`.

[62] R. Taylor, M. Kardas, G. Cucurull, T. Scialom, A. Hartshorn, E. Saravia, A. Poulton, V. Kerkez, and R. Stojnic. *Galactica: A Large Language Model for Science*. version: 1. Nov. 16, 2022. arXiv: `2211.09085[cs,stat]`.

[63] H. Jia and E. Saule. *Graph Embedding for Citation Recommendation*. Dec. 6, 2018. arXiv: `1812.03835[cs]`.

[64] Y. Qiao, L. Sun, J. Han, and C. Xiao. *Heterogeneous Academic Network Embedding Based Multivariate Random-Walk Model for Predicting Scientific Impact*. preprint. In Review, Aug. 24, 2020. DOI: `10.21203/rs.3.rs-56634/v1`.

[65] C. Pornprasit, X. Liu, N. Kertkeidkachorn, K.-S. Kim, T. Noraset, and S. Tuarob. "ConvCN: A CNN-Based Citation Network Embedding Algorithm towards Citation Recommendation". In: *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*. JCDL '20. New York, NY, USA: Association for Computing Machinery, Aug. 1, 2020, pp. 433–436. ISBN: 978-1-4503-7585-6. DOI: `10.1145/3383583.3398609`.

[66] *ConvCN: A CNN-Based Citation Network Embedding Algorithm towards Citation Recommendation | Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*. URL: `https://dl.acm.org/doi/abs/10.1145/3383583.3398609` (visited on 02/14/2023).

[67] C. Pornprasit, X. Liu, P. Kiattipadungkul, N. Kertkeidkachorn, K.-S. Kim, T. Noraset, S.-U. Hassan, and S. Tuarob. "Enhancing citation recommendation using citation network embedding". In: *Scientometrics* 127.1 (Jan. 1, 2022), pp. 233–264. ISSN: 1588-2861. DOI: `10.1007/s11192-021-04196-3`.

[68] E. Palumbo, G. Rizzo, R. Troncy, E. Baralis, M. Osella, and E. Ferro. "Knowledge Graph Embeddings with node2vec for Item Recommendation". In: *The Semantic Web: ESWC 2018 Satellite Events*. Ed. by A. Gangemi, A. L. Gentile, A. G. Nuzzolese, S. Rudolph, M. Maleshkova, H. Paulheim, J. Z. Pan, and M. Alam. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2018, pp. 117–120. ISBN: 978-3-319-98192-5. DOI: `10.1007/978-3-319-98192-5_22`.

[69] C. Yang, Z. Liu, D. Zhao, M. Sun, and E. Y. Chang. "Network Representation Learning with Rich Text Information". In: ().

[70] S. Ganguly and V. Pudi. "Paper2vec: Combining Graph and Text Information for Scientific Paper Representation". In: *Advances in Information Retrieval*. Ed. by J. M. Jose, C. Hauff, I. S. Altıngovde, D. Song, D. Albakour, S. Watt, and J. Tait. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2017, pp. 383–395. ISBN: 978-3-319-56608-5. DOI: `10.1007/978-3-319-56608-5_30`.

[71] X. Kong, M. Mao, W. Wang, J. Liu, and B. Xu. "VOPRec: Vector Representation Learning of Papers with Text Information and Structural Identity for Recommendation". In: *IEEE Transactions on Emerging Topics in Computing* 9.1 (Jan. 2021). Conference Name: IEEE Transactions on Emerging Topics in Computing, pp. 226–237. ISSN: 2168-6750. DOI: `10.1109/TETC.2018.2830698`.

[72] Y. Zhang, F. Zhao, and J. Lu. "P2V: large-scale academic paper embedding". In: *Scientometrics* 121.1 (Oct. 1, 2019), pp. 399–432. ISSN: 1588-2861. DOI: `10.1007/s11192-019-03206-9`.

[73] S. Wang, J. Tang, C. Aggarwal, and H. Liu. "Linked Document Embedding for Classification". In: *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. CIKM '16. Indianapolis, Indiana, USA: Association for Computing Machinery, 2016, pp. 115–124. ISBN: 9781450340731. DOI: `10.1145/2983323.2983755`.

[74] C. Bhagavatula, S. Feldman, R. Power, and W. Ammar. "Content-Based Citation Recommendation". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. NAACL-HLT 2018. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 238–251. DOI: `10.18653/v1/N18-1022`.

[75] S. Arora, H. Khandeparkar, M. Khodak, O. Plevrakis, and N. Saunshi. "A Theoretical Analysis of Contrastive Unsupervised Representation Learning". In: *CoRR* abs/1902.09229 (2019).

[76] N. Raman, S. Shah, and M. Veloso. *Structure and Semantics Preserving Document Representations*. arXiv:2201.03720. type: article. arXiv, Apr. 1, 2022. arXiv: `2201.03720[cs]`.

[77] J. Chan, J. C. Chang, T. Hope, D. Shahaf, and A. Kittur. "SOLVENT: A Mixed Initiative System for Finding Analogies between Research Papers". In: *Proc. ACM Hum.-Comput. Interact.* 2.CSCW (Nov. 2018). DOI: `10.1145/3274300`.

[78] L. Huang, J. Zhu, Y. Chi, and H. Xu. "Automatic Semantic Annotation for Abstracts of Scientific Discourses". In: (), p. 7.

