3-27-2023 2:30 PM

Citation Polarity Identification From Scientific Articles Using Deep Learning Methods

Souvik Kundu, *Western University*

Supervisor: Mercer, Robert E., *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Master of Science degree in Computer Science

Abstract

The way in which research articles are cited reflects how previous work is utilized by other researchers or stakeholders and can indicate the impact of that work on subsequent experiments. Based on human intuition, citations can be perceived as positive, negative, or neutral. While current citation indexing systems provide information on the author and publication name of the cited article, as well as the citation count, they do not indicate the polarity of the citation. This study aims to identify the polarity of citations in scientific research articles using pre-trained language models like BERT, ELECTRA, RoBERTa, Bio-RoBERTa, SPECTER, ERNIE, LongFormer, BigBird, and deep-learning methods. Most citations have a neutral polarity, resulting in imbalanced datasets for training deep-learning models. To address this issue, a class balancing technique is proposed and applied to all datasets to improve consistency and results. Pre-trained language models are used to generate optimal features, and ensemble techniques are utilized to combine all model predictions to produce the highest precision, recall, and F1-scores for all three labels.

Keywords: Citation polarity, BERT, ELECTRA, RoBERTa, Bio-RoBERTa, SPECTRE, ERNIE, LongFormer, BigBird, Ensemble .

Lay Abstract

While writing one research article, citations are used very often to mention the prominent works from earlier periods of time that have motivated the current work or showed very good performance while tackling the same problem that the current paper is trying to solve. The intention of using the citation can be positive, negative, or neutral. Sometimes the readers need to read the referenced research work to grasp the ideas presented in the current paper. Knowing the citation intention will be very helpful for the readers as well before going through the referenced articles. The current citation indexing system provides a lot of information about the referenced article like names of the authors, publications, paper names, etc. However, the intention of using the citation is not possible to retrieve from this citation indexing system. That's why in this work I tried to develop a system that can capture the polarity of the citation used in the ongoing papers.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Citation polarity is described as the point of view that is presented in the citation text toward the cited material. This point of view can be either positive, neutral, or negative. Citation polarity analysis is one of the emerging research areas in the world of Natural Language Processing. (e.g., [2], [15], [30], [32]). The motivation behind introducing this task is to evaluate a research's impact, effectiveness, and contribution to that field. A citation basically reflects how other researchers or applied stakeholders use a previous research work in their new work and provide information on how the previous work has affected the current experiments [14]. As a result, citations can easily be polarised by human intuition whether they are positive, negative, or neutral. However, this became a complicated task even for advanced machine learning models since regular sentences and cited sentences carry different structures, and determining polarity is more complex in citation sentences. Citation indexing first showed that citations could help new research works and since then there has been a boon in this field [11]. Later on, Garzone and Mercer introduced a novel way to identify the internal meaning of a particular citation [12]. All these techniques paved the way for modern-day techniques that are being used in citation polarity.

In the past, the significance or influence of a research paper was evaluated by the numeric value of how many times it had been cited, so early citation analysis focused more on frequency rather than the impact of that work on the new research areas [34]. However, considering the modern-day scenario, this approach is never efficient since the number of research papers is increasing exponentially with time. Frequency can no longer be the only parameter to analyze a research work anymore. This problem led to the idea of detecting the polarity of citations as a means to evaluate citations. A research paper should be evaluated according to its impact on new research, i.e., whether the works agree, disagree, or acknowledge the previous work. A research paper can be labeled as impactful if it has a high correlation with other works done in the same field as well.

Biomedical journals have a significant impact on society since they directly work with the well being of a living creature. Quality and authenticity are very important factors in these research works. So, citation polarity can contribute to properly analyzing a biomedical research paper. The impact of work can never be analyzed solely on the frequency of citations. Using the polarity analysis can help determine which papers are more useful for the advancement of research. For example, biomedical papers suggest many different testing mechanisms on humans and other animals. These tests are expensive and sometimes hazardous. So, depending on the authenticity of a work we can assess whether this kind of test is even necessary or not. When a paper is cited with a positive sentiment in several other new novel works, it demonstrates the authenticity of the previous work and can be trusted.

However among regular research works, biomedical research papers belong to a niche area, and assessing the polarity of citations mentioned in the bio-medical journals is significantly different compared to other generic research papers. The first reason behind this is that these cited sentences consist of several medical-related terminologies like (α, β, etc.). As a result, using a general training corpus is never enough to train a model to predict the polarity of explicitly biomedical citations. This issue was resolved by Meng Jia [15]in her work where she prepared a stand-alone dataset only consisting of biomedical citations and annotated them so that supervising learning techniques can be used to predict the citation polarity of biomedical journals. The dataset was created from PubMed data and the extracted citations were manually annotated so that a gold standard corpus can be created. Her dataset was experimented with using techniques like SVM classifier with POS tags + (1-3) grams along with dependencies unigram.

The purpose of this thesis work is to propose a better model that produces better results compared to previous works and show how effectively citation polarity can be assessed for biomedical papers. Deep learning is extremely powerful when dealing with unstructured data due to its ability to process large numbers of features. The previous works used traditional classification models that cannot produce sufficient feature sets from the input of citation instances. In my work, I considered the shortcomings of the ongoing approach for citation polarity by class-balancing all the datasets. In order to prevent the models from becoming biased towards one class, balancing is required to train the models. In other words, even though it has more data, the model will no longer prioritize the majority class. Then I used BERT-based language models to produce better embeddings and fed all the features from the pre-trained models to the classification model. Using state-of-the-art word and sentence embeddings for every single dataset had better accuracy on test data. Afterward, ensembling techniques are also used to show how a voting mechanism can incorporate results from several models and work more effectively. Also, the thesis results show that the possibility of transfer learning so that the final

Table 1.1: Examples of the citation instances and corresponding labels from different datasets.

Dataset	Examples	Labels
Athar [2]	Fortunately, there is no straightforward generalization of the method of Smith and Smith (2007) to the two edge marginal problem.	Negative
	BLEU For all translation tasks, we report caseinsensitive NIST BLEU scores (Papineni et al., 2002) using 4 references per sentence.	Neutral
	A number of part-of-speech taggers are readily available and widely used, all trained and retrainable on text corpora (Church 1988; Cutting et al. 1992; Brill 1992; Weischedel et al. 1993).	Positive
Jia [15]	However , the determination of the solution structure of the ChBD-chiA1 by Ikegami et al. [CIT] identified only Trp687 as a putative chitin binding residue , in addition to His681 , Thr682 , Pro689 , and Pro693 .	Negative
	It was also suggested by the same authors that the carboxylate oxygen of the residue might be involved in the coordination of a Mg2+ ion shown to be important for the enzyme function [CIT] .	Neutral
	This deduced scenario is similar to the previously hypothesized scenario by van der Ven and Ober regarding HLA-G allogenic response [CIT].	Positive
CORD-19 [35]	However, in contrast to the results of a previous study (18), we found that HCoV-HKU1 was detected as frequently in patients with LRTI as in those with URTI, and there was a lack of association between HCoV-229E infection and respiratory symptoms in otherwise healthy individuals.	Negative
	CDV M proteins display only 3% amino acid variability, and the OS vaccine and 5804P wild-type strains differ at six positions (Fig. 1A), none of which is close to a putative late domain (amino acids [aa] 20 to 23, 23 to 26, 52 to 55, 311 to 314, and 332 to 335 [25]); to other functional motifs identified in closely related M proteins (V 101 VRT [24]); or to residues 64 and 89, which differ between MeV vaccine and wild-type strains (28).	Neutral
	The concept that overcoming apoptosis is of critical importance for the life cycle of JUNV would appear consistent with recent studies of JUNV infection that examined host cell interactions and revealed a positive modulation of cell viability upon infection (43).	Positive

models can perform well on any biomedical citation-related dataset in the future.

