LEARNING CITATION-AWARE REPRESENTATIONS FOR SCIENTIFIC PAPERS

A Thesis Submitted to
the Graduate School of Engineering and Sciences of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCES

in Computer Engineering

by
Ege Yiğit ÇELİK

June 2024
İZMİR

We approve the thesis of **Ege Yiğit ÇELİK**

Examining Committee Members:

Assoc. Prof. Dr. Selma TEKİR
Department of Computer Engineering, İzmir Institute of Technology

Asst. Prof. Dr. Damla OĞUZ
Department of Computer Engineering, İzmir Institute of Technology

Assoc. Prof. Dr. Özgü CAN
Department of Computer Engineering, Ege University

Supervisor, Assoc. Prof. Dr. Selma TEKİR
Department of Computer Engineering
İzmir Institute of Technology

Prof. Dr. Onur DEMİRÖRS
Head of the Department of
Computer Engineering

Prof. Dr. Mehtap EANES
Dean of the Graduate School of
Engineering and Sciences

ACKNOWLEDGMENTS

Firstly, I would like to express my sincere gratitude to my advisor Assoc. Prof. Dr. Selma TEKİR for her guidance, help, and continuous support for this thesis study.

I would also like to thank my family for their love and support over the years.

ABSTRACT

LEARNING CITATION-AWARE REPRESENTATIONS FOR SCIENTIFIC PAPERS

In the field of Natural Language Processing (NLP), the tasks of understanding and generating scientific documents are highly challenging and have been extensively studied. Comprehending scientific papers can facilitate the generation of their contents. Similarly, understanding the relationships between scientific papers and their citations can be instrumental in generating and predicting citations within the text of scientific works. Moreover, language models equipped with citation-aware representations can be particularly robust for downstream tasks involving scientific literature. This thesis aims to enhance the accuracy of citation predictions within scientific texts. To achieve this, we hide citations within the context of scientific papers using mask tokens and subsequently pre-train the RoBERTa-base language model to predict citations for these masked tokens. We ensure that each citation is treated as a single token to be predicted by the mask-filling language model. Consequently, our models function as language models with citation-aware representations. Furthermore, we propose two alternative techniques for our approach. Our base technique predicts citations using only the contexts from scientific papers, while our global technique incorporates the titles and abstracts of papers alongside the contexts to improve performance. Experimental results demonstrate that our models significantly surpass the state-of-the-art results on two out of four benchmark datasets. However, for the remaining two datasets, our models yield suboptimal results, indicating potential for further improvement. Additionally, we conducted experiments on sampled datasets to examine the effects of inherent factors on the datasets and to identify correlations between these factors and our results.

ÖZET

BİLİMSEL MAKALELERİN ALINTI YAPILARINI DİKKATE ALAN GÖSTERİMLERİN ÖĞRENİMİ

Doğal Dil İşleme alanında, bilimsel belgelerin anlaşılması ile ilgili çalışmalar büyük zorluklar içermektedir ve derinlemesine incelenmeye ihtiyaç duymaktadır. Bilimsel makalelerin anlaşılması, içeriklerinin daha etkin bir şekilde oluşturulmasını sağlayabilir. Ayrıca, bilimsel makaleler ile içlerindeki alıntıları arasındaki ilişkinin anlaşılması, bilimsel metinlerde alıntı oluşturma ve tahmin etme süreçlerinde önemli bir rol oynayabilir. Alıntı yapılarını dikkate alan gösterimlere sahip dil modelleri, bilimsel literatürle ilgili diğer görevlerde de önemli etkilere sahip olabilir. Bu tez, bilimsel metinlerdeki alıntı tahminlerinin doğruluğunu artırmayı hedeflemektedir. Bu hedef doğrultusunda, bilimsel makalelerden alınan kesitlerde maskeler kullanılarak alıntılar gizlenmiş ve ardından bu maskeleri tahmin etmek için RoBERTa-base dil modeli daha fazla eğitilmiştir. Her bir alıntının, maske dolduran dil modelleri tarafından tek bir maske için tahmin edilebilecek şekilde olması gerekmektedir. Bu süreç sonunda, modellerimiz alıntıları dikkate alan gösterimlere sahip olmuştur. Ayrıca, bu çalışmada alıntı tahmini için iki alternatif teknik geliştirilmiştir. Temel tekniğimiz sadece bilimsel makalelerdeki paragraf kesitlerini kullanarak alıntıları tahmin ederken, küresel tekniğimiz makalelerin başlıklarını ve özetlerini de kullanarak alıntıları tahmin etmeyi hedeflemektedir. Küresel modelimizin sahip olduğu ek bilgiler sayesinde başarısını artırması beklenmektedir. Deneysel sonuçlar, önerdiğimiz modellerin dört kıyaslama veri kümesinden ikisinde en son teknoloji sonuçlarını önemli ölçüde aştığını göstermektedir. Ancak, diğer iki veri kümesinde modellerimiz beklenenden düşük performans sergilemiş ve yöntemimizin daha fazla iyileşme potansiyeline sahip olduğunu göstermiştir. Ek olarak, veri kümelerinin özünde olan faktörlerinin etkilerini incelemek ve bu faktörler ile sonuçlarımız arasındaki ilişkileri belirlemek amacıyla örneklenmiş veri kümeleri kullanılarak da ek deneyler gerçekleştirilmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1. Motivation

In recent years, the research area of Natural Language Processing (NLP) has been revolutionized thanks to Transformers architecture and models like BERT. In many tasks of NLP research, Transformer-based models have achieved state-of-the-art results. However, specific tasks like citation prediction have not been thoroughly researched. Specifically, large language models' full potential have not been completely leveraged for the task of citation prediction, and this leaves room for improvement in this area of research.

Citations are essential building blocks in scientific writing. Their accurate placements indicate quality, as one should know the literature to claim contributions and put the current study in the context of the existing work from different aspects, such as background, method, etc. (Cohan et al. 2019).

The main goal of the citation prediction task is to predict the corresponding citation of a scientific sentence or paragraph when we hide the in-text citation information from the model. Another aspect of citation prediction is its ability to operate as a citation suggestion mechanism. For a given scientific text, the model can suggest additional papers on a similar topic. These suggestions can be considered additional reading material alongside the targeted paper, corresponding to the ground truth citation value.

Another motivation behind the citation prediction task is the importance of using this task as a basis for citation generation tools in the future. For example, scientific document understanding of NLP models has significantly increased over the years. In return, this also allowed the models to be better at scientific text generation. By this logic, it makes sense that improvements in citation prediction systems can be helpful for citation generation steps during scientific text generation. An example scenario of how citation can be predicted for a given text is shown in Figure 1.1.

There are two levels of citation prediction: the first, whom to cite, and the second, whom to cite in what context. The former is global citation prediction, traditionally per-

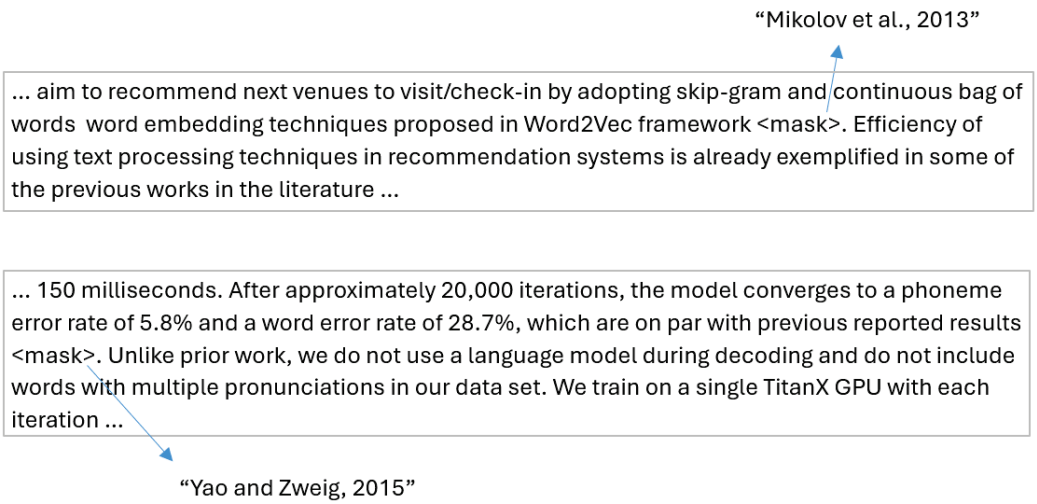

Figure 1.1. An example scenario for the citation prediction task.

formed based on paper metadata such as author names, paper titles, abstracts, conference venues, publisher information, etc. Recently, custom citation-aware language models (SPECTER (Cohan et al. 2020), SciBERT (Beltagy, Lo, and Cohan 2019) learn good citation-aware embeddings for full papers to perform well in this task. The latter task is local citation prediction, aiming to determine the target paper for a citation placeholder. Another closely related task is citation impact prediction. How a paper frames its work through citations is predictive of the citation count it will receive (Jurgens et al. 2018).

Research in the area of citation prediction is limited compared to similar tasks. Many past works have focused on the task of citation count prediction ((Dongen, Maillette de Buy Wenniger, and Schomaker 2020), (Huang et al. 2022)) instead of directly predicting citations. The term citation count refers to the number of times a paper gets cited after publication. This task can help predict a paper’s future scientific impact. However, the works in this area are disconnected from the task of citation prediction, mainly due to their tendency to ignore the contents of the scientific papers.

Language model pre-training based on Transformers (Vaswani et al. 2017) provided new state-of-the-art performances in many downstream tasks. Masked language modeling (MLM) objective is the primary learning strategy behind BERT (Devlin et al. 2019) and its variants (RoBERTa (Liu et al. 2019) etc.). Some works have focused on learning from scientific papers using these approaches. For example, SciBERT (Beltagy, Lo, and Cohan 2019) tries to pre-train a language model using the entirety of scientific papers. The main

improvement of SciBERT is its capability to learn language representations better for complex and challenging scientific data.

The task of citation prediction from context has been addressed in a handful of works. One of the leading works is Hatten (Gu, Gao, and Hahnloser 2022). Hatten is a two-step model that initially selects a group of papers related to a context from a paper and tries to find the best candidates from a given pool of papers. In its second step, the selected candidate papers get re-ranked using a fine-tuned SciBERT (Beltagy, Lo, and Cohan 2019) model, and the most relevant papers are returned.

Hatten's approach focuses on the task of local citation prediction. The authors aim to predict citations within contexts rather than relying solely on paper metadata. However, they leverage the titles and abstracts of cited papers to enhance prediction accuracy, referring to this additional information as global information. Similarly, in our study, we incorporate titles and abstracts as global information, aligning with this terminology. Throughout the remainder of this work, the term "global" denotes additional information rather than the global citation prediction task itself.

There are four benchmark datasets in this area of research. Medić and Snajder (2020) released the datasets named ACL-200, FullTextPeerRead (Peerread for short), and Refseer. Lastly, Hatten (Gu, Gao, and Hahnloser 2022) shared the Arxiv dataset.

The existing citation prediction works are not built upon Transformers but benefit from Transformers in indirect ways, such as re-ranking the results. Distinctively, we propose a language model pre-training by specifically masking citations. In concrete terms, we further pre-train a RoBERTa (Liu et al. 2019) base model using a citation masking strategy so that the model can learn how to represent citations properly. We add citation tokens to the model vocabulary beforehand. Here, the vocabulary items are in the parenthetical author-date citation style, as it's more appropriate for forming global IDs than numerical citations.

Since a citation context may not refer to a reference uniquely but relates to a set of candidate references, the next step is to extend the context with the ground-truth reference's global information, such as title and abstract, using the REALM framework (Guu et al. 2020). This extended method of pre-training has the potential to encode citation information thoroughly. In this way, we acquire citation-aware representations for scientific papers. Pre-trained in this way, the proposed model predicts contextual citations

inside the scientific texts. Our experiments show promising results on the benchmark citation prediction datasets.

Another prominent aspect of our approach is one can use citation token hidden representations as their document-level embeddings alternative to SPECTER (Cohan et al. 2020) or SciBERT (Beltagy, Lo, and Cohan 2019). In other words, these citation-aware contextual representations can capture global properties such as closeness between scientific papers.

We believe that a citation's representation, as learned by a language model, can also encapsulate the representation of its corresponding paper. Consequently, our models have applications beyond merely predicting citations. While SPECTER learns a paper's representation through its citing relationships with other papers, it is unable to predict citations or function as a language model. In contrast, our models can learn the representations of papers by predicting their citations.

Lastly, our approach possesses a unique benefit compared to similar works. Our model is a modified language model that can predict citations inside a scientific text. We do not enforce our model to only predict from a limited pool of citations. The model learns to predict citations by itself while still being able to predict any other token from its vocabulary. Thanks to this ability of our model, each learned citation token representation should be able to properly capture its contextual meaning in the entirety of the language model. Additionally, it is possible to fine-tune our model further for citation-related tasks to benefit from the citation-aware representation of each citation. For instance, the task of text generation for scientific papers involves generating citations within the texts. By fine-tuning our models for this task, we can leverage their ability to accurately predict citations, thereby ensuring that the generated texts contain more precise and relevant citations.

1.2. Contributions

Our main contributions can be summarised as follows. Firstly, we propose an approach that further pre-trains the RoBERTa model with a modified tokenizer to predict citations inside scientific texts. Secondly, our model continues to operate as a language model, unlike past works that used Transformer-based models for purposes like re-ranking. This allows our model to be more flexible compared to past approaches. Another contribu-

tion of our approach is that our models can work as a suggestion mechanism thanks to their ability to predict citations other than the ground truth. Also, we have performed additional observations on the benchmark datasets of the citation prediction task and pinpointed particular challenges with these datasets. Lastly, we have observed certain statistics associated with the datasets, like context per citation counts, to showcase its effects on the success of our approach. We have also performed multiple sampling trials on our large datasets better to understand the effects of dataset sizes on the results.

1.3. Outline of Thesis

The contents of the remaining chapters can be described as follows. Firstly, we provide background information and explain the works related to our research in Chapter 2. In Chapter 3, we explain our approach and the techniques we have used to investigate our results further. We provide our results and discuss them in detail in Chapter 4. This chapter also contains the details of an additional ablation study alongside a part that discusses the limitations and disadvantages of our approach. Lastly, we give our concluding remarks and discuss potential future work in Chapter 5.

CHAPTER 2

BACKGROUND AND RELATED WORK

This chapter discusses basic concepts of the Natural Language Processing area and its techniques, which are related to the task of citation prediction. Firstly, we explain the basics behind language models and the effect of Transformers on this area of research in Section 2.1. Then, we discuss the details of the BERT model and its learning strategy for language models in Section 2.2. Strategies of pre-training and fine-tuning alongside additional BERT model alternatives have been explained in Section 2.3. In Section 2.4, we discuss tasks that are adjacent to citation prediction. Lastly, we explain the task of citation prediction in detail and discuss the concept of citation awareness in Section 2.5.

2.1. Language Models

Natural Language Processing (NLP) is a discipline of artificial intelligence (AI) research. The primary focus of NLP encompasses the comprehension, utilization, and generation of natural language texts. The intricacies inherent in spoken languages present numerous complexities. Consequently, NLP research is tasked with devising solutions capable of surmounting these multifaceted challenges.

NLP encompasses numerous tasks and research topics that have been extensively studied over the years. One of the most significant tasks is classification, which involves selecting an option from two or more candidates based on specified criteria. Examples of this include sentiment classification and spam classification. Other notable tasks within this domain include named entity recognition, question answering, and information retrieval.