[79] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. Rush. "Transformers: State-of-the-Art Natural Language Processing". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. DOI: `10.18653/v1/2020.emnlp-demos.6`.

[80] I. Loshchilov and F. Hutter. *Decoupled Weight Decay Regularization*. Jan. 4, 2019. arXiv: `1711.05101[cs,math]`.

[81] D. P. Kingma and J. Ba. *Adam: A Method for Stochastic Optimization*. Jan. 29, 2017. arXiv: `1412.6980[cs]`.

[82] C. Van Gysel and M. de Rijke. "Pytrec_eval: An Extremely Fast Python Interface to trec_eval". In: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. June 27, 2018, pp. 873–876. DOI: `10.1145/3209978.3210065`. arXiv: `1805.01597[cs]`.

[83] M. Neumann, D. King, I. Beltagy, and W. Ammar. "ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing". In: *Proceedings of the 18th BioNLP Workshop and Shared Task*. Proceedings of the 18th BioNLP Workshop and Shared Task. Florence, Italy: Association for Computational Linguistics, 2019, pp. 319–327. DOI: `10.18653/v1/W19-5034`.

[84] Y. Liu. *Fine-tune BERT for Extractive Summarization*. Sept. 5, 2019. arXiv: `1903.10318[cs]`.

[85] Q. V. Le and T. Mikolov. *Distributed Representations of Sentences and Documents*. May 22, 2014. DOI: `10.48550/arXiv.1405.4053`. arXiv: `1405.4053[cs]`.

[86] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. *Enriching Word Vectors with Subword Information*. June 19, 2017. DOI: `10.48550/arXiv.1607.04606`. arXiv: `1607.04606[cs]`.

[87] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. *Deep contextualized word representations*. Mar. 22, 2018. DOI: `10.48550/arXiv.1802.05365`. arXiv: `1802.05365[cs]`.

[88] F. Wu, T. Zhang, A. H. d. Souza Jr., C. Fifty, T. Yu, and K. Q. Weinberger. *Simplifying Graph Convolutional Networks*. June 20, 2019. DOI: `10.48550/arXiv.1902.07153`. arXiv: `1902.07153[cs,stat]`.

[89] AllenAI. *Results provided in the spreadsheet by the authors of SciRepEval*. `https://docs.google.com/spreadsheets/d/1JMq-jR4M8KU119cvglUDmMwwzd60Z3vyvn3VqhPn9EY/view#gid=1450677429?usp=sharing`. 2023.

[90] D. Mercier, S. T. R. Rizvi, V. Rajashekar, A. Dengel, and S. Ahmed. "ImpactCite: An XLNet-based method for Citation Impact Analysis". In: *Proceedings of the 13th International Conference on Agents and Artificial Intelligence*. 2021, pp. 159–168. DOI: `10.5220/0010235201590168`. arXiv: `2005.06611[cs]`.

[91] D. Hu, X. Hou, X. Du, M. Zhou, L. Jiang, Y. Mo, and X. Shi. *VarMAE: Pre-training of Variational Masked Autoencoder for Domain-adaptive Language Understanding*. version: 1. Nov. 1, 2022. arXiv: `2211.00430[cs]`.

[92] B. Bhattacharya, I. Burhanuddin, A. Sancheti, and K. Satya. "Intent-Aware Contextual Recommendation System". In: *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. Nov. 2017, pp. 1–8. DOI: `10.1109/ICDMW.2017.8`. arXiv: `1711.10558[cs,stat]`.

[93] S. Mysore, T. O'Gorman, A. McCallum, and H. Zamani. "CSFCube - A Test Collection of Computer Science Research Articles for Faceted Query by Example". In: Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2). Nov. 6, 2021.

[94] S. Zhang, R. Xu, C. Xiong, and C. Ramaiah. *Use All The Labels: A Hierarchical Multi-Label Contrastive Learning Framework*. Number: arXiv:2204.13207. Apr. 27, 2022. arXiv: `2204.13207[cs]`.

[95] M. Parisot and J. Zavrel. "Multi-objective Representation Learning for Scientific Document Retrieval". In: *Proceedings of the Third Workshop on Scholarly Document Processing*. Gyeongju, Republic of Korea: Association for Computational Linguistics, Oct. 2022, pp. 80–88.

[96] Q. Chen, A. Allot, R. Leaman, R. Islamaj, J. Du, L. Fang, K. Wang, S. Xu, Y. Zhang, P. Bagherzadeh, S. Bergler, A. Bhatnagar, N. Bhavsar, Y.-C. Chang, S.-J. Lin, W. Tang, H. Zhang, I. Tavchioski, S. Pollak, S. Tian, J. Zhang, Y. Otmakhova, A. J. Yepes, H. Dong, H. Wu, R. Dufour, Y. Labrak, N. Chatterjee, K. Tandon, F. A. A. Laleye, L. Rakotoson, E. Chersoni, J. Gu, A. Friedrich, S. C. Pujari, M. Chizhikova, N. Sivadasan, S. Vg, and Z. Lu. "Multi-label classification for biomedical literature: an overview of the BioCreative VII LitCovid Track for COVID-19 literature topic annotations". In: *Database* 2022 (Aug. 31, 2022), baac069. ISSN: 1758-0463. DOI: `10.1093/database/baac069`.