The corpora that have been used in this thesis are segmented into three class labels: Positive, Negative, and Neutral. Table 1.1 shows a few examples of the citation instances and corresponding labels.

The above-mentioned examples belong to three datasets. The first one is prepared by Awais Athar [2] who created the gold standard annotated dataset from the ACL anthology citation repository, the second one, also a gold standard annotated dataset is explicitly prepared by

Meng Jia [15] based on PubMed Central [28] which consists of full-text biomedical articles and the third one is the COVID-based dataset called CORD-19 [35] which is a silver standard annotated dataset compiled by the Allen Institute for AI.

The later chapters of the thesis work are constructed as follows: Chapter 2 describes several related research works: Word embedding, Sentence embedding, Attention mechanism, Pre-trained language model, Text level classification, Sentence Classification, Citation intent classification, and Multi-label text classification. After that, Chapter 3 demonstrates the entire methodology that involves data extraction, data cleaning, class/label balancing, data preprocessing, embedding techniques, and proposed Transformer architecture with ensembling techniques. The preliminary experimental results are presented showing the effectiveness of deep learning and transfer learning in Chapter 4. In the end, the last chapter provides a general summary of the entire research work along with the shortcomings and possible future work that can contribute to the development of this research attempt.

In summary, my contributions to this research work are as follows:

- Several pre-trained language models have been used here to prepare the features and they have then been forwarded to the proposed artificial neural network model to classify correct polarity labels of citation instances.
- We propose different class balancing techniques that are applied to the datasets to maintain consistency and produce better results.
- A merged dataset has been proposed combining Jia's [15] and Athar's [2] augmented datasets to train all the models and test Jia's actual work.
- We have also explored transfer learning to improve future models for assessing citation polarity in biomedical research.
- A new dataset has been prepared to consist of the paragraphs that belong to each citation entry from Jia's [15] work.
- Ensembling techniques have been implemented to combine all the results from different models and create a voting system to choose the most appropriate polarity of a given citation instance.

Chapter 2

Literature Review

This research highlights the use of word embeddings to convert words into numerical vectors, allowing for input into artificial neural networks using libraries such as Word2Vec [23] and FastText [5], amongst others. Document-level embedding using libraries such as Sent2Vec [26] is also discussed. Attention-based models are examined, as well as the Transformer Model [33] and its self-attention mechanism. The study concludes that pre-trained language models like BERT [10] can benefit transfer learning and unsupervised learning, but adjustments are needed for optimal results. The research aims to provide an accurate representation of the text-processing workflow and the application of various models to real-world tasks.

2.1 Word Embedding

Deep learning models excel at figuring out the underlying structures of the data and drawing conclusions about the task at hand from the data. However, these models need inputs in numerical form rather than raw string data because they can't be used with that. The process of mapping each word in the lexicon to a mathematical representation—more particularly, a particular vector representation—is known as word embedding. It makes the following models that use any large vocabulary more computationally demanding. Few techniques have been put out recently to create a vocabulary that is size-independent, preserves semantic information, and has a fixed-sized vector representation. Word embedding generation methods are covered in this section.

2.1.1 Efficient Estimation of Word Representations in Vector Space

In order to compute continuous vector representations of words from extremely huge datasets, Mikolov et al. [23] suggested two unique model designs. Researchers noticed a striking im-

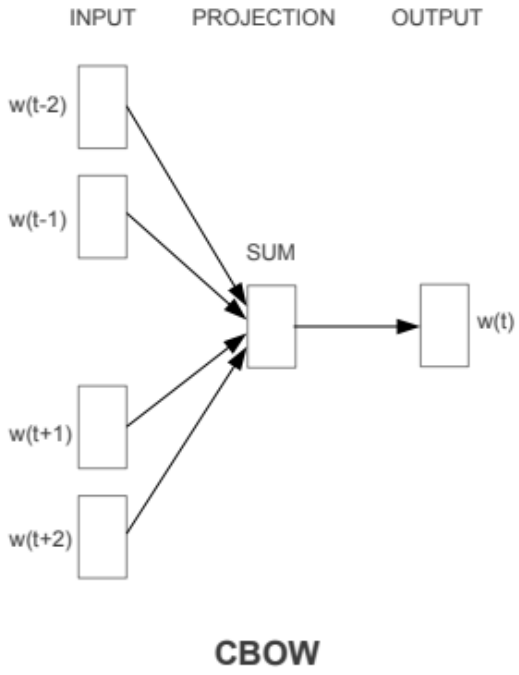

Figure 2.1: Word2Vec: Continuous Bag of Words Model [23]

provement in accuracy at substantially lower computational costs; for example, learning high-quality word vectors from 1.6 billion-word data set only requires one day. This study also showed that these vectors provide better performance for determining syntactic and semantic word similarity on the test set. Word2Vec's efficacy comes from its capacity to put together vectors of related word relationships. When provided with a sizable enough dataset, Word2Vec can make precise predictions about a word's meaning based on its usage in the text. These estimates result in word associations with other terms in the corpus. For instance, the representations of the words"King" and "Queen" are very close to each other since they both belong to royalty.

This model has two training methods. The first one is CBOW (Continuous Bag of Words). This design resembles a feed-forward neural network very much. With this model architecture, a target word is essentially predicted from a list of context terms. The idea behind this model is pretty straightforward: given the phrase "Hope you have a nice day," they select "a" as the target word, and the terms "hope," "you,","have,", "nice," and, "day" as our context words. Using the distributed representations of the context words, this model will try to forecast the target word.

Secondly, in its most basic form, the skip-gram neural network model is actually fairly straightforward. Although the model can be trained as a straightforward neural network with

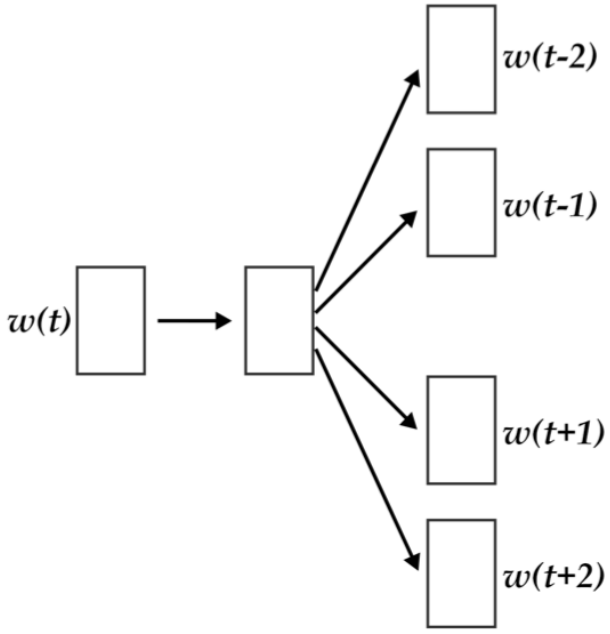

Figure 2.2: Word2Vec: Skip Gram Model [23]

a single hidden layer to carry out a certain task, the authors use it with a different purpose in their research. The actual objective is to simply learn the hidden layer's weights; as we will see all these weights correspond to the focused "word vectors". When given a word in the midst of a sentence, consider the words around it and pick one at random. The chance of each word in our vocabulary being the "nearby term" we selected will be provided to us by the network.