In recent years, numerous techniques have been developed within the NLP domain. However, specific topics within NLP continue to hold considerable potential for further development. One such topic is the task of citation prediction.

Another crucial task in NLP is text generation, which aims to produce texts that emulate human speech using various techniques. The most prominent technique for text

generation is the use of language models. Language models seek to capture the essence of a spoken language by employing various methods to determine the likelihood of word co-occurrence within sentences. These models treat sentences as sequences of words and aim to predict the probability of subsequent words. Fundamentally, a language model attempts to predict the next word given a sequence of words.

Language models initially started out as statistical language models. One such statistical technique is n-grams (Brown et al. 1992). The goal of this approach is to find which words are more likely to appear together in groups. The ”n” represents the number of words in the selected groups. While this is a simple and efficient approach, it fails to capture long-term connections between words and cannot generate sentences that are semantically correct in general.

With the rise of neural networks, certain techniques like RNNs and LSTMs have become the main techniques for language models. Recurrent Neural Networks (RNNs) (Elman 1990) can take previous outputs into consideration for their next set of inputs. This is particularly useful when the model is trying to predict the next word while still considering past word outputs during text generation. This allows RNNs to operate with a certain level of memory while generating sentences. However, this approach can be computationally expensive and still fail to predict words in certain situations properly. For example, the first word of a sentence will barely have any effect on the last word if the sentence is very long. RNNs generally consider more recent output words to be more significant, while older output words are more likely to be forgotten. This problem is called vanishing gradients.

To solve this problem, a technique called Long Short-term Memory (LSTM) (Hochreiter and Schmidhuber 1997) has been introduced. This approach tries to improve the memory of RNNs by adding a cell mechanism that is capable of selectively discarding or retaining information from the memory. The LSTM network is more capable of preserving information that belongs to the start of longer sequences thanks to its cell mechanism. However, this approach is still computationally expensive since it takes its inputs sequentially.

After these techniques, the model architecture called Transformers (Vaswani et al. 2017) has been introduced. This architecture possessed clear benefits compared to older techniques, and it is currently the most dominant approach used in the NLP area.

The transformers are an advanced neural network approach that is highly capable of understanding the context and relationship between sequential data. Their main advantage is their ability to process whole sequences at once instead of one-by-one like past approaches. Figure 2.1 shows the overall architecture of Transformers.

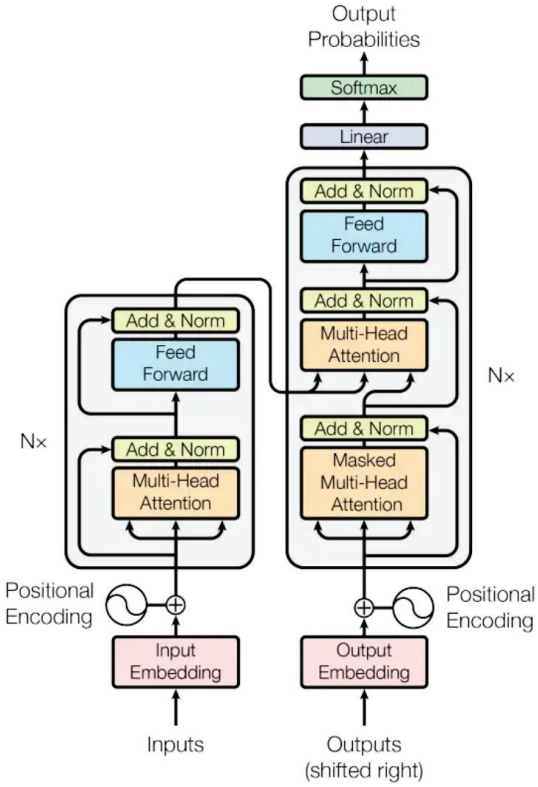

Figure 2.1. Transformers architecture from the paper of Vaswani et al. (2017).

The main components of the Transformers architecture are the self-attention mechanism and encoder-decoder structure. The encoder-decoder structure is made up of multiple layers of encoders and decoders that transform an input into its embedding and the embedding into its output. The attention mechanism helps the model focus on different parts of the input and allows the model to learn from the input better. The self-attention is a strategy that causes the model to focus on different parts of the input, and the model improves its learning capabilities as a result. These components help this architecture to learn the more complex connections between inputs and outputs.

2.2. BERT Model

One of the most prominent models in recent years is a Transformer-based model called BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al. 2019). Currently, models with Transformers (Vaswani et al. 2017) architecture and especially BERT language models are the focus of NLP research. One of the most prominent techniques that have come from BERT is the task of mask-filling. The models that have been trained with this objective are called Masked Language Models (MLM).

BERT model contains only encoders in its design. Unlike the Transformers architecture, the BERT model does not make use of decoder layers since it mainly focuses on understanding inputs instead of generating outputs. Although BERT is only made up of encoders, it still makes use of Transformers' self-attention strategy. Thanks to these design choices, it can perform significantly well on tasks like classification, question answering, and NER.

The most important advantage of BERT compared to older methods is that it can take its inputs into consideration bidirectionally. Traditional language models process text either from left-to-right or right-to-left. However, BERT is capable of processing the entirety of the context of a given sentence or paragraph from both sides at the same time. It does not need to take its inputs sequentially.

In principle, the main goal of BERT is to operate as a language model. As with other language models, BERT also aims to predict words to understand natural languages. BERT tries to train as a language model using two strategies. Firstly, it devises a strategy called the Masked Language Model (MLM). In this strategy, approximately 15% of the words of an input sentence are selected. 80% of the selected tokens are replaced with a special mask token. 10% of the tokens are replaced with random tokens and the remaining ones are left untouched. Afterward, BERT tries to predict these hidden words, and this task is referred to as mask-filling.

The BERT model has a tokenizer that accompanies it. This tokenizer can process the given inputs to provide the BERT model with proper tokenized inputs. Tokenizer also contains the vocabulary of the model. BERT tries to predict masked words using its tokenizer vocabulary. Vocabulary does not only contain full words, but it also possesses necessary stop-words, special symbols, sub-words, etc. The BERT model can also learn

a new word that was not inside its vocabulary thanks to this masking strategy. It is also possible to manually provide additional words or special tokens to the model's tokenizer as well.

The second strategy of BERT is called Next Sentence Prediction (NSP). BERT tries to predict the second sentence using the first sentence input while assuming both sentences are connected. It marks the first sentence's beginning with a special "[CLS]" token and separates the second sentence with a "[SEP]" token.

During the training process, BERT attempts to learn using both strategies. The authors of BERT suggest that this allows BERT to be a flexible and robust language model that is able to consider both the contents of a sentence and connections between sentences. BERT is capable of capturing a sentence's meaning both semantically and syntactically. It can learn the complex intricacies of natural languages significantly better than older language models.

The authors of BERT have released two different versions of BERT. These are called BERT-Base and BERT-Large. They have been trained using the same strategies while possessing highly different sizes and capabilities. BERT-Base contains 12 encoder layers and $110M$ parameters. Meanwhile, BERT-Large possesses 24 encoder layers and $340M$ parameters.

Naturally, when trained on the same tasks, BERT-Large provides superior results compared to BERT-Base. BERT-Large's increased size allows it to have a deeper understanding of the meaning and connections of a natural language. However, BERT-Large requires larger storage spaces and longer training times due to its size. This leads to a need to balance between larger and smaller models depending on the requirements of an NLP task.

In recent years, Transformers models like BERT can be easily accessed using the Transformers library of Python programming language. This library contains many necessary tools for preprocessing a dataset and training a model. Especially, BERT and its derivatives can easily be accessed and trained using the Transformer library.

2.2.1. BERT Tokenizer

The BERT model and its derivatives have their own tokenizers to process their input data. These models use a unique tokenization strategy to acquire the embeddings (i.e., representations) of words in an input. BERT tokenizer generates the embeddings of input words alongside their segment embeddings and positional embeddings. Segment embeddings denote which segment a specific token belongs to in the provided input. Segments of an input text can be designated using a special separator token. Meanwhile, positional embeddings are generated using the position index of the tokens. These embeddings can be learned and improved further during the pre-training of BERT models.

BERT tokenizer can also learn and represent words it has never seen before. BERT tokenizers possess their own vocabularies as well. BERT can handle unknown words by breaking them into known sub-words. Known sub-words refer to the words that are inside the vocabulary of the BERT tokenizer. For example, the word ”sleeping” can be broken into ”sleep” and ”##*ing*” using the BERT tokenizer. The tokenizer's vocabulary contains the word ”sleep”. Thanks to the sub-word strategy of BERT, it does not need to contain the full word ”sleeping”. Thus, the model can learn the representations of new words by simply breaking them up.

BERT-like RoBERTa (Liu et al. 2019) model also has its unique tokenizer. It has multiple differences from the BERT tokenizer. Firstly, RoBERTa uses a larger byte-level BPE vocabulary with 50K sub-word units instead of the character-level BPE vocabulary of size 30K that was used for BERT. Secondly, RoBERTa does not have token-type IDs. So, it does not need to show the position of each token according to the token's segment. It is enough to simply use a separator token to designate segments of a text. Lastly, RoBERTa uses different symbols and conventions for its tokenization processes. For example, BERT uses ”##” symbol for sub-words while RoBERTa uses ”$\acute{G}$”. Another convention difference between BERT and RoBERTa is their separator tokens, which are ”[SEP]” and ”</s>”, respectively.

In Figure 2.2, we have shown the tokenization of an example sentence using the RoBERTa tokenizer. The word ”stochastic” did not exist in RoBERTa's vocabulary. However, the model is able to learn its embedding by breaking it into ”sto”, ”ch”, and ”astic” sub-words. Another interesting aspect of this tokenizer is the $\acute{G}$ symbol before

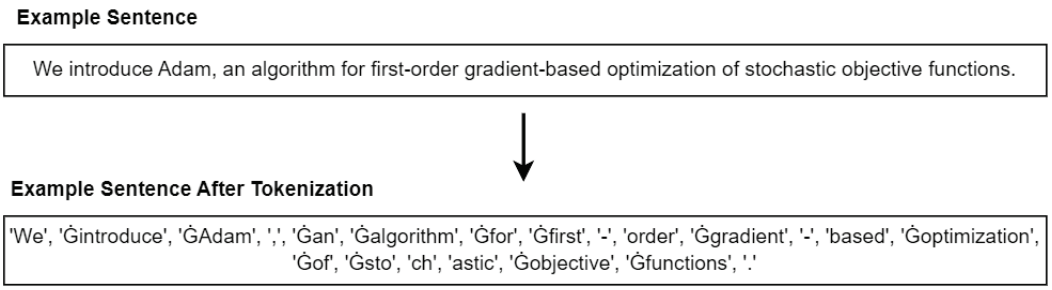

Figure 2.2. Tokenization of an example sentence using default RoBERTa-base tokenizer.

certain words. RoBERTa tokenizer uses these special symbols to mark the beginning of each individual word in the input sentences. For example, the ”ch” sub-word of ”stochastic” does not possess this special symbol since it is not at the beginning of the full word.

2.3. Pre-training, Fine-tuning, and BERT Alternatives

BERT model training has two main stages: pre-training and fine-tuning. Pre-training is a stage where the BERT model is trained from scratch for both MLM and NSP tasks. This stage requires a large amount of unlabeled data. As a result, the pre-trained model can be used as a versatile language model that can predict masked tokens inside texts. The fine-tuning stage consists of further training the BERT model for a specific NLP task. These tasks are referred to as downstream tasks.

The BERT model needs to be fine-tuned on a downstream task to achieve improved results. The fine-tuning process is much faster than the pre-training process, requiring labeled data for its downstream task. The BERT model's ability to adapt to any downstream task with minimal modifications is a significant sign of how BERT is a very versatile model for NLP research in general.

Another advantage of BERT is that it does not need word embeddings provided beforehand. Older language model techniques required pre-made frozen embeddings like Glove (Pennington, Socher, and Manning 2014), Word2Vec (Mikolov et al. 2013), etc. Word2Vec learns high-dimensional word embeddings using vanilla neural networks to predict a word according to its context or predict a context according to a target word. The

former approach is called CBOW (Continuous Bag-of-Words), while the latter is called the Skip-Gram method. On the other hand, Glove performs a similar process for learning word embedding while using matrix factorization techniques rather than neural networks.

Word2Vec and Glove techniques were static embeddings that stayed the same during the training of models. Because of this situation, they occasionally failed to represent the true contextual meanings of words correctly. However, the BERT model's tokenizer can learn and readjust word embeddings by itself. It can also further improve the embeddings of words during fine-tuning. This allows BERT to learn the embeddings of entirely new words or tokens as well.

BERT-Base and BERT-Large are not the only pre-trained models for this area of research. There are many different models derived from the core ideas of BERT, which have been released over the years. One such model is RoBERTa (Liu et al. 2019). RoBERTa removes the NSP strategy of the BERT model since the authors believe that the MLM goal is the main reason behind the success of the BERT model. They claim that the NSP strategy's contribution to the results of BERT is minimal.

The authors of RoBERTa perform multiple refinements on the BERT model as well. The resulting RoBERTa model can outperform BERT on various NLP tasks. For this reason, many recent papers have preferred to use RoBERTa as a pre-trained model to fine-tune rather than BERT. This also shows that BERT might not be completely optimized as it had clear room for improvement.

There are other models derived from BERT, as well. For example, ERNIE (Sun et al. 2019) allows multitasking in its pre-training step. This allows it to better understand different uses of a language while pre-training for multiple tasks. The authors describe ERNIE as capable of learning how to learn itself thanks to this multitasking structure.

Another example is SpanBERT (Joshi et al. 2020), which uses a different masking strategy than BERT. While BERT randomly chooses 15% of tokens for masking, SpanBERT randomly chooses groups of consecutive tokens to mask. The authors claim that this helps the model to learn to predict consecutive token groups so that the model can understand the meanings of word groups and their connections to the rest of the sentence better.

Also, there is PMI-BERT (Levine et al. 2020) model, which chooses which tokens to mask according to a statistic called Pointwise Mutual Information (PMI) rather than

replacing them randomly.

There is also SciBERT (Beltagy, Lo, and Cohan 2019) model, which focuses on learning from the entirety of scientific papers. BERT generally learns from a corpus that is heavily varied in its content. However, the pre-training of the SciBERT model is performed exclusively on scientific texts. Specifically, SciBERT is pre-trained on a randomly sampled dataset of size $1.14M$ from the Semantic Scholar database. This sampled corpus comprises 18% papers from the computer science domain and 82% from the broad biomedical domain. SciBERT also uses its own in-domain vocabulary called SciVocab. The resulting token overlap between BERT-Base's BaseVocab and SciVocab is 42%.

The authors of SciBERT show that its results can outperform BERT on downstream tasks that involve scientific data. Another benefit of SciBERT is its ability to be trained on the entire text of scientific papers instead of only titles and abstracts of scientific papers. Generally, models like BERT are limited to a 512-token input size. To overcome this challenge, the authors of SciBERT have used a technique that splits the full text of scientific papers so that the split part of papers can fit into the limit of 512 tokens. They have observed that the average length of papers is 2796 tokens and split them using ScispaCY (Neumann et al. 2019), which is optimized for scientific text.