These models, which are founded on the Autoencoder principle, aim to preserve semantic information while mapping the vocabulary size-dependent one-hot encoding of words to a fixed-sized lower-dimensional vector representation. Because floating point numbers handle precision better, each element of this vector is a float value to guarantee precision. The word representation produced by this shallow model, which was trained on a corpus of 1.6 billion token words, gets more semantic and syntactic information than earlier models. Additionally, this model could bring words that are similar to one another closer together as well as multiple levels of similarity between words. To guarantee consistency for all the words in the input layer, this model accepts all the words as one-hot encodings. A one-hot form that can be used to calculate a chance distribution over each word in the corpus is also present in the output layer. For a given particular word or group of specific words, the model seeks to maximize the log probability of the output word in both cases. In order to calculate this probability distribution, they use a softmax layer at the output.

$$J = \max(\sum_{t=1}^{C} \sum_{k=-n,k\neq 0}^{n} \log P(w_t|w_{t+k})) \tag{2.1}$$

$$p(w_{t+k}|w_t) = \frac{e^{v'_{w_t}{}^T v_{w_{t+k}}}}{\sum_{w\in V} e^{v'^T_{w_t} v_{w_{t+k}}}} \tag{2.2}$$

The hidden-output layer component of these two Word2Vec models can be removed after training, allowing the model to produce low-dimensional vector representations of the words. This vector notation for CBOW pertains to context words, whereas for the SG model, it pertains to target words. Both of these models have higher accuracy with a larger context window size, while the computational time is reduced.

2.1.2 Enriching Word Vectors with Subword Information

FastText embeddings [5] create word embeddings using subword data. Words are expressed as the sum of the character-gram vectors, which are learned representations of character-grams. The word2vec type models gain subword information as a result. This makes it easier for the embeddings to understand ends and prefixes. After a word has been character-gram, a skiagram model is taught to discover its embeddings. FastText follows the same architecture and philosophy as the skip-gram Word2Vec model, with a few small differences. Unlike skip-gram, this model adds n-grams of subwords rather than vocabulary words. For example, the tokens generated for the word "*eating*" are $< ea, eat, ati, tin, ng >, apple$. Because each word is broken down into a group of character n-grams, the vocabulary sizes increase and a separate dictionary function must be maintained to identify each character n-gram, reducing the accuracy of the model even though it improves precision and gives a better semantic and syntactic representation of the words. The aim function of the skip-gram model is also modified to take subword usage into consideration. The word and subword vector representations' combined sum is what the FastText objective function tries to maximize.

$$j = \sum_{t=1}^{C} \sum_{k=-n}^{n} \sum_{w\in G_{w_t}} \log p(w_{t+k}|w) \tag{2.3}$$

Because this model also contains vector representations for subwords, FastText not only improves the quality of word representations but also has the potential to create vector representations for words that are not in the lexicon. In the case that any vocabulary words run out, this model constructs the vector representation by adding the vector representations of all possible subwords.

2.1.3 BioWordVec

Distributed word representations are now necessary for information search, text mining, and biomedical natural language processing (BioNLP). A large corpus of unlabeled text is typically used to calculate word embeddings at the word level, ignoring any information that may be present in ontologies or domain-specific structured resources that contain information about the internal structure of words. Such information has the potential to greatly improve the quality of word representation, according to some recent research in the field. BioWordVec [38] combines subword data from unlabeled biomedical text with a widely-used biomedical restricted vocabulary known as Medical Subject Headings to create an open collection of biomedical word vectors and embeddings (MeSH). For a number of biomedical NLP tasks, experts assessed the reliability and applicability of the created word embeddings. Their benchmarking findings show that for these challenging tasks, the word embeddings can perform noticeably better than the previous approaches.

The BioWordVec method first starts by creating a MeSH term graph from the MeSH RDF data in order to achieve this goal. A method of random sampling is used to create a number of MeSH term sequences. Then, by using the FastText model and training it on this data, they taught the FastText model text sequences as well as MeSH word sequences. The training process for BioWordVec is broken down here in detail in Figure 2.3.

Prior to sampling MeSH term sequences, relations in the MeSH term graph are used to create an ordered sequence of the heading nodes. To learn the word embedding for biomedical concepts, PubMed sentence sequences are merged with MeSH heading sequences. They applied a random walk method named node2vec [13] for selecting major heading nodes' MeSH term sequences from the MeSH term graph. Suppose, E, N and G denote the edge set, node-set, and graph respectively. If π_{vx} [equ. 2.5] is the transition probability from v to x and

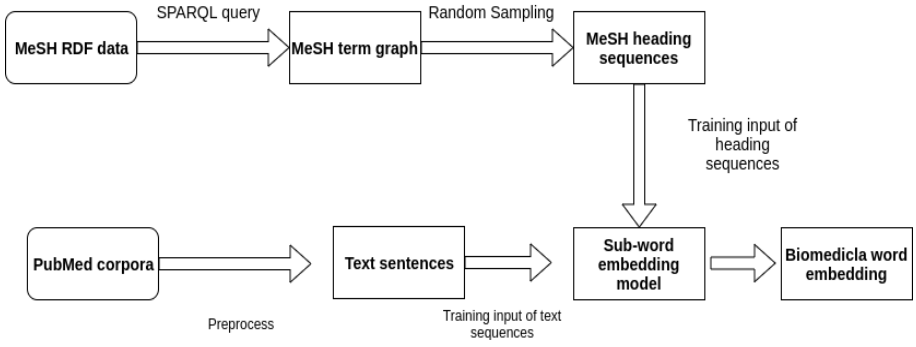

Figure 2.3: Schematic of learning word embedding based on PubMed literature and MeSH [38]

$d_{tx} \in \{0, 1, 2\}$ denotes the shortest path between two nodes x and t then the probability distribution of c_i is:

$$P(c_i = x|c_{i-1} = v) = \begin{cases} \pi_{vx} & if(v, x) \in E \\ 0 & otherwise \end{cases} \tag{2.4}$$

$$\pi_{vx} = \alpha(t, x) = \begin{cases} \frac{1}{p} & if d_{tx} = 0 \\ 1 & if d_{tx} = 1 \\ \frac{1}{q} & if d_{tx} = 2 \end{cases} \tag{2.5}$$

Unlike the traditional FastText [5] which works with only text sentences, BioWordVec incorporates the MeSH and segments of text from PubMed. For the PubMed data, the objective function si same as the traditional FastText model [equ. 2.3]. For the MeSH term portion, they defined the objective function as: The ultimate objective function of the model is a linear mixture of both of these objective functions.[equ. 2.6].

$$J = J_{PubMed} + J_{MeSH} \tag{2.6}$$

2.2 Sentence Embedding

By allowing the encoding of text as fixed-sized vectors, embeddings have significantly altered the field of natural language processing (NLP) in recent years. The development of a number of techniques for producing sentence embeddings often referred to as sentence vectors are one of the most recent innovations to result from this novel manner of encoding textual data. Longer texts can now be represented numerically as vectors that computer algorithms, such as machine learning (ML) models, can process easily, thanks to these embeddings. The main concepts underlying this technique will be covered in this section, along with a list of some of its potential uses and an overview of some of the most cutting-edge sentence embedding techniques now being used in NLP research and the language industry. For extracting and modifying the meaning of the segments they represent, embeddings are fixed-length, multi-dimensional vectors. One method is to determine how semantically similar two sentences are to one another. A sentence's concepts over the context must also be understood in order to comprehend it properly. A few sentence embedding approaches have recently been presented with this concept in mind. Furthermore, deep learning models can only deal with data of a constant length due to the changing nature of phrase length. This section investigates various unsupervised sentence embedding methods aimed at producing fixed-length vectors for

sentences containing contextual information.

2.2.1 Sent2Vec

Sent2Vec [26] has been proposed by researchers as part of this body of work. It is a simple unsupervised model that allows for the composition of sentence embeddings by combining n-gram embeddings and word vectors, all the while training both composing and the embedding vectors themselves. The computational expense of our embeddings is low both during inference and training of the sentence embeddings which is simply O(1) vector operations per word-processed. This holds true regardless of whether the embeddings are being trained or inferred. This allows this model to learn from extremely large datasets in a streaming manner, which is a major advantage in an environment where there is no human supervision. This is in stark contrast to all methods that are built on neural networks. The ability to draw conclusions quickly is a significant advantage in downstream activities and industry applications. This method shows notable performance benefits over the most cutting-edge unsupervised and even semi-supervised models currently available. When translated to a diverse collection of prediction benchmarks, the general-purpose embeddings that were produced as a result demonstrate high levels of robustness. This paradigm can be thought of as an optimization issue similar to:

$$\min_{U,V} \sum_{S \in C} f_S(UV\iota_S) \tag{2.7}$$

where, $\mathcal{V}$ is the vocabulary, $U \in \mathbb{R}^{\mathcal{V} \times h}$ and $V \in \mathbb{R}^{h \times |\mathcal{V}|}$ are the two parameter matrices. Columns in matrix V and U show the vector representation of the context words and target words respectively.

The complete formula for the sentence embedding for any sentence S is defined as:

$$\nu_S = \frac{1}{|R(S)|} V\iota_{R(S)} = \frac{1}{|R(S)|} \sum_{w \in R(S)} \nu_w \tag{2.8}$$

where $R(S)$ denotes the set of all possible n-grams as well as the uni-grams for any sentence S.

The final function according to this research work has been proposed as:

$$\min_{U,V} \sum_{S \in C} \sum_{w_t \in S} (q_p(w_t)\ell(u_{w_t}^T \nu_{S \setminus \{w_t\}})) + |N_{w_t}| \sum_{w' \in N_{w_t}} q_n(w')\ell(-u_{w'_t}^T \nu_{S \setminus \{w_t\}}) \tag{2.9}$$

This model not only offers cutting-edge speed but is also very straightforward and inexpensive to compute. Once the model is trained, for any sentence S, it requires only $|R(S) \times h|$ floating point operations for the n-gram. The computational complexity of this model is $O(1)$ vector op-

eration per word only. Furthermore, it allows parallel training using parallel stochastic gradient descent due to its simplicity.

To conclude, in this paper [26] they presented an original unsupervised approach that is computationally efficient and can be used to train and infer language embeddings. This technique, on average, delivers greater performance than all other unsupervised competitors, with the exception of the SkipThought vectors, when supervised evaluations are used to judge it.

2.3 Attention in Natural Language Processing

The attention mechanism, one of Deep Learning's most recent innovations, was created especially for natural languages processing tasks like machine translation, picture captioning, dialogue creation, and other similar tasks. Encoder decoder (seq2seq) RNN models can benefit from this method, which was created to improve their overall performance. The model searches for a set of locations in the source phrase where the most important information is concentrated when trying to predict the next word in attention. The model then predicts the next word based on all of the previously generated target words as well as the context vectors related to the source locations in question. Rather than putting the input sequence into a single fixed context vector, the attention model generates a context vector that is specifically filtered for each output time step. This allows the model to produce more accurate results. [3]

2.3.1 Hierarchical Attention Networks for Document Classification

Yang et al. [36] suggested a hierarchical attention network for categorizing documents. This model has two standout features: 1) A hierarchical structure that mimics the hierarchical structure of documents, and 2) two levels of attention mechanisms applied at the word- and sentence level that allow the model to pay differential attention to more and less important content when building the document representation. Experiments on six large-scale text categorization problems show that the suggested architecture performs significantly better than earlier approaches. The attention layer visualization shows how the model chooses highly informative words and sentences.

In this work, they have concentrated on document-level classification. Considering a text with L consisting of s sentences s, each of which has T words. The words in the ith phrase are represented by wit with t [1, T]. With the help of the suggested model, they can convert a raw document into a vector representation on which we may base a classifier to do document classification. The process of gradually constructing the document-level vector from word vectors utilizing a hierarchical structure.

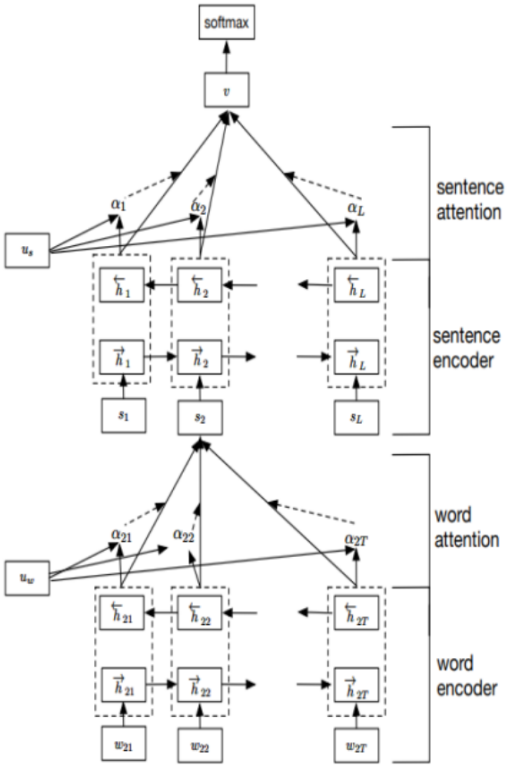

Figure 2.4: The Architecture of Hierarchical Attention Network [36]

$$\bar{u}_i t = \tanh(W h_i t + b_w) \tag{2.10}$$

$$\alpha_i = \frac{e^{\bar{u}_i t^T} u_w}{\sum_i e^{\bar{u}_i t^T u_w}} \tag{2.11}$$

$$s = \sum_t \alpha_i t h_i t \tag{2.12}$$

The attention-weighted word representation of the sentence's words is added to produce the sentence vector representation. The sentence vectors are handled by Bi-GRU in the following section of the model in a manner similar to how the word embeddings were handled, and a sentence-level annotation is generated. The attention mechanism is then used to create the vector by applying it to these sentence annotations, as was previously indicated.