The work of Cohan et al. (2020) named SPECTER manages to create document embeddings of scientific papers from their titles and abstracts. They have managed to represent a paper P as a dense vector v to acquire an appropriate representation of the paper. They can also use these embeddings for downstream tasks.

$$L = max((d(P^Q, P^+) - d(P^Q, P^-) + m), 0) \tag{2.1}$$

SPECTER is trained using a triplet margin loss function described by Equation 2.1, where m is the loss margin parameter and d is a distance function. The distance function, as depicted in Equation 2.2, utilizes L2 norm distance, where v_A is the dense vector representation of paper A. P^Q denotes the target paper whileP^+ and P^- represent

positive and negative example papers selected for the target, respectively. Positive papers are chosen from the reference list of the target paper, whereas negative papers are randomly selected papers that are not cited by the target paper. SPECTER aims to minimize the distance from positive papers while simultaneously maximizing the distance from negative papers, thereby learning an optimal representation of paper Q.

$$d(P^A, P^B) = \|v_A - v_B\|_2 \tag{2.2}$$

SPECTER does not function as a language model and therefore cannot predict citations; however, it excels in learning and representing scientific papers accurately. In our approach, our goal is to develop paper representations akin to those generated by SPECTER. The authors of SPECTER have also considered using the full text of papers in their approach since it could provide the complete picture of scientific papers. However, they have opted to leave this improvement as an item of future work.

Another interesting BERT model that has been released by Chalkidis et al. (2020) is called LegalBERT. This model aims to improve the results of the BERT model in the field of law. By specifically pre-training a BERT model for this field, they achieve significant improvements in downstream tasks related to law compared to directly fine-tuning vanilla BERT on law-related downstream tasks. This also shows the importance of pre-training domain-specific models to perform better on those domains. Using vanilla BERT with fine-tuning on a specific domain-related task may not always achieve the best results. Instead, it is important to pre-train domain-specific BERT-like models from scratch as well.

Lastly, there are Transformer-based models that use fundamentally different strategies from BERT. GPT (Generative Pre-trained Transformer) (Radford et al. 2018) models are pre-trained using only decoder layers instead of encoder layers like BERT. The main reason for this design choice is that GPT models are more focused on text generation compared to BERT models. Another type of Transformer-based model is T5 (Text-To-Text Transfer Transformer) (Raffel et al. 2020), which uses both encoder and decoder layers.

2.4. Adjacent Tasks to Citation Prediction

The citation prediction task aims to guess the correct citation value for a given scientific text section. This scientific text may come from any part of a paper. However, this text should contain a reference to another scientific paper inside it. The model trained for the citation prediction task should be able to predict which paper is being referenced in the given text. Depending on the approach to solving this problem, the prediction can be based on any information that belongs to either paper. Information from both the referencing paper and the referenced paper can be useful for this task. This information can consist of details like author names, paper publishing information, year, title, abstract, other referenced papers, commonly referenced papers in both papers, etc.

While the specifics of this task appear to be simple, it does not have nearly enough research papers focusing on it. In the area of citation prediction, there is a limited amount of research papers. Instead, most of the past works have focused on the Citation Count Prediction task. Citation counts refer to the number of instances where a paper will be cited after it has been published. While this task may appear to be similar to the citation prediction task, it is a fundamentally different task. However, it is a useful task that can show the potential of a paper before it is published. The ability to predict the impact of a paper beforehand can also be considered an important task. Many past works focus on this task as well. We are also going over the details of this task since it can be considered adjacent to the task of citation prediction.

An example paper in this area belongs to Brody and Harnad (2005), where they tried to predict the future citations of a paper using web usage statistics. They have mainly focused on web access counts of a citation and its age to predict another paper's future number of online accesses. Meanwhile, Abrishami and Aliakbary (2019) used a SimpleRNN model to predict long-term citations using short-term citations. Interestingly, works on this task made use of Transformer-based models before the works on citation prediction. In fact, the works on citation prediction still have not made complete use of modern model architectures in their approaches.

Additionally, Bai, Zhang, and Lee (2019) used a measure called Paper Potential Index (PPI), which is based on a combination of certain manually acquired features like the inherent quality of a paper, a paper's impact decaying over time, the impact of a paper's

early citers, etc. The authors of this paper suggest that their manually acquired features can be used together to calculate a measure that can predict a paper's future impact. There is also SChuBERT (Dongen, Maillette de Buy Wenniger, and Schomaker 2020). It uses BERT architecture, and it is trained using certain contents of scientific papers like titles, abstracts, section names, etc. While it uses similar data and tools with citation prediction task's models, SChuBERT still aims to predict a paper's future citation count and impact.

From the perspective of citation prediction task, all these research papers focus on a fundamentally different task because none of them aim to predict citations inside paragraphs of a scientific text. They do not consider the contents of a scientific text and look for which paper that text may be referring to. The works on citation count prediction tasks generally ignore citation connections inside the paragraphs of a paper. Instead, they sometimes focus on the bibliography section at the end of the papers while trying to predict the future impact of a paper.

2.5. Task of Citation Prediction

The main focus of our research is the task of citation prediction. Only a handful of works in recent years have been released on this task. This might be caused by the difficulty of using limited information from a scientific paper to predict which different scientific paper it may be referring to. In contrast, it might be easier to predict which scientific paper the given context belongs to. However, trying to predict which scientific paper is being referred to requires the trained model to capture a deeper semantic connection between papers, and this relatively increases the difficulty of the task.

An initial work on this task belongs to Yu et al. (2012). They aim to predict papers based on their three main features. These features are authors, venues, and keywords. They use a meta-path-based prediction model that makes use of probabilities. However, they do not make use of the contexts of scientific texts and simply use their superficial features for their predictions. They also ignore the contents of titles and abstracts since they cannot make use of them in a probabilistic prediction model. Additionally, their goal is slightly different from citation prediction since they try to predict which paper the given features belong to rather than trying to predict which papers it may be referencing.

Another work that focuses on citation prediction belongs to Tanner and Charniak

(2015). They have also focused on the actual task of citation prediction using a technique they refer to as Hybrid Generative Discriminative Approach. It is done by training a model called LDA-Bayes using manually crafted features like "number of overlapping authors" or "author's previously cited sources". The authors still do not make use of the contents of scientific papers due to these manually crafted features. However, they are specifically trying to predict a given paper's potential citation connection. Hence, it can be said that their research is exactly on the citation prediction task instead of any adjacent tasks.

The work of Luo et al. (2023) also can be considered adjacent to the task of citation prediction. Specifically, their work is much closer to the citation prediction task compared to the other works in the citation count prediction task. Their work's only difference is that it is focused on a different field's data and potential usage areas instead of scientific papers. The authors use BERT architecture models like RoBERTa and LegalBERT (Chalkidis et al. 2020) and train the models using the full texts of legal data. Hence, their approach is limited to the area of law. They have created a dataset called PACER to specifically test predicting masked provision/law names in legal texts, which can be considered relatively close to the general citation prediction task. PACER datasets contain three components: 1) Past legal records that show the reasoning of a lawyer are referred to as precedents. 2) The definition of laws contained in provisions of the legislature is shortened to provisions. 3) Contexts that contain the masked provision names and their ground truth values are also within PACER.

Luo et al. (2023) aim to predict provision names by minimizing L2 normalized distance between contexts, precedents and provision names. Afterward, they calculate a score value for each provision using the aforementioned distances. They pass these scores through a feedforward layer and select the top-scoring provision name for the target mask inside a context. Additionally, the authors do not fill the mask inside the context in the general technique of BERT language models. Instead, they choose a provision/law name out of the PACER dataset. The names of provisions are required to be inside the dataset so that they can be scored and predicted. Since the contents of the PACER dataset and the mask-filling technique of the model are fundamentally different from the general citation prediction task, we cannot directly use these for our purposes.

The study by Medić and Snajder (2020) trains a Bi-LSTM model to determine similarity scores between a target paper and its candidate papers. The target paper provides

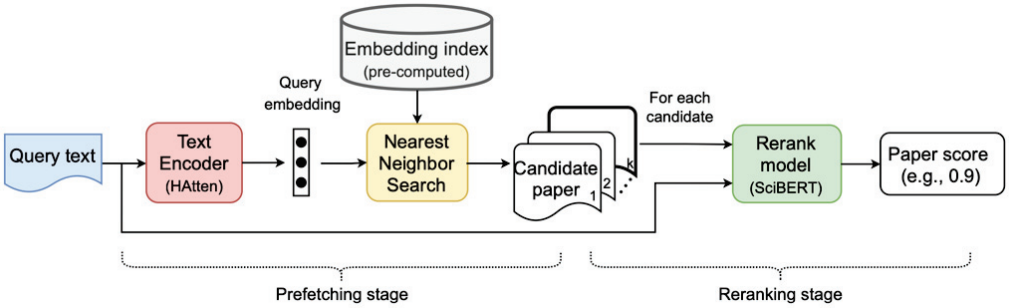

Figure 2.3. Overall process of Hatten from the work of Gu, Gao, and Hahnloser (2022).

a context with a hidden citation, and the model utilizes the titles and abstracts of candidate papers to calculate their similarity scores. The candidate papers with the highest similarity scores are selected as the predicted citations for the target paper.

Lastly, there is the work of Gu, Gao, and Hahnloser (2022) called Hatten. Hatten uses a Hierarchical Attention Text Encoder and SciBERT-based Re-ranking scheme for the task of citation prediction. It starts by pre-fetching potential candidate papers from a pool of citations. Then, it re-ranks these pre-fetched papers to find the best recommendation among them. The overall process of Hatten is shown in Figure 2.3.

The authors of Hatten consider this technique for the task they refer to as "Local Citation Recommendation". Their pre-fetching mechanism uses local citation contexts, which are context windows, titles, and abstracts taken from scientific papers. Initially, they perform pre-fetching out of a very large pool of scientific papers. So, the results of pre-fetching are used to reduce the amount of candidate papers to choose from. Additionally, the term context can be defined as a text section taken out of a paper's paragraph that contains the citation in its middle position. Both sides of the citation should generally have an equal number of tokens.

Afterward, their proposed SciBERT re-ranker technique uses global citation contexts, which also include the cited paper's title and abstract. Its purpose is to re-rank the previously pre-fetched citations to have a better rank order according to their closeness to the initial paper which the context, title and abstract originated from. SciBERT was able to understand better connections between papers to acquire a better ranking of predicted citations, thanks to its versatility. Since pre-fetcher reduces the number of papers that need to be re-ranked, the SciBERT model is able to work in a more efficient manner. The

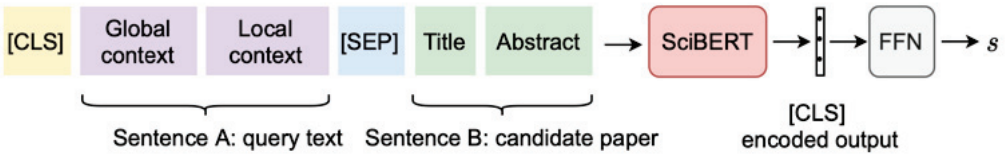

Figure 2.4. Overall look at the SciBERT re-ranker of Hatten from the work of Gu, Gao, and Hahnloser (2022).

overall look at the SciBERT re-ranker is shown in Figure 2.4.

Using this approach, the authors of Hatten have achieved considerably good results on the four benchmark datasets, which are called ACL-200, FullTextPeerRead, Refseer, and Arxiv. The initial three datasets belong to the work of (Medić and Snajder 2020) while the last one has been made by the authors of Hatten. ACL-200 and FullTextPeerRead datasets are relatively smaller datasets compared to Refseer and Arxiv datasets. Because of this reason, the models for citation prediction task can be evaluated on both smaller and larger datasets to observe the adaptability of the models.

The authors of Hatten leveraged the SciBERT re-ranker to significantly improve the results of the pre-fetching step. While using the SciBERT re-ranker, the authors have used titles, abstracts, and contexts to fine-tune the model. However, the authors did not provide the details of this step in their results. We also needed to use a similar approach in our training steps as well. So, to solve this problem, we have used the approach of REALM (Guu et al. 2020) as inspiration.

Authors of REALM have proposed a retrieval-based language model pre-training. In their retrieval step, they needed to use multiple pieces of global information tied together. To achieve this, they simply connected two sentences using a special separator token like ”[SEP]” and provided these connected sentences as input to the model. Transformers-based models like BERT can easily discern the connection between two inputs separated by a special symbol and adjust their learning process accordingly. Similar to this approach, we have also connected contexts, titles, and abstracts of scientific papers using a separator token when trying an approach similar to Hatten’s SciBERT re-ranker.

Our work shares multiple similarities with related research; however, previous works exhibit specific limitations when compared to our model. For example, Hatten can only predict from a given pool of papers by re-ranking its initial predictions, leveraging the

SciBERT model for re-ranking rather than utilizing its full language model capabilities. In Table 2.1, we illustrate the differences between the capabilities of previous works and our approach. Initially, we compared which approaches can function as language models in their final states. Additionally, we grouped the tasks of predicting citations and legal provisions into a single comparison category due to their similarity. Furthermore, we evaluated whether the approaches can represent papers and citations. The ability to represent papers or the citations referring to those papers can be considered roughly equivalent. Both SPECTER and our approach generate embeddings that can represent papers.

Table 2.1. Comparison between the capabilities of our approach and related works.

Approach Capabilities	RoBERTa	SciBERT	SPECTER	Luo et al. (2023)	Medić and Snajder (2020)	Hatten	*Our approach*
Language model	+	+	-	-	-	-	+
Can predict citations or legal provisions	-	-	-	+	+	+	+
Representations of papers and citations	-	-	+	-	-	-	+

Our approach retains its functions as a language model after custom pre-training. The ability to extend a model by incorporating newly published papers each year without requiring complete re-training is a significant advantage. Our approach also benefits from this capability, as we can continue training our models on a corpus that includes more recent papers, thereby enhancing the predictive capabilities of our models. It is more cost-effective to fine-tune the model for a smaller subset of recent papers than to retrain the model on the entire dataset with slight extensions.

CHAPTER 3

METHODOLOGY

We propose a custom masking strategy to predict citations by treating every citation as a single token. To meet this objective, we further pre-trained RoBERTa-base (Liu et al. 2019) on the existing benchmarks for citation prediction. The existing datasets provide citation contexts from various articles where all contexts have a target citation in the middle. The context sizes are counted in terms of characters, which causes some incomplete words at the start and end of the contexts.

Figure 3.1. Three citation instances in the parenthetical author-date citation style.

Before training with the masked language modeling (MLM) objective, the token set needs to be slightly modified by adding new items representing whole citations in the parenthetical author-date citation style. Figure 3.1 shows some citation instances in this style. Each added token corresponds to a paper's citation from the benchmark datasets. Since we aim to predict complete citation names from single masks inside contexts, we must treat each citation name as a token. Figure 3.2 shows the difference between a tokenized sentence before and after adding citation tokens to the tokenizer.

We do not refer to our approach as fine-tuning, as the resulting model retains its functionality as a language model. Instead, we perform a custom further pre-training on

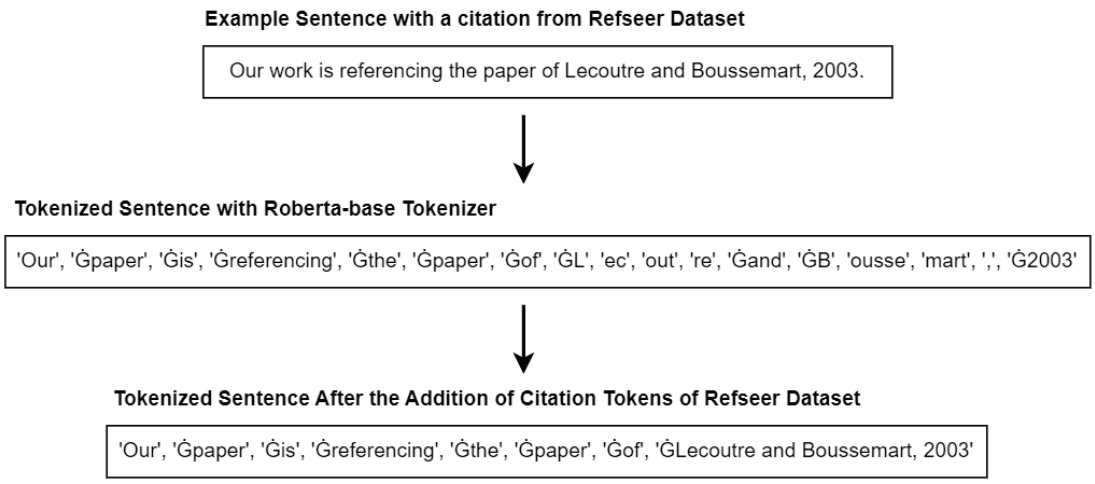

Figure 3.2. Tokenization of an example sentence from the Refseer dataset (before and after addition of citation tokens).

the RoBERTa-base language model to enable it to predict citations. The custom aspect of this approach involves the addition of citation items in the vocabularies. For the purposes of this work, we use the term ”further pre-training” to describe our methodology.