Whatever the data size, the improvement is still prominent. The proposed model here improved the previous best baseline methods by 3.1% and 4.1%, respectively, for smaller data sets like Yelp 2013 and IMDB. This result holds true when compared to other, larger data

sets. This also exceeded the prior best models on Yelp 2014, Yelp 2015, Yahoo Answers, and Amazon Reviews by 3.2%, 3.4%, 4.6%, and 6.0%. Additionally, the increase happened irrespective of the task: sentiment classification, which incorporates Yelp 2013–2014, IMDB, and Amazon Reviews.

The attention mechanism is as follows:

$$M = \tanh(W^y Y + W^h R_{mean} \otimes e_L) \tag{2.13}$$

$$\alpha = softmax(w^T M) \tag{2.14}$$

$$R_{attention} = Y\alpha^T \tag{2.15}$$

Here, Y is the matrix comprising of the Bi-LSTM output vectors, R_{mean} denotes the output of the mean-pooling layer, $R_{attention}$ represents the attention (α) weighted vector representation of the sentence.

In the end, researchers show that by using their proposed model with much fewer parameters they got better accuracy of 85.0 on test data compared to other sentence encoding and neural network models on the SNLI tasks.

2.3.2 Transformer

Encoders and decoders are components of the large neural networks that underpin the majority of sequence-transduction models. These networks can be convolutional or recurrent. The models with the best overall results link the encoder and the decoder through an attention mechanism. Researchers here proposed a new, straightforward network design that we call the Transformer [33]. It is purely predicated on attention processes and does not make use of recurrence or convolutions in any way. Experiments conducted on two different tasks involving machine translation demonstrate that these models have higher quality, are more parallelizable, and need a much shorter amount of time to train. It is demonstrated that the Transformer is applicable to a wide variety of additional tasks by effectively applying it to the parsing of English constituency data, both with a huge amount of training data and with a restricted amount.

The encoder is shown on the left side, which converts the input sequence $(x_1, ..., x_n)$ into a sequence of continuous representations $z = (_1, ..., z_n)$, and the decoder is shown on the right side, which decodes the output of the encoder combined with the output (embeddings) to produce the output sequence The decoder operates in an auto-regressive fashion, meaning that for each output, the model takes in as new input all of the symbols that were generated before it.

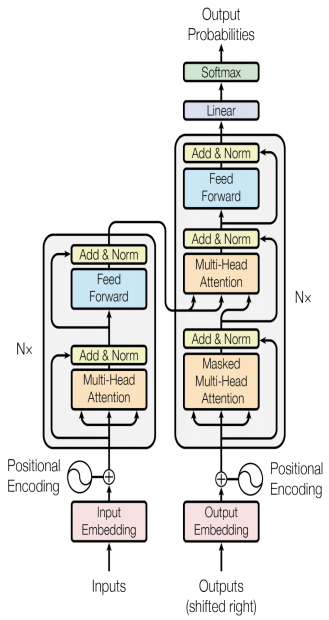

Figure 2.5: The Transformer - model architecture [33]

Figure 2.5 clearly demonstrates that the architecture of the transformer is made up of encoder and decoder blocks, and these blocks can be layered (Nx) onto one another to produce a model that is ever more complicated.

The encoder blocks in the vanilla transformer architecture have N=6 blocks where every block has 2 sub-layers, a position-wise completely connected feed-forward network, and a multi-head self-attention mechanism. A residual connection is used at the boundary of each sublayer. After that, layer normalization is performed. LayerNorm(x + Sublayer(x)) is the final result of each sub-layer. The model's outputs have a dimension of 512 and this holds true for all embedding layers and sub-layers.

The decoder follows a format that is analogous to that of the encoder. The decoder makes use of an additional multi-head attention layer, which operates on the output of an encoder block, which is the primary distinction between the two. Once more, residual connections are used all the way around each sub-layer, and then layer normalization comes after that. The multi-head attention that operates on output embeddings masks all subsequent positions for every position, which means that a prediction for step i is solely dependent on positions that have been observed previously.

At the beginning of the attention section of the study, a concise description of what attention is and how it functions in deep learning is provided. This explanation of attention maps a query and many key-value combinations to an output, which is the weighted sum of the values. Computed as the compatibility function between the query and the value's key, the weight of a

value is determined in this manner.

There are two methods that the transformer architecture uses attention: the first is scaled dot-product attention, and the second is multi-head attention, which uses scaled dot-product attention internally. While considering one word, the transformer calculates a score for every word in the sequence against that corresponding word. This score depicts the level of focus that the considered word should give on the other words. The final output of the self-attention layer is computed by:

$$Attention(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d}})V \tag{2.16}$$

Multiple queries q (keys k, values v) have been stacked to form a matrix here, denoted by the notation Q (K, V). These matrices are the result of multiplying the input matrix X by a set of previously learned matrices called W^Q, W^K, and W^V. As a consequence of this, each and every one of Q, K, and V is a weighted representation of the input X. By multiplying them, attention ensures that all of the components x of the input X are evaluated against one another and that it is possible to identify pairings of components that are coherent with one another. This eliminates the need for sequential calculation and offers a better understanding of the links that exist between the various components of the input.

Control can be exerted on the dimensions of Q, K, and V by manipulating the dimensions of W^Q, W^K, and W^V respectively. This makes it possible to create representations of varying degrees of complexity, according to the requirements and available resources. The same operation repeats for all the encoders in the stack. The decoder that occurred before an encoder-decoder block provides the queries, denoted by q, while the encoder's output provides the keys and values. Since the decoders take the keys and values from the encoder rather than the previous decoder, they can access the information included in the input embeddings as opposed to only depending on the information present in the output embeddings. The output of the encoder that came before it is used as a source for the queries, keys, and values that are generated by an encoder. This indicates that an encoder has access to all positions and representations stored in the memory of the encoder that came before it.

In order to evaluate the model's success in two different facets of machine translation, both the English-German and English-French datasets were used. The Transformer model outperformed all other earlier models in both datasets, and also lowered the amount of time required for training by approximately a quarter of what it was initially.

2.3.3 Structural Scaffolds for Citation Intent Classification in Scientific Publications

Understanding the function of a citation in scientific papers is crucial for automated analysis of the body of scientific literature and machine reading of specific publications. For instance, background data, methodology used, and outcome comparison. Cohan et al. [8] suggested structural scaffolds, a multitasking method that incorporates structural data from scientific papers into citations, in order to effectively classify citation intents. This model outperforms the state-of-the-art on an existing ACL anthology dataset (ACL-ARC) without using hand-engineered features or external linguistic resources, as done by other methods, with a 13.3% absolute increase in F1 score. Additionally, they debuted SciCite, a brand-new dataset of citation intents that is five times larger than earlier datasets and covers a broader variety of scientific fields.

Citations can take various forms. While some citations just serve to acknowledge earlier work, others reflect actual method application. As a result, understanding the purpose of citations is crucial for enhancing automated analysis of scholarly literature and scientific influence measurement. Existing feature-based models for this issue analyze the citation context in relation to a collection of manually created features (like linguistic patterns or cue phrases), disregarding additional signals that might enhance prediction. In order to avoid the issues related to external features, the researchers claimed in this article that better representations can be obtained straight from data. They consequently suggested a neural multitask learning framework for knowledge incorporation into scientific paper citations based on their structure.

Predicting the section title and determining whether a sentence requires a citation are the two scaffolds, and predicting the citation intent is the main task (top left) (citation worthiness). This work proposes a neural multitask learning framework for the classification of citation intents. They introduced and used two structural scaffolds in particular, which are support activities linked to the format of scientific papers. Although the auxiliary tasks may not be interesting in and of themselves, they provide information for the primary task. This structural data from scientific papers are incorporated into the citation intents of our model using a sizable auxiliary dataset. Figure 2.6 depicts this model's overall layout.

The first scaffold task that they took into consideration is a sentence's "citation worthiness," which determines whether a sentence requires a citation. Knowing that the language used in citation sentences varies from normal sentences in scientific writing might be useful for more precise language modeling of the citation contexts. In a publication, researchers label the sentences that contain citations, and the negative samples are the phrases that lack citation markers. The model's objective in this job is to forecast whether a given sentence calls for a

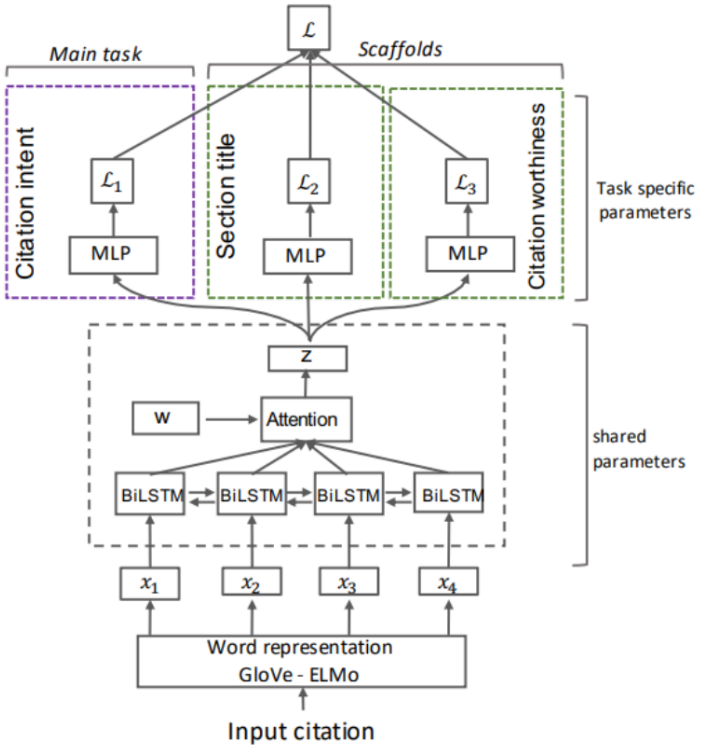

Figure 2.6: Architecture of the Scaffold model [8]

citation.

The second scaffold job is to predict the section title in which a citation appears. Scientific papers frequently adhere to a standard format in which the authors present the issue, explain the methodology, present the results, discuss the findings, and finally, draw a conclusion. The goal of a citation might be pertinent to the paragraph in which it appears in the paper. Citations pertaining to techniques, for instance, are more likely to be found there. As a consequence, they constructed a scaffold for predicting citation intents using section title prediction. It's crucial to understand that this scaffold job is distinct from merely including a section title as an extra feature in the input. Using section titles from a bigger set of data than the training set for the primary task, the model is learning linguistic patterns that are helpful for citation intents. For this scaffold job, a large number of scientific papers have been used, for which it is known that the section information for each citation automatically generates large amounts of training data.

Multitask learning is an inductive transfer learning approach that improves generalization by using domain information contained in training signals from related tasks as an inductive bias. It is necessary for the model to have at least some parameters that can be shared between tasks. As illustrated in Figure 2.6, each scaffold task will have task-specific parameters for effective classification, while the parameters for the network's lower layers will be shared across

tasks. For each task, they used a Multi-Layer Perceptron (MLP) and then a softmax layer to calculate prediction probabilities.

$$y^{(i)} = \text{softmax}(\text{MLP}^{(i)}(z)) \tag{2.17}$$

The researchers found that their scaffold-enhanced models outperform the most recent method on this test by a wide margin. Adding the first scaffold job from "BiLSTM-Attn + section title scaffold" raised the macro F1 score from the initial "BiLSTM-Attn" baseline of 51.8 to 56.9 (= 5.1). Similar benefits are also seen when the second scaffold in "BiLSTM-Attn + citation worthiness scaffold" is added: 56.3 (=4.5). The F1 score increases to 63.1 (= 11.3) when both scaffolds are utilized concurrently in "BiLSTM-Attn + both scaffolds," indicating that the two tasks offer to complement signal that is helpful for citation intent prediction. Results from the SciCite dataset, where we find comparable patterns. Model performance is improved by each scaffolding task. Further advancements emerge from combining the two scaffolds. Additionally, employing ELMo[29] representation in addition to both scaffolds yields the greatest results. Given that this dataset is more than five times larger than the ACL-ARC, the performance figures are generally higher and the F1 improvements are typically smaller. This is because it is easier for the models to acquire optimal parameters using the larger annotated data. The best baseline on this dataset gets an F1 score of 82.6 with the neural baseline plus ELMo contextual vectors. Given that neural models frequently produce greater gains with more training data, this was anticipated.

Chapter 3

Methodology

This chapter discusses how my work has been executed to produce the results in Chapter 4. It begins with a description of three sentence-based datasets, which includes any necessary dataset preprocessing, cleaning of the dataset, and class balancing. Following this, the deep learning language models that are used are introduced. Then, the proposed transformer architecture is trained to classify sentences into one of three citation polarities. Next, the ensembling techniques are detailed. Finally, the creation of a paragraph-based dataset is outlined.

3.1 Description of the Datasets

3.1.1 Meng Jia's Sentence-based Dataset

Meng Jia in her MSc thesis work [15] created a dataset consisting of 778 sentences containing citations. These sentences were obtained from an XML dataset that was available from PubMed Central [6], a source for full-text biomedical research articles.

Each sample in the dataset consists of the polarity label, the citation sentence itself, and a parse of the sentence. She was interested not only in the three citation polarities, Neutral, Positive, and Negative but also subcategories of them. Neutral consists of Perfunctory/Background, Statement, General comparison, and Multi-comparison subcategories. Then Confirmation and Being-confirmed subcategories make the Positive main group. Finally, Negative includes Contrast/Conflict and Unsolved subcategories. The dataset was labeled with eight subcategories. So, some pre-processing of the dataset was done: the labels were changed to the three main polarities and the parsed sentence was removed.