We propose two techniques for the citation prediction task: Base and Global. In our Base technique, our model is only provided masked contexts and their ground truths as inputs. The state-of-the-art work of Hatten (Gu, Gao, and Hahnloser 2022) uses contexts, titles, and abstracts as inputs for their models. We limit the inputs to only contexts to observe how our model performs compared to Hatten in a disadvantaged setting. In our Global technique, our model receives titles and abstracts similar to Hatten's approach. We aim to compare our metrics against Hatten under the same conditions. In short, the only difference between our Base and Global versions is the contents of their inputs. We provide the details of our Base approach in Section 3.1. In Section 3.7, we explain the details of our Global approach.

3.1. Base Technique: Learning Citation Representations with Contexts

During training, we provide the model with some contexts taken out from multiple articles inside the datasets. All contexts have a targeted citation in their middle points, and the size of the context is determined by character counts instead of token counts. This causes some incomplete words at the context's start and endpoints. We kept this structure

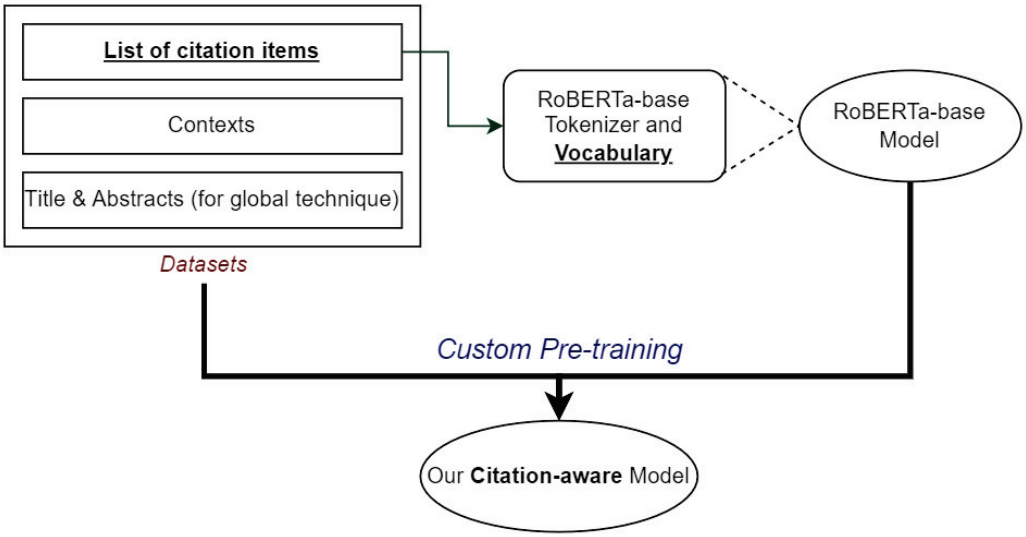

Figure 3.3. An overall look at our approach.

the same since all datasets have been provided in this manner.

The model takes the contexts after they are masked with a proper masking strategy for the target citations. Then, it tries to compare predicted mask values with the unmasked ground truth context. Our overall approach has been depicted in Figure 3.3.

Our approach depends on citations that have been added to the vocabulary. This is one of the biggest performance bottlenecks of our approach. For the larger datasets, the addition of all citation tokens to a tokenizer can take up to $2 - 3$ days. Since this is a sequential task of adding all tokens one by one, it cannot be parallelized to save time, and its performance is dependent on the processor instead of the GPU.

The work of Gu, Gao, and Hahnloser (2022) has been limited to predicting out of a large pool of papers from their datasets. However, our model can predict any word as a citation token since it also functions as a language model. Even with this setup, our model can achieve good results while predicting citations since its tokenizer has access to all citation names as additional tokens. So, while our approach has a performance bottleneck due to the addition of citation tokens, it can operate more flexibly compared to past works.

Interestingly, our model can predict citations by only using context, unlike past works that required a multitude of additional information like title, abstract, journal/conference information, etc. Our model can still predict while having similar success rates to past works.

Another interesting aspect of our approach stems from the connection between citations and contexts. The citations are simply author names and years. However, after training, our model can learn to associate these citations with contexts. From the perspective of a language model, author names and years should have a very minimal connection to the words in the contexts semantically. However, our model can still learn their connections to a certain degree and become capable of predicting with high success rates. The only difference here is that vocabulary has been extended to contain citations as they are unique identifiers. Even if we formed citations using random letters and numbers instead of using author names and years, we believe our model would still be able to predict with similar rates of success.

Instead of adding new vocabulary items, we have also considered masking the citation spans inside a context. Then, we would be able to fine-tune a model like SpanBERT (Joshi et al. 2020) to predict the citations for a given group of consecutive masks. However, it's not straightforward to decide the number of masks in each citation context as it varies. For example, the citation "Stalzer, 2017" is tokenized as ['ĠSt', 'al', 'zer', ',', 'Ġ2017'] with RoBERTa-base tokenizer. Meanwhile, "Mcmahan and Orabona, 2014" is tokenized as ['ĠMc', 'm', 'ahan', 'Ġand', 'ĠOr', 'ab', 'ona', ',', 'Ġ2014']. They have five and nine tokens, respectively.

While manually providing the necessary number of masks for each context is an option, it severely reduces the flexibility of our model. Thus, it is not a good solution. Conversely, the number of masks may also serve as an additional feature for the prediction, assuming we can find a way to accurately predict the number of masks for contexts. However, even if we implement a mechanism that predicts the number of masks beforehand, the model may learn its predictions with bias due to the varying number of masks. For example, citations with longer names may have a higher chance of being predicted for longer sequences of masks while ignoring the actual contents of the context and its relevance to the citation.

Since the citations are tokenized in these formats, none of them correspond to a single token. Due to this reason, it is not possible for our model to predict a complete citation for a single token. Additionally, we have considered using the T5 model for this task. The T5 model is shown to be capable of predicting multiple tokens for a single mask. However, we were not able to leverage this ability of T5 due to its complexity and time

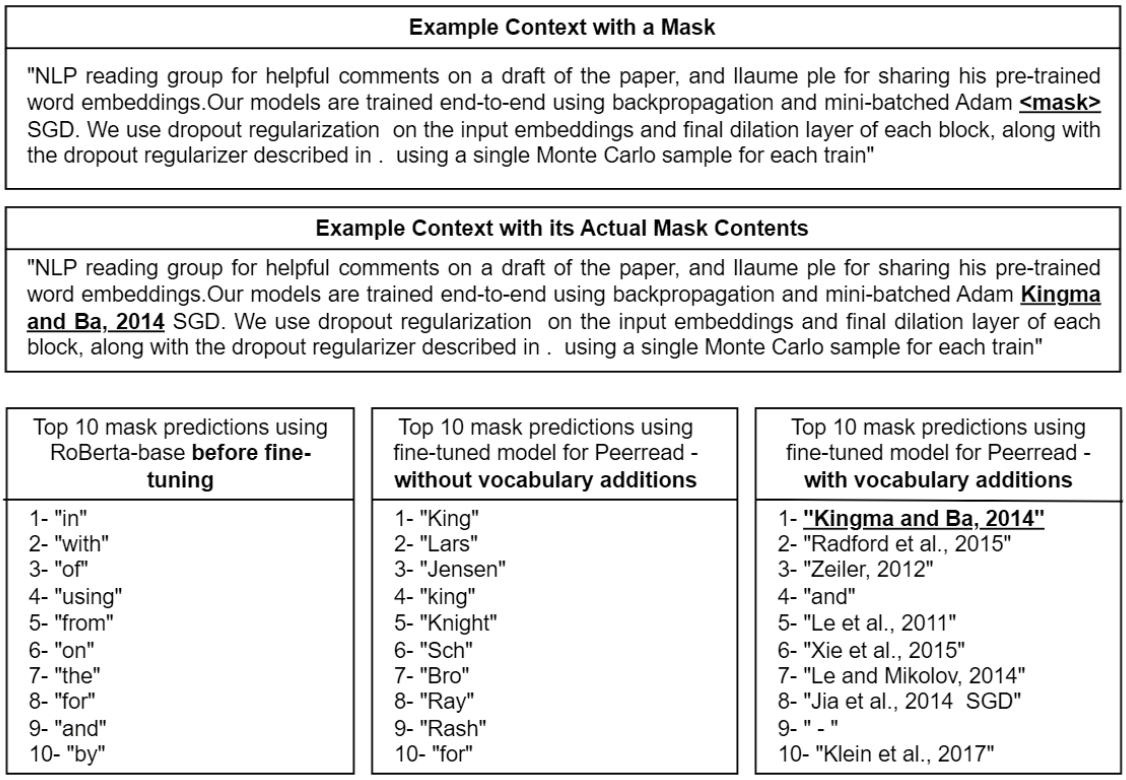

Figure 3.4. Example scenario for citation prediction under different conditions.

constraints. We leave this approach as an item of future work.

To solve this problem, we tried to add every single citation to the tokenizer's vocabulary. For example, we manually added ”Gribkoff et al., 2014” to the tokenizer, and the model became capable of predicting it as a single token corresponding to a single mask. Since we need to perform this tokenizer addition operation incrementally, it takes a considerable amount of time to add every citation of a dataset to the tokenizer.

In our experiments, the base tokenizer we used came from the RoBERTa-Base model. All citations have been made sure to be in the forms given in Figure 3.1. In other words, we have treated every citation like a single token by adding them to the vocabulary. In this configuration, our model is now capable of predicting citations as a single token for a single mask.

In Figure 3.4, we have shown an example scenario using a masked context. The ground truth of the mask in this example is ”Kingma and Ba, 2014”. Afterward, we ask our model to provide us with the top 10 predictions for the masked citation under three

different conditions. In the first condition, we add the model's tokenizer citations as a single token, but we do not perform any training on the RoBERTa-Base model. As can be seen from the top 10 predictions, the model fails to predict any token that is relevant for the citations in this scenario.

In the second condition, we do not make any additions to the vocabulary of the tokenizer and directly use RoBERTa-Base's vocabulary. However, we perform training and then observe the top 10 predictions. We notice predictions like "King" or "king" that are relevant to the name "Kingma" from the citation. However, the model fails to predict the actual citation token since we did not make any additions to its vocabulary.

In the last condition, we perform the additions to the vocabulary and perform training. As a result, we observed the ground truth citation on the first prediction, which was in line with our goals. Also, there are some additional citation suggestions in the remaining 9 predictions as well. We can assume these are also relevant to the contents of the given masked context too. So, this model can also operate as a suggestion mechanism as well.

3.2. Preprocessing of the Datasets

We conduct our experiments on the existing citation prediction datasets that are ACL-200, FullTextPeerRead, RefSeer (Medić and Snajder 2020), and Arxiv (Gu, Gao, and Hahnloser 2022). Table 3.1 presents the statistics of these datasets.

Table 3.1. Statistics of the datasets for citation prediction.

Dataset Name	Train Size	Validation Size	Test Size	Number of Papers	Publication Years
ACL-200	30390	9381	9585	19776	2009 - 2015
FullTextPeerRead	9363	492	6184	4837	2007 - 2017
RefSeer	3521582	124911	126593	624957	- 2014
Arxiv	2988030	112779	104401	1661201	1991 - 2020

To use these datasets for our approach, we need to preprocess them in a certain way. Our model requires the contexts that possess the target citations from these datasets.

By default, these datasets include a "TARGETCIT" marker at the location of citations within a context. They also contain other information such as complete author names, publishing information, etc. We only need the citation contexts with and without masks. However, these datasets generally do not have these contexts directly in this manner. So, we performed a preprocessing step to acquire both versions of contexts alongside the target citations. Additionally, we have directly used "<mask>" tokens as masks instead of "TARGETCIT" tokens. This is a necessary modification because our chosen pre-trained model, RoBERTa-base, requires the mask tokens in the former format. We also mask the other appearances of the duplicate citations inside the same context.

After the preprocessing, we split the benchmark datasets into training and evaluation partitions using a conventional ratio of $8:2$. The original datasets consisted of train, validation, and test partitions. However, our approach involves further pre-training on the train partition and evaluating on the evaluation partition. Consequently, we only require two partitions instead of three. The sizes of our training and evaluation partitions are also shown in Table 3.1.

During the preprocessing, we eliminated certain problematic citation contexts from some of the original datasets, and we had to decrease the total sizes of the datasets as needed. The final statistics of our preprocessed datasets can be found in Table 3.2.

We also did not need extra information like citation count, publication details, etc., inside the datasets. For this reason, we did not include them in our preprocessed datasets. Our preprocessed datasets only have the following information inside them: "Masked context", "Unmasked context" and "Citation Token Target".

Table 3.2. Statistics of the preprocessed datasets.

Preprocessed Dataset Name	ACL-200	FullTextPeerRead	RefSeer - All	Arxiv - All
Number of local contexts	63316	16669	3739189	3205210
Size of the training split	50652	13335	2991351	2564168
Size of the evaluation split	12664	3334	747838	641042
Number of eliminated local contexts	452	0	39577	0
Number of Vocabulary Additions	5259	2043	351896	368284

3.3. Challenges of the Datasets

We encountered some issues during the processing of the datasets. ACL-200 and RefSeer include some local contexts with specific problems. One such problem is the conflict in author names between the context and the ground-truth citation. For example, the “Petrović et al., 2010” citation was incorrectly written as “Petrovic et al., 2010” in the target citation column of ACL-200. Another problem is incorrect ordering in citations with two authors. For example, the citation ”Rivera and Zeinalian, 2016” is the name provided in local contexts, but the name found in paper data, which contains titles and abstracts, is ”Zeinalian and Rivera, 2016”. Another potential issue happens when the dataset points to an incorrect reference paper for a citation context. Furthermore, there are instances of empty author names in some contexts. We removed all these cases from the two aforementioned datasets to ensure consistency.

We also needed to ensure that each context had its ”<mask>” token in its middle position after tokenization. We believe that our model will learn more effectively when the context window around the mask token is roughly the same length for both of its sides.

Another critical aspect of the preprocessing was determining the correct length for citation contexts. An exploratory analysis of context lengths shows that ACL-200’s contexts are significantly longer than the other datasets. After tokenization, we observed that a limit of 200 – 400 tokens was optimal for the datasets. This limit allows sufficiently long contexts without a need for excessive amounts of padding tokens. Only ACL-200 has 607 contexts that exceed the 400 limit. This number is quite small compared to the whole number of contexts in the dataset. Moreover, considering the rest of those contexts, the amount of discarded tokens is negligible. Table 3.3 shows the chosen maximum token limits for the datasets.

3.4. Evaluation Metrics

To measure the success of our model, we use four different metrics: Hits@10, Recall@10, Exact match, and MRR. The trained model of each dataset is evaluated with these metrics. Each dataset’s test set comprises 20% of the preprocessed datasets.

Hits@10 metric is calculated by checking the top 10 predictions of the model for

Table 3.3. Maximum token limits for the preprocessed datasets.

Dataset Name	Maximum Token Limit
ACL-200	400
FullTextPeerRead	400
Refseer – All	200
Arxiv - All	300

a given masked input context. The model, as previously mentioned, predicts target tokens in the form of complete citations. If one of these top 10 predicted citations is the same as the actual target citation, that counts as a successful prediction for the given context.

Recall@10 metric is calculated similarly to hits@10. However, it can check if predictions belong to more than one actual target. Our datasets have been preprocessed in such a way that there is only one actual target. So, this causes recall@10 to be the same as hits@10 in our experiments. In all our experiments, these two metrics have always resulted in the same value.

The exact match (EM) is calculated by checking whether the first prediction of our model is the same as the target citation. This metric can also be considered as the accuracy of our model since there is only one actual target citation for each context.

MRR (Mean Reciprocal Rank) is a metric that considers the position of the first relevant item in a ranked top-k prediction list. In our experiments, we used k as 10 and checked the model's top 10 predictions. Thus, we considered the position of the correctly predicted citation for each context and calculated the MRR value using its formula. In Equation 3.1, U corresponds to the total number of contexts in the dataset, and i corresponds to the position of the first relevant item for context u in the top-K results.

$$MRR = \frac{1}{U} \sum_{u=1}^{U} \frac{1}{rank_i} \tag{3.1}$$

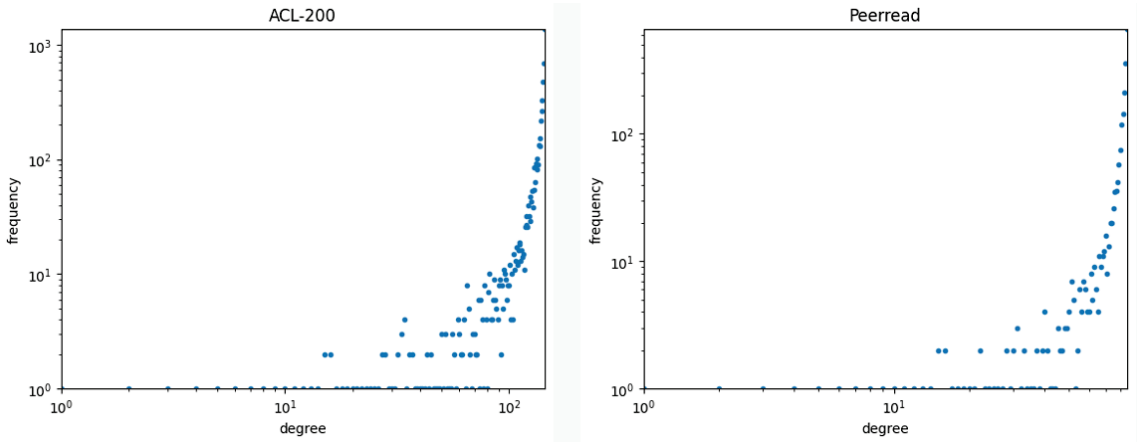

Figure 3.5. ACL-200 and PeerRead - Log-log graphs of contexts per citation counts.

3.5. Factors on Performance

We have also tried to understand how certain attributes of the datasets can improve the learning process. In simple terms, our model tries to predict a citation token (made up of name(s) and a year) from at least one context that includes it. As one might expect, the distribution of these contexts for each citation token is not balanced. Some citation tokens have hundreds of contexts, while others have only a single context. Naturally, the learning process becomes more complicated when a citation token has been used inside only a single context. Thus, we tried to observe the overall distribution of contexts per citation to better understand the learning capabilities of our model.

This section analyzes the impact of datasets' characteristics on the overall performance. The sparsity/density of the citation graph may influence the learning performance. The distribution of the number of citation contexts per citation token gives insights into the local connectivity pattern of the citation graph, and local patterns may induce global behavior. In terms of learning, models are expected to be more accurate in predicting the citation tokens that appear in a large number of contexts. Thus, to better understand the learning capabilities of models, Table 3.4 summarizes the overall distribution of contexts per citation. The table reports citation tokens' mean, median, standard deviation, minimum, and maximum cited times. It also includes the number of citation tokens with exactly one occurrence. These statistics are calculated according to each citation's corresponding context counts. For example, the ACL-200 dataset possesses a citation that has

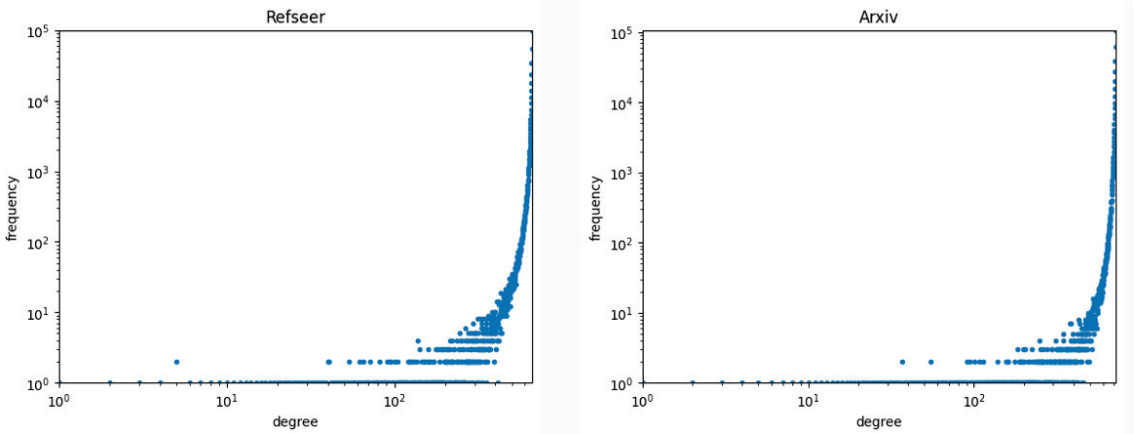

Figure 3.6. RefSeer-All and Arxiv-All - Log-log graphs of contexts per citation counts.

been cited in 829 contexts in the ACL-200 dataset. However, the mean and median of ACL-200 are 113.69 and 73, respectively. So, the maximum count of 829 contexts is an outlier compared to the rest of the dataset.

We draw log scale plots for each citation graph. In these graphs, the number of citations corresponds to a point on the x-axis. The y-axis corresponds to its number of occurrences, and they are sorted based on their number of occurrences in descending order. Figure 3.5 and Figure 3.6 shows the log-log graphs of the datasets.

As the graphs show, the datasets contain many underutilized citations that cause difficulties in the learning process. The past works, including Hatten (Gu, Gao, and Hahnloser 2022), had trouble correctly predicting citations due to the skewed nature of the datasets. Our model also faces problems due to this reason. However, we reached significantly better results thanks to our approach that leverages citations as a single token for mask-filling tasks. Especially, our global models were mostly capable of overcoming the skewed nature of the datasets and produced even higher results than our base models. Most of the citations that our models have failed to predict naturally belong to the under-utilized citations. The approaches of the past works' models most likely fail to predict these underutilized citations alongside many other properly utilized citations. This could explain the difference between the success metrics of our models and past works' models.

The log-log graphs show that the citations with the higher number of contexts correspond to dots in higher degree values. The frequency value of dots corresponds to the number of contexts that belong to a citation. Meanwhile, the degree values of dots

correspond to the number of citations that have the same frequency value, i.e., the citations that have the same number of contexts citing them. For example, the ACL-200 dataset has a total of 1385 citations with exactly one context. In the log-log graph of ACL-200, notice that the dot corresponding to the 1385 frequency value is at the top right corner of the graph. Similarly, the Peerread dataset has 669 as its total number of citations with one context, and the dot corresponding to it can be seen on the top right corner of its graph.

From the log-log graphs, we can observe the effect of the number of citations for each dataset. Since Peerread has fewer citation items compared to ACL-200, we can see fewer dots on the Peerread graph. Similarly, the graphs of the larger datasets have significantly more dots than smaller datasets. Another important aspect of the graphs is where the dots with higher frequency values have the most density. We observe that the Peerread dataset has the most balanced distribution among the benchmark datasets. While the ACL-200 dataset has more density at the higher degree values compared to Peerread, it still has a more balanced distribution compared to larger datasets. The dots in the higher degrees of the larger datasets are grouped in a very dense line-like cluster. We believe this denseness can lead to a negative impact on the success of the model on the larger datasets. If a log-log has a more balanced distribution on its area with low frequency and high degree, we believe it will achieve better results with our models. As can be seen from the graph of the Refseer dataset, there is a very high density area with low frequency ($9 - 12$) and high degree ($400 - 500$). Similar observations can also be made for the Arxiv dataset.

We can observe additional key points from these graphs and statistics. For example, there is a very large difference between citation item totals of smaller datasets and larger datasets. Also, the number of citations with exactly one context in larger datasets is very high as well. Since these individual examples are the only possible way to learn their corresponding citations, it becomes very unlikely to learn these citations without more examples that belong to them. Thus, it is very difficult for the model to correctly predict the citation of an example like this during evaluation. It is possible to avoid this issue if the benchmark datasets were better curated. Lastly, the large number of examples on the x-axis of the log-log graphs corresponds to the number of citations with exactly one context. So, the aforementioned issue can be observed in Refseer and Arxiv's log-log graphs.

Table 3.4. Context per citation count statistics of the datasets.

Dataset Name	ACL-200	FullTextPeerRead	RefSeer - All	Arxiv - All
Total # of citations	5259	2043	351896	368284
# of citations with exactly one context	1385	669	100765	107667
Minimum	1	1	1	1
Maximum	829	892	4030	10437
Mean	113.69	77.51	402.07	615.53
Median	73	44.5	331.5	361.5
Standard Deviation	141.62	122.22	349.51	860.93

3.6. Effects of Sampling Techniques on the Datasets

To test the effect of the datasets' nature and size on the learning performance, we applied different sampling techniques to generate samples from Arxiv and RefSeer. Initially, we constructed samples of 200000 and 300000 for Refseer and Arxiv and named these new samples Refseer-200k and Arxiv-300k, respectively.

Table 3.5. Statistics of the sampled datasets.

Dataset Name	Refseer-200k	Arxiv-300k-random	Arxiv-300k-neg-sampling
Dataset size	200000	300000	300000
Maximum Token Limit	200	300	300
Total number of citations	66193	103796	3728

Arxiv-300k is more heavy-tailed than Refseer-200k, implying that there will be fewer learning signals due to references cited only once. Generally, this causes a reduction in the success of citation predictions.

After observing this issue, we decided to find the effect of the sampling technique on the Arxiv-300k dataset. We aim to view the success rate of our model when it has been trained on a dataset sampled under certain conditions.

Our initial Arxiv-300k dataset has been renamed to Arxiv-300k-random since it is made with uniform sampling. Another sampled dataset we have created is Arxiv-300k-neg-sampling. This dataset is sampled using the negative sampling algorithm so that the

overall distribution of the original Arxiv dataset can be preserved. Negative sampling is applied by sampling with the unigram distribution $U(w)$ raised to the $3/4$ power so that the power makes less frequent citations be sampled more often.

Table 3.6. Contexts per citation counts statistics of the sampled datasets.

Dataset Name	Refseer-200k	Arxiv-300k-random	Arxiv-300k-neg-sampling
Minimum	1	1	1
Maximum	299	1087	10370
Mean	56.77	128.20	586.16
Median	49.5	89.5	205.5
Std	45.58	136.05	1073.78

The statistics of these sampled datasets are shown in Table 3.5. Additionally, we have demonstrated the contexts per citation count statistics in Table 3.6. The total number of citations in the sampled dataset is less than those of the original datasets as a consequence of the sampling process. Maximum token limits are identical to their original datasets since the maximum lengths of the context inputs do not change during sampling. After sampling, the contexts per citation counts statistics change significantly due to the smaller size of the datasets. The minimum and maximum values correspond to the citations with minimum and maximum number of contexts. For example, in the original Refseer dataset, there was a citation that has been referenced in 4030 contexts. However, in the Refseer-200k dataset, this citation was not selected during sampling, and the new maximum context count of a citation is 299. Additionally, the mean, median, and standard deviation values of sampled datasets have also changed for reasons similar to those before.

After our initial observations, we have noticed that the total number of citation items in uniformly sampled datasets (Refseer-200k and Arxiv-300k-random) tends to be larger. Notice that these large numbers are closer to the numbers of the original Refseer and Arxiv datasets. Meanwhile, the dataset created using the negative sampling approach has significantly fewer citation items. These numbers are closer to the values of the smaller datasets, which are ACL-200 and Peerread. Additionally, we have observed a very large difference in the contexts per citation counts statistics between uniformly sampled datasets

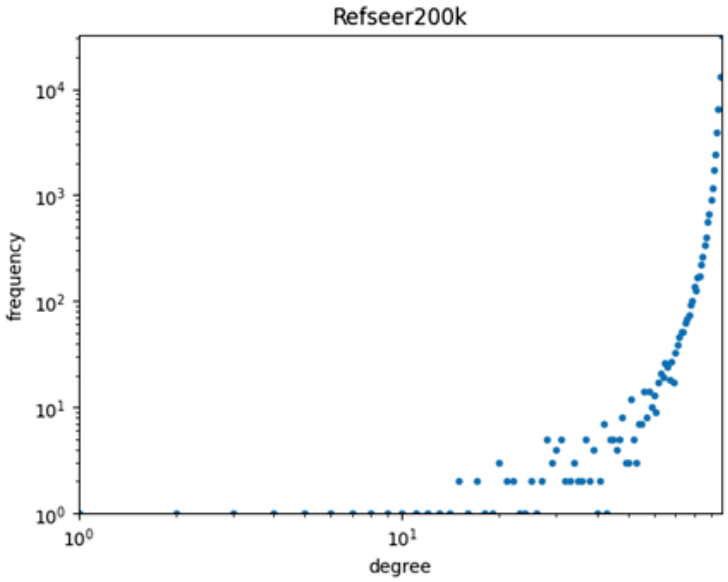

Figure 3.7. Refseer-200k - Log-log graph of contexts per citation counts.

and the neg-sampling datasets. There is a significant gap in their maximum, mean, median, and standard deviation statistics. We believe these statistics have a significant impact on the success of our model.

We have also drawn the log-log graphs of these datasets in Figure 3.7 and Figure 3.8 to better illustrate their distribution compared to their original versions. Like the four main datasets, these log-log graphs show the distribution of the contexts per citation. Notice that Arxiv-300k-random's graph is very similar to the original Arxiv graph. Similar observation can be made for the Refseer-200k dataset as well since it also uses uniform sampling. However, the Arxiv-300k-neg-sampling dataset has a more balanced distribution compared to uniform sampled ones and the original datasets. This balanced distribution can be observed in graph's area with low frequency and high degree. We believe this balanced distribution has a positive impact on the success of the models on the negative sampled datasets. Additionally, we have observed that the uniformly sampled datasets tend to have more contexts in the lower degrees, while the negative sampled dataset has more contexts in the higher degrees.