Annotating a citation dataset is comparatively difficult since it involves consideration of authors' intentions and sentiments [32]. The dataset was annotated by Meng Jia. She was the only annotator so she reported the Cohen κ using the intra-annotator agreement approach from

Athar's research [1]. The annotation has a κ score of 0.71 which means that the annotation has a substantial agreement and provides validation to the fact that the annotated corpus is reliable [19]. This is the only human-annotated dataset with citation sentences drawn from the biomedical literature that I am aware of.

Citation datasets are highly skewed towards the Neutral category and the Negative sample count is negligible. This dataset is no exception. The distribution is highly imbalanced: 69.9% of entries belong to Neutral, 22.9% to Positive, and the remaining 7.2% to Negative. Balancing of the classes is described in Section 3.2.

The goal of this thesis is to provide a deep learning model that performs well on the citation polarity task for citations found in the biomedical literature. So, I have used this dataset as a base of my work to evaluate the performance of the proposed models.

3.1.2 Awais Athar's Sentence-based Dataset

Awais Athar in his work [2] created a citation corpus based on the ACL anthology dataset. A new corpus was annotated under this work with more than 8,736 citation sentences labeled as positive, negative, and neutral/objective. Under this work best micro-F score of 0.760 was reported using a combination of n-grams of length and dependency relations. Like the dataset of Meng Jia, this dataset is also highly skewed towards objective/neutral sentiment. Among 8,736 sentences, 7627 are objective. The remaining consists of 829 positive and only 280 negative entries. This annotated dataset has $\kappa = 0.89$ (N = 8736; k = 1; n = 3). Here N is the total number of annotations, n is the number of classes and k is the number of annotators. This score showing the intra-annotator agreement validates that it is a stable annotation according to Krippendorff (1980) [17]. This dataset has been used to train models dedicates to this corpus as well as for transfer learning. The κ score and other validating grounds make this a gold standard dataset for citation polarity detection.

3.1.3 CORD-19: The COVID-19 Open Research Dataset

In my thesis work, I have also incorporated this dataset consisting of citations of research papers solely dedicated to COVID-19 and related historical coronavirus research works. CORD-19 gained significant popularity and this dataset has been downloaded over 75k times showing the legitimacy and acceptance of this corpus for different tasks [35].

This dataset has over 52K papers with over 41K full texts. It also includes papers published in more than 3200 journals. The citations mentioned in this dataset belong to three main labels: Supporting, Contradicting, and Mentioning. These labels can also be interpreted as Positive, Negative, and Neutral like the other datasets used in my research work. The majority portion

Table 3.1: Distribution of labels in CORD-19 Datase.

Label Name	Number of Samples
Mentioning	348,438
Supportive	14,053
Contradicting	1,705

of this corpus is collected from PubMed Central (PMC). Besides PMC, bioRxiv, and medRxiv are also significant sources behind the creation of this dataset.

From Table 3.1 above, there are 364,196 citations in this corpus. Among this, 348,438 entries are labeled as mentioning (Neutral), 14,053 as supportive (positive), and the remaining 1,705 as contradicting (Negative). There are diverse sub-fields incorporated in this dataset which makes it more versatile. These subfields are Virology, Immunology, Molecular biology, Genetics, Intensive care medicine, and others. This dataset has been used for multiple competition-level tasks like text mining competition by Kaggle, TREC-COVID shared task, etc. This dataset is also highly imbalanced like the above-mentioned datasets. In order to solve this situation, the downsampling method has been used. Contradicting has the lowest number of entries. So, using 1705 as the reference value both mentioning and supportive categories have been downsampled. After creating the class-balanced dataset it was split into train and test sets to maintain the consistency of the ratio of the labels. The main motive behind using this work under my thesis is to contribute to the ongoing pandemic situation and experiment with transfer learning so that the proposed model architectures can be immune to any new dataset.

3.1.4 Proposed Paragraph-based Dataset Converted from Jia's Sentence Dataset

Based on the same dataset from Meng Jia[15], I prepared a new dataset with the same number of citation entries but rather than having a single sentence it now has a paragraph corresponding to a polarity label. Most of the citation datasets only consist of a single sentence. However, often using single sentences causes ignorance of multiple important phenomena. Single-line citations are over-simplistic at times and it becomes difficult to identify the sentiment properly. Research papers seldom have rich discussions of cited work in multiple sentences and they can share multiple intents or emotions simultaneously. In order to address this issue a new dataset has been proposed called MULTICITE [20].

This dataset contains multiple sentences for every single entry in the dataset with multi-label context. To incorporate this idea in my thesis work I prepared a version of the MULTI-

Table 3.2: Examples of the citation instances and corresponding paragraphs.

Citation Instance	Corresponding Paragraph
In continental Europe , several large surveys on the throat bot flies on roe deer have been performed (e.g. [CIT]) , but only C. stimulator has been reported	The only published reports we can find where C. ulrichii has attacked other species than moose, are a case in which first instar larvae were found in the conjunctival sac in the eye of a human [19], and another case where 39 young larvae were deposited by a female C. ulrichii on the upper lip of a human [1]. In continental Europe, several large surveys on the throat bot flies on roe deer have been performed (e.g. [20]), but only C. stimulator has been reported. Reindeer throat bot fly larvae have been found in the nasal cavities of dogs in Sweden [21].
The survival percentage of 63 for all colic cases in the present study is on the same level as reported from other studies on hospitalised colic cases [CIT].	The survival percentage of 63 for all colic cases in the present study is on the same level as reported from other studies on hospitalised colic cases [2,15]. The probabilities of survival in surgically and medically colic cases (48 and 78%, respectively) also correspond to previous studies [15,16].
There were found no discrepancies between the phenotypic character demonstrated for present isolates and the ones outlined for the different Mannheimia species by Angen et al. [CIT] .	There were found no discrepancies between the phenotypic character demonstrated for present isolates and the ones outlined for the different Mannheimia species by Angen et al. [1]. However, given the considerable genetic diversity among the isolates classifying with M. ruminalis will be referred too, as M. ruminalis-like organisms. Interestingly, all isolates classified with M. ruminalis were $\beta - haemolytic$ on ovine blood, although this species generally is regarded as non-haemolytic. The M. ruminalis-like organisms were the most prevalent Mannheimia species present in all four flocks.
The usefulness in using $\beta - glucuronidase$ activity for the identification of *E.coli* has been confirmed by Schraft and coworkers who reported almost identical colony counts based on $\beta - glucuronidase$ activity and on classical biochemical reactions [CIT].	Typical E. coli on SEC plates appear as blue-green colonies as it has been found that about 97% of *E.coli* produce $\beta - glucuronidase$ which reacts with an indicator dye in the plate media to produce dark green to blue-green colonies. Colonies other than *E.coli* are not conspicuous because they are colorless or have a light grey-beige color. The usefulness in using $\beta - glucuronidase$ activity for identification of *E.coli* has been confirmed by Schraft and coworkers who reported almost identical colony counts based on $\beta - glucuronidase$ activity and on classical biochemical reactions[6].