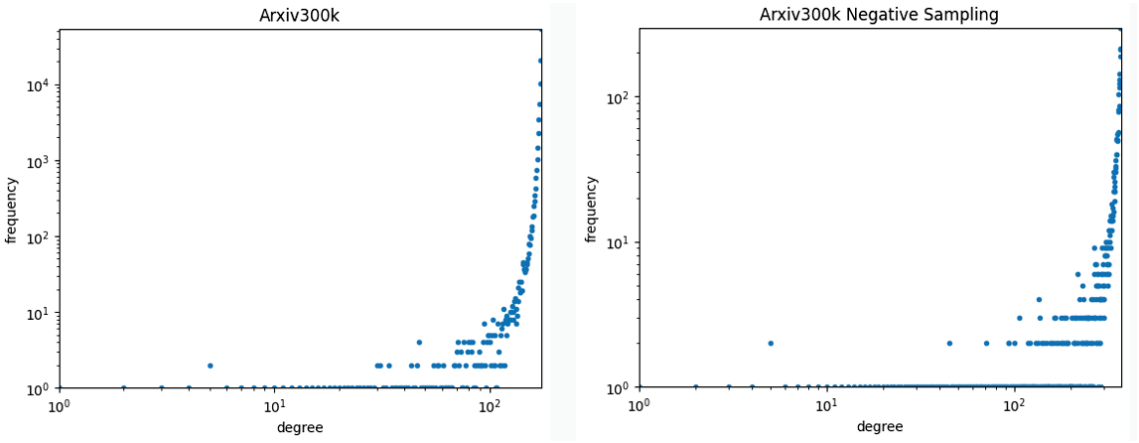

Figure 3.8. Arxiv-300k - Random and Negative Sampling - Log-log graphs of contexts per citation counts.

3.6.1. Details of Sampling Techniques

In this section, we provide the sampling algorithms of the three techniques we have used while acquiring our sampled datasets. In Algorithm 3.1, we show the pseudo-code of our random sampling algorithm. This algorithm has been used for both Refseer-200k and Arxiv-300k-random datasets. In Algorithm 3.2, we have provided the pseudo-code of our negative sampling approach, which is used for the Arxiv-300k-neg-sampling dataset.

The random sampling algorithm directly uniform samples from all indices of the context list and pre-processes them to acquire the dataset format we use for our models. In line 1 of Algorithm 3.1, the three necessary inputs for the random sampling procedure are shown. In line 2, we acquire a list of all indices of C. Then, we perform uniform sampling on this list to acquire N samples in line 3. Lines 4, 5, and 6 consist of creating three empty lists to store the necessary information for our dataset. Afterward, we begin looping over all contexts and decide whether to add each context to our dataset. In line 8, we check if the current context's index value is inside the selected indices in line 3. If it is not inside the selected indices, we will skip that context. In line 10, we find the ground truth context from the paper information dictionary P. We prepare the unmasked and masked contexts in lines 11 and 12. Then, we append these values to their corresponding lists in lines 13, 14, and 15. If we have reached the target sample size of N, we stop the loop early in lines 16 and 17. Finally, we merge the three lists we have acquired into a single dataset D in the remaining two lines.

Algorithm 3.1 Random sampling algorithm for Refseer-200k and Arxiv-300k-random.

```
1: procedure RANDOM_SAMPLE_FROM_DATASET(C: a list of all contexts from a dataset,
   P: a dictionary containing all papers and their information, N: sample size)
2:     C_idx ← list of all indices of C
3:     S ← Randomly sample N values from C_idx (uniform sampling)
4:     G ← an empty list for ground truth contexts
5:     M ← an empty list for masked token contexts
6:     T ← an empty list for ground truth citation targets
7:     for every element i in C do
8:         if index of i is not in S then
9:             continue
10:        t ← target citation token of i acquired from P
11:        g ← ground truth context from i
12:        m ← context from g with its citation replaced with <mask>
13:        Append t to T
14:        Append g to G
15:        Append m to M
16:        if total number of elements in G is equal to N then
17:            break
18:    Merge the lists of T, G, and M to acquire the sampled D dataset
19:    return D
```

The negative sampling algorithm uses alpha as 3/4 to increase the chances of selecting citations with fewer contexts. Then, we add these selected citations' contexts and relevant data until we reach the required sample size. In line 1 of Algorithm 3.2, we use an additional input compared to Algorithm 3.1, which is a dictionary that consists of appearance counts of each citation name, i.e., how many contexts each citation corresponds to. We perform the operation in lines 2 and 3 so that we can acquire the probability distribution of each citation name. In line 5, we use the $alpha$ value to add noise to the probability distribution. By summing and dividing the noise distribution, we normalize it in lines 6 and 7. In line 8, we initialize an empty list for the citation names we will choose during the negative sampling process. We keep the total counts of selected contexts in line 9 so that we can stop when the sample size N is reached in lines 11 and 12. Afterward, we perform sampling according to the probability distribution in $normProbDistOfCites$ to select a citation name in line 13. We append it to our list that contains the selected citation names in line 14. Then, we add the number of contexts corresponding to the selected citation in line 15. In line 16, we remove the selected citation from the distribution dictionary so that it cannot be selected again. Starting from line 17, we perform the identical steps of Algorithm 3.1, except in line 22. We decide whether to add the contexts

of a citation to the dataset according to the *chosenCiteNames* list.

The global version of these sampled datasets was also acquired using the same methods. Their only difference is the addition of titles and abstracts to the masked and ground truth contexts.

3.7. Global Technique: Learning Citation Representations with Global Info

To further improve the results of our approach, we propose another technique alongside our base approach. Initially, our base approach was being trained only using the context windows taken from scientific papers. The masked citations we try to predict have been made sure to be in the middle of these contexts. We believe that our model should be able to benefit from certain additional information from the scientific papers that originally contained the context windows. This additional information will be mainly the titles and abstracts of scientific papers.

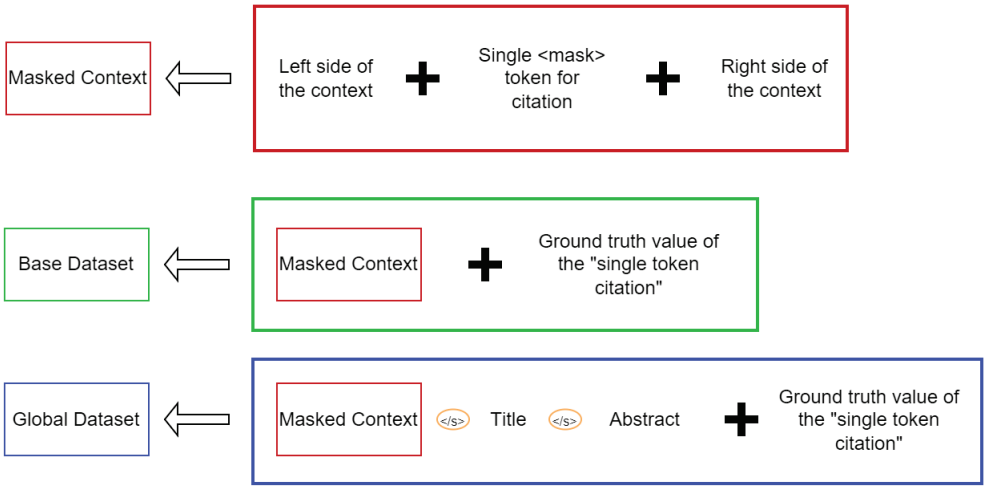

Figure 3.9. Overall look at the structure of the datasets.

We refer to our improved models as “global” models since they contain titles and abstracts along with the contexts. To properly use titles and abstracts in our models, we recreated our datasets inspired by the approach of REALM paper (Guu et al. 2020). Our global models are further pre-trained using contexts, titles, and abstracts separated

by special separator tokens. Specifically, we used the "</s>" token designated by the pre-trained RoBERTa-base model. During the training process, our global model has become capable of learning how to predict a mask in the context while also considering the additional information from the title and abstract. Figure 3.9 shows an overall look at at the structure of the base and global datasets.

We preprocessed the four main datasets according to these guidelines. Due to having a limited token size after adding titles and abstracts, we had to limit our context windows' token numbers. For ACL-200 and Peerread datasets, we have created two alternative versions with 50 token context windows and 200 token context windows. Each context window has been made sure to have the masked citation in its middle point as well. Afterward, we intuitively selected a maximum token limit for the addition of the titles and abstracts.

Table 3.7. Maximum token limits for the preprocessed global datasets.

Dataset Name	Context Token Limit	Total Token Limit
ACL-200-context-50	50	500
ACL-200-context-200	200	500
Peerread-context-50	50	500
Peerread-context-200	200	500
Refseer – All	50	400
Arxiv - All	50	400
Refseer-200k	50	400
Arxiv-300k-random	50	400
Arxiv-300k-neg-sampling	50	400

We have observed that a limit of 500 tokens is appropriate for these two datasets. The remaining two datasets' token limits have been selected as 50 for contexts and 400 for total. The token limits for the datasets have been shown in Table 3.7. Additionally, we have created the global versions of Refseer-200k, Arxiv-300k-random, and Arxiv-300k-neg-sampling datasets as well. These sampled datasets have the same token limits as their complete versions. Other specifics of all datasets have remained the same since only adding titles and abstracts does not affect statistics like mean, median, etc.

Algorithm 3.2 Negative sampling algorithm for Arxiv-300k-neg-sampling.

procedure NEGATIVE_SAMPLE_FROM_DATASET(C: a list of all contexts from a dataset, P: a dictionary containing all papers and their information, A: a dictionary containing all appearance counts of each citation, N: sample size)
    $totalCiteCount \leftarrow$ total of each value in A
    $probDistOfCites \leftarrow$ each value in A divided by $totalCiteCount$
    $alpha \leftarrow 3/4$
    $noiseDist \leftarrow$ each value of $probDistOfCites$ to the power of $alpha$
    $noiseSum \leftarrow$ the sum of each value in $noiseDist$
    $normProbDistOfCites \leftarrow$ each value in $noiseDist$ divided by $noiseSum$
    $chosenCiteNames \leftarrow$ empty list
    $totalContextCount \leftarrow 0$
    while True **do**
        if $totalContextCount > N$ **then**
            break
        $randomlySampledCiteName \leftarrow$ randomly sample a citation name from the keys of $normProbDistOfCites$ using the probability distribution in its values
        Append $randomlySampledCiteName$ to $chosenCiteNames$
        $totalContextCount \leftarrow totalContextCount+$ the number of contexts belonging to $randomlySampledCiteName$
        Remove $randomlySampledCiteName$ from $normProbDistOfCites$ since it has been selected
    $G \leftarrow$ an empty list for ground truth contexts
    $M \leftarrow$ an empty list for masked token contexts
    $T \leftarrow$ an empty list for ground truth citation targets
    for every element i in C **do**
        $t \leftarrow$ target citation token of i acquired from P
        if t is not in $chosenCiteNames$ **then**
            continue
        $g \leftarrow$ ground truth context from i
        $m \leftarrow$ context from g with its citation replaced with <mask>
        Append t to T
        Append g to G
        Append m to M
        if total number of elements in G is equal to N **then**
            break
    Merge the lists of T, G, and M to acquire the sampled D dataset
    return D

CHAPTER 4

EXPERIMENTS AND RESULTS

4.1. Experiments

The experiments for the smaller datasets (ACL-200 and Peerread) have been performed on a device with NVIDIA Titan V GPU. The remaining larger datasets (Arxiv and Refseer) have been trained on two devices with NVIDIA V100 and NVIDIA RTX6000 GPUs. The models for these datasets have been further pre-trained for both their base and global versions. Additionally, our sampled datasets have also been further pre-trained on a device with NVIDIA Titan V GPU.

The number of epochs for the training has been decided after observing each model's losses and perplexity values. While training for smaller datasets can be performed in a relatively short time, larger datasets require considerably longer training time. We were able to observe the results of smaller datasets with different numbers of epochs like 30, 100, etc. However, their training could only be performed up to a range of 5 – 10 epochs for the larger datasets due to their large size.

Thanks to the smaller size of ACL-200 and Peerread datasets, they took 25 hours and 6 hours, respectively, during their training process. Meanwhile, their global version lasted 32 hours and 8 hours, respectively, for their training process.

Due to our limited resources, the larger datasets generally required a training time ranging from 2 weeks to 4 months for epoch numbers like 5 and 10. Their global versions increased these training times up to 6 – 7 weeks as well. However, training times vary a negligible amount between Refseer and Arxiv since their overall sizes are approximately similar to each other. With our limited hardware resources, if we tried to train the larger datasets for 30 or 100 epochs, these training operations would potentially require more than 6 months to be completed. Due to these reasons, we limited the total number of epochs between 5 and 10 in general.

The training times of our sampled datasets are less than their complete versions. Refseer-200k dataset's base and global training steps last for 65 and 112 hours, respectively.

Meanwhile, Arxiv-300k possesses similar training times for its two different versions. Their base and global versions require training times of approximately 130 hours and 180 hours, respectively.

After completing their training, they have been evaluated on their corresponding evaluation sets. Our evaluation process also takes a considerable amount of time since generating the top 10 predictions for each example is a resource-intensive task. Especially with our limited hardware resources, acquiring the larger datasets' evaluation results may take up to $3 - 5$ days since larger datasets' evaluation sets are also extensive. However, smaller datasets naturally require less time for evaluation, which is roughly between 30 minutes and 2 hours.

The issue of slow evaluation time for larger datasets is not a problem exclusive to our work. The authors of Hatten (Gu, Gao, and Hahnloser 2022) have also pointed out long evaluation times as an issue. They have reported using only a smaller subsection of each large dataset for their evaluation steps to speed up evaluation times. We still tried to use complete evaluation datasets for our evaluations to keep our results as accurate as possible.

4.2. Results

Table 4.1. The result metrics of the base datasets and their sampled versions.

Dataset Name	Number of Epochs	Perplexity	Hits@10	Recall@10	Accuracy (Exact Match)	MRR
ACL-200	100	1.02	73.02%	73.02%	48.12%	0.568
Peerread	100	1.04	64.40%	64.40%	41.93%	0.496
Refseer	3	1.19	0.52%	0.52%	0%	0.001
Arxiv	4	1.14	4.16%	4.16%	0%	0.013
Refseer-200k	100	1.09	35.30%	35.30%	22.68%	0.271
Arxiv-300k-random	100	1.04	14.23%	14.23%	7.23%	0.095
Arxiv-300k-neg-sampling	100	1.02	72.26%	72.26%	48.08%	0.564

In Table 4.1 and Table 4.2, the evaluation metric results for the base and global models have been shown. Alongside the four main datasets, the training results of the sampled datasets have also been provided. We generally aimed for 100 epochs of training in our experiments. The reason our large datasets' models (shortly referred to as large models) have only $3 - 4$ epochs is due to our limited time and hardware resources.

Additionally, our models failed to achieve adequate results on these datasets. Even with our limited resources, their experiments went up to 10 epochs. However, we have seen that epoch 3 or 4 has the best results in these experiments after further analysis. So, we have specifically shown the results of these epochs in the Table 4.1 and Table 4.2. The reasons behind this issue are thoroughly explained in the Discussion chapter.

Table 4.2. The result metrics of the global datasets and their sampled versions.

Dataset Name	Number of Epochs	Perplexity	Hits@10	Recall@10	Accuracy (Exact Match)	MRR
ACL-200-global-context-50	100	1.01	96.93%	96.93%	95.57%	0.962
ACL-200-global-context-200	100	1.06	97.13%	97.13%	95.53%	0.963
Peerread-global-context-50	100	1.11	93.76%	93.76%	93.67%	0.937
Peerread-global-context-200	100	2.00	93.27%	93.27%	92.80%	0.930
Refseer-global	3	1.21	0%	0%	0%	0
Arxiv-global	3	1.11	11.85%	11.85%	0%	0.026
Refseer-200-global	100	1.09	62.33%	62.33%	55.96%	0.580
Arxiv-300k-random-global	100	1.06	42.04%	42.04%	37.04%	0.387
Arxiv-300k-neg-sampling-global	100	1.05	98.28%	98.28%	98.05%	0.981

The comparisons between state-of-the-art works and our approaches can be seen in Table 4.3. For the Hatten approach, its most successful version with 2000 pre-fetched recommendation candidates has been selected for comparison. Our approach does not perform a pre-fetching step and directly returns predictions.

We also performed additional experiments to see the effect of different numbers of epochs for some of the smaller and sampled datasets. The perplexity, hits@10, and recall@10 result metrics of these datasets trained for 30 and 60 epochs have been shown in Table 4.4. The datasets selected for this experiment have been selected to showcase the effect of epoch number under many varied circumstances even if some datasets have not been used for this experiment.

4.3. Ablation Study

Since we have observed a clear improvement in the global strategy, we have decided to perform an ablation study to discover the effects of the title and abstract on the success of our model. For this purpose, we have removed the contexts from the global strategy.

Table 4.3. The comparisons of the results with past works. The results of Hatten have been taken from its 2000 pre-fetched candidate version, which is the most successful one.

Dataset	Approach	Recall@10
ACL-200	Medić and Snajder (2020)	0.716
	HAtten	0.633
	Our Base Approach	0.730
	Our Global Approach	0.969
Peerread	Medić and Snajder (2020)	-
	HAtten	0.757
	Our Base Approach	0.644
	Our Global Approach	0.938
Refseer	Medić and Snajder (2020)	0.534
	HAtten	0.454
	Our Base Approach	0.005
	Our Global Approach	0
Arxiv	Medić and Snajder (2020)	-
	HAtten	0.439
	Our Base Approach	0.042
	Our Global Approach	0.118

Essentially, each entry in our datasets would be made up of "citation token + title + abstract" instead of "context + title + abstract". We performed the experiments for this task on ACL-200 and Peerread datasets since they take less time to train compared to others. After 100 epochs of training were conducted, the results shown in Table 4.5 were obtained.

From the results, we can see that the removal of context slightly reduces the performance of our model. However, this does not mean that the contexts are unimportant because the models that have been trained only with contexts are still relatively successful compared to other works in this field. Contexts can still be helpful for the learning capabilities of our model. Also, the higher rate of success in our global models has shown that adding citation tokens to a model can dramatically improve its results. This improvement can be observed in the ablation study models as well. So, the main contribution of our models comes from the addition of citation tokens to the models' vocabularies.

Table 4.4. The effect of the number of epochs on the results of some of our datasets.

Dataset Name	Number of Epochs	Perplexity	Hits@10	Recall@10
Refseer-200k-global	30	1.07	63.25%	63.25%
Refseer-200k-global	60	1.08	61.50%	61.50%
Refseer-200k	30	1.08	29.54%	29.54%
Refseer-200k	60	1.09	30.58%	30.58%
Arxiv300k-random-global	30	1.05	45.32%	45.32%
Arxiv300k-random-global	60	1.06	43.56%	43.56%
Arxiv300k-neg-sampling-global	30	1.03	98.30%	98.30%
Arxiv300k-neg-sampling-global	60	1.04	98.27%	98.27%
Peerread-global-context-50	30	1.00	93.82%	93.82%
Peerread-global-context-50	60	1.00	93.79%	93.79%

Table 4.5. Ablation study results on ACL-200 and Peerread.

Dataset Name	ACL-200 - ablation	Peerread - ablation
Perplexity	1.00	1.00
Hits@10	96.84%	93.16%
Recall@10	96.84%	93.16%
Accuracy (Exact Match)	95.12%	93.13%
MRR	0.959	0.932

4.4. Qualitative Analysis on Prompting Large Language Models

We have performed experiments on large language models to see how they perform on our task. By prompting large language models, we aim to analyze how they respond to the needs of our task. We chose the "Llama-2-70b-chat" model to perform our prompting trials on. At each prompt, we provided a list of citation tokens from our datasets alongside some examples of masked contexts and their ground truth mask values. Afterward, we asked the model to fill the mask in another context by selecting a citation from the list that had been provided initially.

Due to the limited nature of the chat windows of large language models, we had to limit the total number of citations by 200. In this limited setting, we have performed a few

trials to view how a large language model responds to our prompts. Out of the four trials, only one managed to correctly predict the value of the mask. The two of them selected an incorrect citation from the list, while the last one filled the mask with a citation outside the given list.

We have also performed additional trials that can be compared to our global approach. The main difference between these trials is that each citation in the given list is accompanied by its title and abstract. We had to limit the number of citations to 30 since the addition of abstracts made the initial list very long. In these very limited trials, only one of them out of four managed to correctly predict the mask. The remaining examples have given the same incorrect prediction. This may be caused by the very general terms given in the abstract for the incorrect prediction. The large language model might choose this generally appropriate option if it fails to determine a better prediction.

As can be seen from these trials, the large language models are capable of correctly predicting the mask in the contexts of our task. However, they are mostly held back by their limited chat prompt sizes, and they still fail to give logical predictions most of the time.

We have shown two example trials for the prompting examples in Figure 4.1. The second one is an example of our global approach. Due to the space limitations, we have partially shown the list of citations and example contexts with their ground truth mask values.

In part (a) of Figure 4.1, we have shown one of our example prompting trials with the base approach. The upper section of our example is our prompt, while the lower section is the answer of the “Llama-2-70b-chat” model. In the prompt section, we start by providing a complete list of citations of the chosen dataset. The figure only shows this list’s start and end due to space limitations. After this list, we explain how the citation prediction operation works to the model. In the following five paragraphs, we provide five example contexts and their ground truth values to the model. Due to space limitations, we have only shown a single example in Figure 4.1. Lastly, we asked the model to return which citation should be predicted for the new input which is a masked context. The model answers this prompt by directly providing the predicted citation as ”Shwartz et al., 2016” and provides its reasoning for choosing this citation. As can be seen from the ground truth value, ”Shwartz et al., 2016” is the correct prediction.

In part (b) of Figure 4.1, we show one of our examples with the global approach. This example is mainly similar to part (a). The main difference between these examples is how the list of citations is provided. Each citation name is accompanied by its title and abstract. Because of space limitations, we only show the beginning of the list. Also, we could only fit 30 citations inside this list since the prompt input window of this model is limited. The remaining parts of the prompt stay the same. The answer of the model has the same contents as before. However, the model predicted the mask as ”Collobert et al., 2011”, which is incorrect since the ground truth citation is ”Kim, 2014”.

4.5. Discussion

Our study yields three significant insights. Firstly, we have observed the effects of treating word groups like citations as a single unified token. Normally, each citation is made up of multiple tokens for the default tokenizers of pre-trained language models. For example, a citation like “Nenkova and Passoneau, 2004” is tokenized as [’ĠN’, ’en’, ’k’, ’ova’, ’Ġand’, ’ĠPass’, ’on’, ’neau’, ’,’ , ’Ġ2004’]. However, with the addition of single unified tokens, it is tokenized straightforwardly as “Nenkova and Passoneau, 2004”. After this step, a model is further pre-trained for the citation prediction task. This simple tokenizer addition step results in significant improvements in the task of citation prediction. This approach may also have some potential for other tasks in NLP when there is a need for treating multiple-word tokens as a single unified one.

Our second observation from our study is the effect of the distribution of contexts per citation in the datasets. In the past works, these datasets’ contents have not been studied extensively. We believe that our observed distribution has very significant effects on the results of the citation prediction task. We have also shown multiple statistics and graphs of these datasets since past works have not performed such additional analyses. Further analysis of these datasets may also yield some additional factors on the results as well. Also, we have discovered many issues in the contents of the datasets, especially incorrect or missing author names and years. We also argue that these problems can cause some performance reduction in the results as well.

Our approach also possesses the following benefits. Hatten’s state-of-the-art work uses a two-step technique of pre-fetching and re-ordering. However, both our approaches

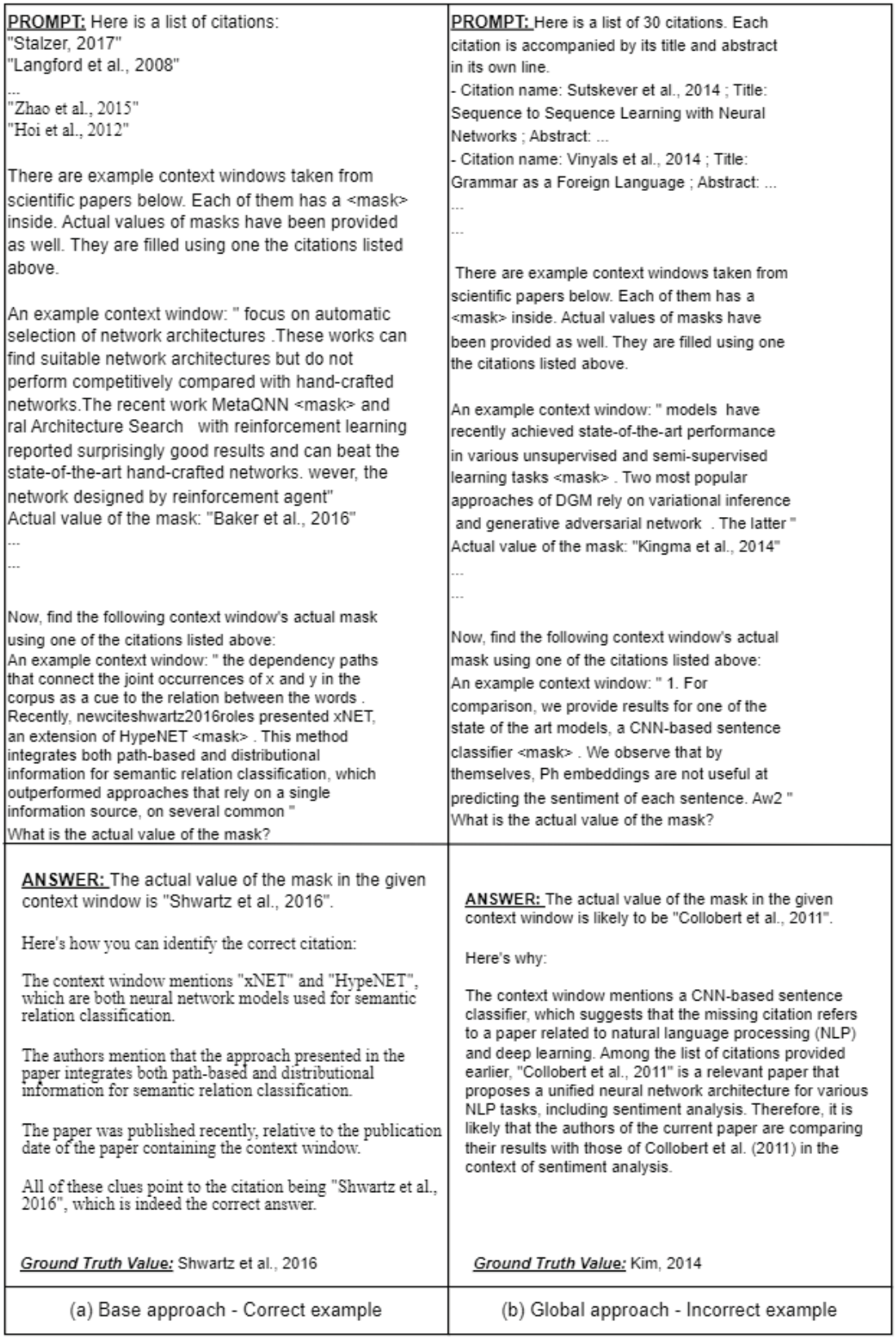

Figure 4.1. Prompting examples on large language models for both approaches.

consist of only a single step of mask prediction. While we need to add all citations as a single token to the model's vocabulary before training, it can be considered as a preprocessing step. This can be considered another advantage of our approach.

We have also seen the effects of the difference between the results of our base and the smaller dataset's global versions. There is a significant improvement between these two versions. The results of the smaller dataset's global versions are even capable of surpassing 90% in the results of our metrics. Interestingly, our base version results are also very good. They generally have similar or better results than the state-of-the-art results even though they only have access to the contexts from the datasets. However, the state-of-the-art models generally have complete access to the datasets, including titles and abstracts. Our global models can also access them, and they have significantly higher results. However, these observations only apply to the results of smaller dataset's models. The models of larger datasets cannot achieve the highly successful results of the other models. Both their base and global versions perform below our expectations.

There could be multiple reasons that cause the large model's results to underperform. We have generally aimed to have a large number of epochs in our models. However, we were not able to run our experiments with a large number of epochs on the larger datasets due to their long training times and our limited hardware resources. We performed their experiments up to 10 epochs. Since our large models may be unable to learn for our task properly, we only chose their $3 - 4$ epoch versions with the best results. So, these models might not have been trained enough to achieve good results. Although we have demonstrated that the number of epochs does not significantly impact the results for smaller datasets, there remains a possibility that 10 epochs may be insufficient for larger datasets. Due to the substantial size of these datasets, the models may not fully capture the connections between citations and contexts within only 10 epochs. Therefore, it is advisable to ensure adequate training is performed on these models to achieve optimal results.

There are other factors that affect the results of large models as well. One of the biggest causes behind the problem of large datasets may be their total number of citation items. Larger datasets contain significantly more citation names than smaller datasets. The model may experience difficulties while trying to learn how to predict citations when it has too many citations to choose from. This scenario can also be observed in our

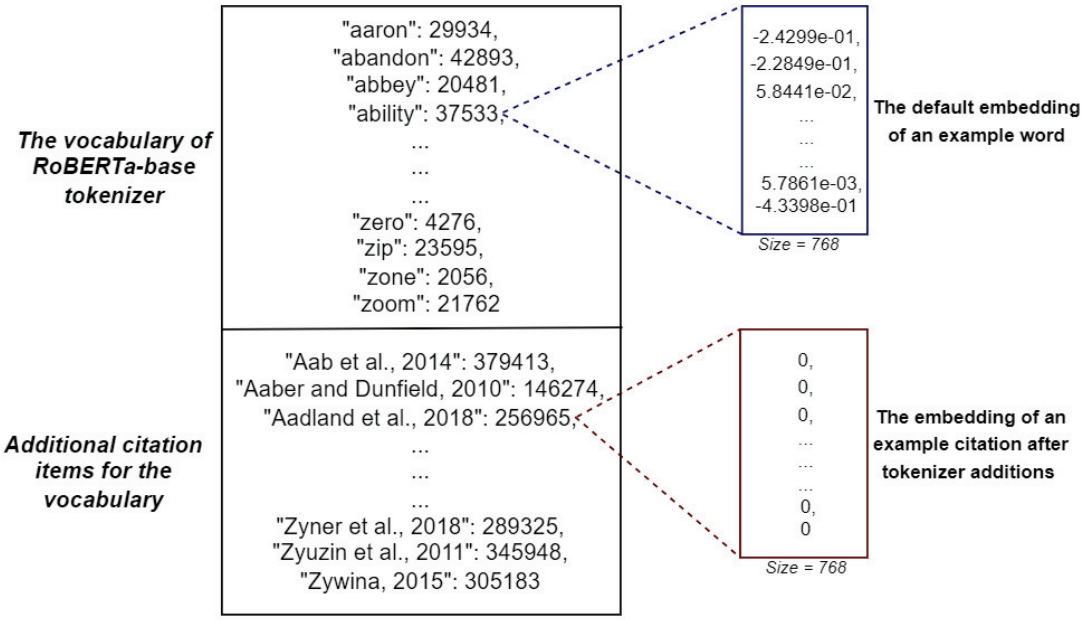

Figure 4.2. Comparison between the default word embeddings and citation embeddings in a tokenizer.

sampled datasets. Uniformly sampled datasets (Refseer-200k and Arxiv-300k-random) perform noticeably worse than negative sampling datasets. Their statistics, like mean, median, etc., might also be the cause of this problem. Since sampled datasets showcase the importance of factors like the number of citation items, we can expect a similar reason behind the problem of the larger datasets' models.

Another reason behind this problem may be how BERT-like tokenizers operate. These tokenizers can learn new words they have never seen from scratch. In other words, any new word's embedding can be learned during fine-tuning after initializing it as with all zeros. While this advantage of BERT-like tokenizers allows the models to be very adaptable, starting initial embeddings from 0 is still a disadvantage. This issue is illustrated in Figure 4.2. While this is also a problem for smaller models, they contain less total number of citation items. So, initializing new citation tokens as 0 does not affect them as significantly as larger models.

We have also performed training only on the RoBERTa-base model. It might have been possible to achieve better results under different conditions. Due to time limitations, we had to perform our model evaluations at certain number of epochs. Instead, we could have evaluated our models more often to select the best possible model for each dataset.

To solve the problem of large models, we could perform training on a model like RoBERTa-large to increase the total number of parameters in our model. It may be possible that RoBERTa-base might not have enough parameters to properly learn from large datasets with too many citation items. RoBERTa-large could capture a deeper understanding of larger datasets and achieve improved results on larger datasets. Additionally, we could try to initialize our tokenizers to have proper initial embeddings for manually added citation items. Instead of learning their embeddings from scratch, the model can benefit from proper initial embeddings.

Also, we could have used other pre-trained models like SciBERT or SpanBERT. Further pre-training on SciBERT instead of RoBERTa may have yielded better results since it shows better results on scientific texts. Meanwhile, SpanBERT could be used to predict multiple masks of a citation instead of a single one to remove the requirement of adding all citation items to the tokenizers. However, SpanBERT may lead to additional problems, as we have explained in the Methodology chapter. Alternatively, a model like T5 (Raffel et al. 2020) can predict multiple word tokens for a single mask token. This could have allowed us to try to predict multiple-word citations directly from a single mask token rather than adding all citations to the vocabulary of the model. Due to these reasons, it might have been possible to achieve better results in our experiments with larger datasets using alternative pre-trained models. The results of smaller datasets might have been further improved as well.

4.5.1. Limitations

To accurately evaluate the results of our research, we need to discuss certain limited and problematic aspects of our approach. We recognize the following limitations in this study. Our experiments were only performed on the four benchmark datasets used in the past works. While these datasets may be comprehensive enough for the task of citation prediction, they have multiple shortcomings. Two of the datasets are very small compared to the others. Especially, Peerread has less than 20000 examples. Also, the datasets have a significant amount of incorrect or missing information that causes the elimination of many of their examples.

Another limitation of our work is that our models can only produce results for their

own datasets. It will not work with the other datasets. This is caused by the need to know the names of the citation tokens to be able to predict them for a given context. Without the corresponding list of citations for a dataset, the model will not be able to predict relevant results for the contexts. However, it may predict citations that are close to the given context in relevancy as a suggestion mechanism.

These limiting factors are also a part of the past works as well. In fact, even the state-of-the-art model has these limitations while also requiring global information for its predictions. Meanwhile, our work can also function without global information, and it achieves slightly higher results compared to them. With the addition of global information, the success of our model further increases.

The main limitation of our approach is its results on the larger benchmark datasets. We could not achieve good results in larger datasets compared to smaller ones. Our model is capable of reaching results that can surpass state-of-the-art works for smaller benchmark datasets. However, it does not perform well for larger datasets because of reasons like a large number of citation items, initialization of citation tokens by the tokenizer, etc. Trying to predict citations from a single token might also be causing problems for our approach since we need to manually add citation name tokens to the tokenizer beforehand. Thus, in its current state, our work fails to perform adequately for two of the benchmark datasets.

CHAPTER 5

CONCLUSION

In this work, we have proposed a new technique for predicting citations inside scientific papers. We utilized an approach that further pre-trains a mask-filling language model in a unique way to achieve significant improvements over the state-of-the-art results on two of the benchmark datasets. Our novel approach focuses on treating each citation as a single-word token and adjusting the vocabulary of the model as necessary. Thanks to this approach, we were able to directly predict citations using our further pre-trained mask-filling models.

The models developed in our approach can be considered as language models capable of predicting citations. In other words, our model retains the functionality of a language model and can be fine-tuned accordingly. Through our custom pre-training strategy, our models can learn the representations of citations. Similar to SPECTER, each citation's embedding also represents its corresponding paper. These representations can be utilized to assess the proximity between papers. Unlike previous works, our models demonstrate greater robustness for application in other tasks. For instance, fine-tuning our model for scientific text generation may enhance the accuracy of citations generated within the output texts.

We also performed additional experiments to further analyze our results. Using our base and global approaches, we analyzed the effects of the different levels of information for citation prediction. Also, we observed the effects of the distribution inside the datasets and tried to find their relationship with the results. We investigated the benchmark datasets' statistics related to a factor we referred to as contexts per citation and observed how this factor affects the success of the model. Lastly, we believe our proposed model demonstrates a unique utilization method of mask-filling models on the task of citation prediction and leverages the strengths of the models' tokenizers in a novel way.

5.1. Future Work

For future work, we may need to investigate other predictions of our model's top 10 predictions for a given context. These additional predictions can also be considered similar scientific papers to the target citation's paper. So, it might be possible to consider them as suggested reading material for a given citation as well. We also believe that this technique of learning representations for a multi-word token as a single one can be used in other research areas as well. Additionally, we can restrict our model to only predicting citations instead of other words from the vocabulary. Since Hatten can only predict citations out of a pool, adding a similar restriction might allow us to compare our results under similar conditions while potentially improving our success rates.

The most crucial improvement we can make on our approach is to increase its result metrics on the larger datasets' models. To improve the results of large models, we should try to pinpoint the reason behind their failure to learn and predict citations from datasets that have a large number of citation items. Firstly, we can perform our training on different pre-trained models like SciBERT, RoBERTa-large, SpanBERT, or T5. Especially, T5 can predict multiple words for a single token. Using T5 can negate the requirement of manually adding all citation names to tokenizers and allow our model to achieve increased results on large datasets. The T5 architecture may enable models to predict citations for datasets on which they have not been specifically trained. In other words, T5 has the potential to overcome the limitation observed in both Hatten's approach and our approach, which rely on the ability to predict citations solely for the datasets used during training. Lastly, we can try to handle the initialization of the embeddings of our manually added citation names to tokenizers and aim to increase the success of our large models.

BIBLIOGRAPHY

Abrishami, Ali, and Sadegh Aliakbary. 2019. "Predicting citation counts based on deep neural network learning techniques." *Journal of Informetrics* 13 (May): 485–499. https://doi.org/10.1016/j.joi.2019.02.011.

Bai, Xiaomei, Fuli Zhang, and Ivan Lee. 2019. "Predicting the citations of scholarly paper." *Journal of Informetrics* 13 (February): 407–418. https://doi.org/10.1016/j.joi.2019.01.010.

Beltagy, Iz, Kyle Lo, and Arman Cohan. 2019. "SciBERT: A Pretrained Language Model for Scientific Text." In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP),* edited by Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, 3615–3620. Hong Kong, China: Association for Computational Linguistics, November. https://doi.org/10.18653/v1/D19-1371. https://aclanthology.org/D19-1371.

Brody, Tim, and Stevan Harnad. 2005. "Earlier Web Usage Statistics as Predictors of Later Citation Impact." *CoRR* abs/cs/0503020 (March).

Brown, Peter, Vincent Dellapietra, Peter Souza, Jennifer Lai, and Robert Mercer. 1992. "Class-Based n-gram Models of Natural Language." *Computational Linguistics* 18 (January): 467–479.

Chalkidis, Ilias, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. "LEGAL-BERT: The Muppets straight out of Law School." In *Findings of the Association for Computational Linguistics: EMNLP 2020,* edited by Trevor Cohn, Yulan He, and Yang Liu, 2898–2904. Online: Association for Computational Linguistics, November. https://doi.org/10.18653/v1/2020.findings-emnlp.261. https://aclanthology.org/2020.findings-emnlp.261.

Cohan, Arman, Waleed Ammar, Madeleine van Zuylen, and Field Cady. 2019. “Structural Scaffolds for Citation Intent Classification in Scientific Publications.” 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),* edited by Jill Burstein, Christy Doran, and Thamar Solorio, 3586–3596. Minneapolis, Minnesota: Association for Computational Linguistics, June. https://doi.org/10.18653/v1/N19-1361. https://aclanthology.org/N19-1361.

Cohan, Arman, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel Weld. 2020. “SPECTER: Document-level Representation Learning using Citation-informed Transformers.” In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics,* edited by Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, 2270–2282. Online: Association for Computational Linguistics, July. https://doi.org/10.18653/v1/2020.acl-main.207. https://aclanthology.org/2020.acl-main.207.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. “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),* edited by Jill Burstein, Christy Doran, and Thamar Solorio, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics, June. https://doi.org/10.18653/v1/N19-1423. https://aclanthology.org/N19-1423.

Dongen, Thomas van, Gideon Maillette de Buy Wenniger, and Lambert Schomaker. 2020. “SChuBERT: Scholarly Document Chunks with BERT-encoding boost Citation Count Prediction.” In *Proceedings of the First Workshop on Scholarly Document Processing,* edited by Muthu Kumar Chandrasekaran, Anita de Waard, Guy Feigenblat, Dayne Freitag, Tirthankar Ghosal, Eduard Hovy, Petr Knoth, et al., 148–157. Online: Association for Computational Linguistics, November. https://doi.org/10.18653/v1/2020.sdp-1.17. https://aclanthology.org/2020.sdp-1.17.

Elman, Jeffrey L. 1990. “Finding Structure in Time.” *Cogn. Sci.* 14:179–211. https://api.semanticscholar.org/CorpusID:2763403.

Gu, Nianlong, Yingqiang Gao, and Richard H. R. Hahnloser. 2022. "Local Citation Recommendation with Hierarchical-Attention Text Encoder and SciBERT-Based Reranking." In *Advances in Information Retrieval,* edited by Matthias Hagen, Suzan Verberne, Craig Macdonald, Christin Seifert, Krisztian Balog, Kjetil Nørvåg, and Vinay Setty, 274–288. Cham: Springer International Publishing. ISBN: 978-3-030-99736-6.

Guu, Kelvin, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. "REALM: retrieval-augmented language model pre-training." In *Proceedings of the 37th International Conference on Machine Learning*. ICML'20. JMLR.org.

Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. "Long Short-term Memory." *Neural computation* 9 (December): 1735–80. https://doi.org/10.1162/neco.1997.9.8.1735.

Huang, Shengzhi, Yong Huang, Yi Bu, Wei Lu, Jiajia Qian, and Dan Wang. 2022. "Fine-grained citation count prediction via a transformer-based model with among-attention mechanism." *Information Processing & Management* 59 (2): 102799. ISSN: 0306-4573. https://doi.org/https://doi.org/10.1016/j.ipm.2021.102799. https://www.sciencedirect.com/science/article/pii/S0306457321002776.

Joshi, Mandar, Danqi Chen, Yinhan Liu, Daniel Weld, Luke Zettlemoyer, and Omer Levy. 2020. "SpanBERT: Improving Pre-training by Representing and Predicting Spans." *Transactions of the Association for Computational Linguistics* 8 (July): 64–77. https://doi.org/10.1162/tacl_a_00300.

Jurgens, David, Srijan Kumar, Raine Hoover, Dan McFarland, and Dan Jurafsky. 2018. "Measuring the Evolution of a Scientific Field through Citation Frames." Edited by Lillian Lee, Mark Johnson, Kristina Toutanova, and Brian Roark. *Transactions of the Association for Computational Linguistics* (Cambridge, MA) 6:391–406. https://doi.org/10.1162/tacl_a_00028. https://aclanthology.org/Q18-1028.

Levine, Yoav, Barak Lenz, Opher Lieber, Omri Abend, Kevin Leyton-Brown, Moshe Tennenholtz, and Yoav Shoham. 2020. "PMI-Masking: Principled masking of correlated spans." *ArXiv* abs/2010.01825. https://api.semanticscholar.org/CorpusID:222134068.

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. *RoBERTa: A Robustly Optimized BERT Pretraining Approach.* arXiv: 1907.11692 [cs.CL].

Luo, Chu Fei, Rohan Bhambhoria, Samuel Dahan, and Xiaodan Zhu. 2023. "Prototype-Based Interpretability for Legal Citation Prediction." In *Findings of the Association for Computational Linguistics: ACL 2023,* edited by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, 4883–4898. Toronto, Canada: Association for Computational Linguistics, July. https://doi.org/10.18653/v1/2023.findings-acl.301. https://aclanthology.org/2023.findings-acl.301.

Medić, Zoran, and Jan Snajder. 2020. "Improved Local Citation Recommendation Based on Context Enhanced with Global Information." In *Proceedings of the First Workshop on Scholarly Document Processing,* edited by Muthu Kumar Chandrasekaran, Anita de Waard, Guy Feigenblat, Dayne Freitag, Tirthankar Ghosal, Eduard Hovy, Petr Knoth, et al., 97–103. Online: Association for Computational Linguistics, November. https://doi.org/10.18653/v1/2020.sdp-1.11. https://aclanthology.org/2020.sdp-1.11.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, G.s Corrado, and Jeffrey Dean. 2013. "Distributed Representations of Words and Phrases and their Compositionality." *Advances in Neural Information Processing Systems* 26 (October).

Neumann, Mark, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. "ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing." In *Proceedings of the 18th BioNLP Workshop and Shared Task,* edited by Dina Demner-Fushman, Kevin Bretonnel Cohen, Sophia Ananiadou, and Junichi Tsujii, 319–327. Florence, Italy: Association for Computational Linguistics, August. https://doi.org/10.18653/v1/W19-5034. https://aclanthology.org/W19-5034.

Pennington, Jeffrey, Richard Socher, and Christopher Manning. 2014. "GloVe: Global Vectors for Word Representation." In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP),* edited by Alessandro Moschitti, Bo Pang, and Walter Daelemans, 1532–1543. Doha, Qatar: Association for Computational Linguistics, October. https://doi.org/10.3115/v1/D14-1162. https://aclanthology.org/D14-1162.

Radford, Alec, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. “Improving language understanding by generative pre-training.”

Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer.” *Journal of Machine Learning Research* 21 (140): 1–67. http://jmlr.org/papers/v21/20-074.html.

Sun, Yu, Shuohuan Wang, Yu-Kun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. “ERNIE: Enhanced Representation through Knowledge Integration” (April).

Tanner, Chris, and Eugene Charniak. 2015. “A Hybrid Generative/Discriminative Approach To Citation Prediction.” In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* edited by Rada Mihalcea, Joyce Chai, and Anoop Sarkar, 75–83. Denver, Colorado: Association for Computational Linguistics, May. https://doi.org/10.3115/v1/N15-1008. https://aclanthology.org/N15-1008.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. “Attention is all you need.” In *Advances in neural information processing systems,* 5998–6008. http://arxiv.org/abs/1706.03762.

Yu, Xiao, Quanquan Gu, Mianwei Zhou, and Jiawei Han. 2012. “Citation Prediction in Heterogeneous Bibliographic Networks.” In *SDM*. https://api.semanticscholar.org/CorpusID:16401004.