CITE dataset [20] dedicated to only biomedical research papers. Following the work by Meng Jia [15], I traversed back to the original full article repository of PubMed from where the citations were initially retrieved and annotated into multiple labels. I extracted the entire paragraph from those single-line citations and appended them to the dataset. So this new dataset has both

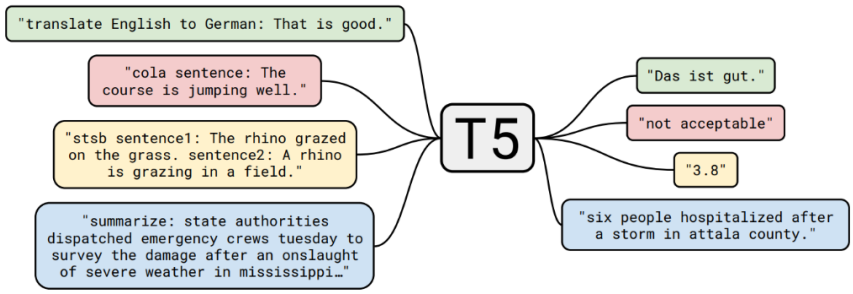

Figure 3.1: Application of Transformer (T5) model [27]

single-line citation sentences along with their corresponding paragraphs. The full articles were in XML format so I first converted them into a python data frame and using advanced query extracted the paragraph that belonged to each individual single-sentence citation. I kept the sentiment polarity the same for the paragraphs since all of the individual citations belong to each paragraph. This new dataset was cleaned, which is described later in this chapter.

Table 3.2 shows sentences belonging to the corresponding paragraphs.

3.2 Data Augmentation

All three above-mentioned datasets are class imbalanced. The common factor is that most of the citation sentences belong to the Neutral category. As a result, the models tend to predict every test example as neutral. Even though apparently these distribution produces decent accuracy, the f1 score, precision, and recall for other labels were significantly poor. To solve, this issue I needed to augment the datasets. There were two viable options available. The first one is duplicating the citation entries. The second option was paraphrasing the classes that had fewer entries. Duplication sometimes creates extra redundancy and the model does not get new features to learn for these values. However, paraphrasing produces different sentences keeping the sentiment the same as the original sentence. So, I opted for the paraphrasing option, and upon test results, this produced better accuracy as well.

The method that has been used here uses a fine-tuned Text-To-Text Transformer (T5) model. T5 is a transformer-based architecture and the main principle of this model is based on a text-to-text approach. There are some significant differences between this T5 model and conventional BERT.

For my setup to perform data augmentation, I used the TextGenie library. It first does paraphrasing using the above-mentioned T5 model then uses BERT mask filling to find keywords and replace them with a mask that is being fed to the BERT model to predict another similar

Paraphrased samples from Jia's Dataset [15]

Sentence from Jia's Dataset	Paraphrased version
The survival percentage of 63 for all colic cases in the present study is on the same level as reported from other studies on hospitalized colic case	In the present study, the survival percentage of 63 was on the same level as reported from other studies on hospitalised colic cases
	There is a survival percentage of 63 for all colic cases in present study. This is the same as reported from other studies on hospitalised colic cases
	The survival percentage of 63 is the same as reported from other studies on hospitalised colic cases.

Table 3.4: Paraphrased samples from Athar's Dataset [2]

Sentence from Athar's Dataset	Paraphrased version
Fortunately, there is no straightforward generalization of the method of Smith and Smith (2007) to the two edge marginal problem.	In this way, there is no straightforward generalization of the method of Smith and Smith (2007) to the two-edge marginal problem.
	Here, at least, there is no straightforward generalization of the method of Smith (2007) to the two-edge marginal problem.
	After Smith and Smith (2007), luckily here is no direct generalization of the two edge marginal problem.

word to put in the masked word. Finally, this library considers the voice of a sentence. If a sentence is fed in a passive voice, it transforms into an active voice.

In addition, to initialize this library a spaCy model name is also mandatory. The default model "en" has been used. The parameters for this setup are: sent (the given sentence), prefix (it is for the T5 model input), *n_predictions* (this determines how many augmentations the function should return), *top_k* (maximum number of predictions), and *max_length* (maximum length of a sentence that can be given the model, the default value is set to 256.

Tables 3.3 and 3.4 show examples of citation sentences respectively from Jia's and Athar's work. Using the above mentioned library, I produced three paraphrases for each sentence. Since the quantity of neutral labeled sentences is significantly higher, paraphrasing has been done only on positive and negative labeled sentences. After that, an equal number of samples from each class has been sampled to make a proper class-balanced dataset. These new augmented results are the final version of datasets that are used to train the models.

3.3 Data Cleaning

Data cleaning has been an important factor in this research. Two datasets needed the use of several data-cleaning techniques. The paragraph dataset has been extracted directly from the PubMed repository. These files are only available in XML format. XML files come with different types of tags. For example: <italic>, <xref>, <href> etc. These tags don't have significant value when the work is based on creating a feature set from the given sentences and later passed to the model. They affect the accuracy of the models and having more unnecessary words means more usage of time and space complexity. The paragraph dataset here is significantly bigger than the other ones mentioned in the above part of the discussion. Medical journals contain a lot of characters that are not used in regular research work. Like: α,β, γ, etc. These characters are also expressed using special tags in XML documents. Like tags they cannot be just removed because these characters have significant importance to the sentiment. After this, the CORD-19 dataset has similar characteristics. When initially released this dataset had all the XML tags and the dataset was not cleaned that can directly be forwarded to the models. As a result data cleaning has been a major part of this research work.

For cleaning, the Regex library has been used most in this research work. A particular Regex command has been proposed that removes all the typical XML tags from the sentences. However, after removing it also keeps the generic representation of the tags. Even if <xref>, </xref> tags have been removed, they still will have the citation information inside them. For example, "<xref ref-type="bibr" rid="B8">8</xref>,<xref ref-type="bibr" rid="B9">9</xref>" gets converted into " [8,9] ". This shows that the citation reference is maintained even though the tags were removed.

As a result of this process, shown is Table 3.5 the datasets became more interpretable, and the length of the sentences got reduced which improved efficiency and most importantly better overall accuracy was achieved.

3.4 Data Pre-Processing

3.4.1 Label Encoding

All the datasets used here have been converted into three labels: Neutral, Positive, and Negative. Rather than using strings as the labels as strings in this research work, all these labels were encoded into numeric values. The purpose is to normalize the labels that contain values only between 0 and n_classes-1. As a result, there was no ambiguity in labels across all the datasets. LabelEncoder library from Sklearn has been used here. Most deep learning techniques involv-

Table 3.5: Examples from cleaned Dataset

Citations with XML tags	Cleaned version
[<xref ref-type="bibr" rid="B17">17</xref>] found that a temperature deviation from 38°C was a significant variable in the multivariable logistic regression model expressing the outcome of a colic case. In the present study we did not find such a relationship when temperature deviation from 38°C was included in the model.</p>	[17] found that a temperature deviation from 38°C was a significant variable in the multivariable logistic regression model expressing the outcome of a colic case. In the present study we did not find such a relationship when temperature deviation from 38°C was included in the model.
The <italic>rpt6-1 </italic>mutant arrests cell division within one cycle at the restrictive temperature [<xref ref-type="bibr" rid="B13">13</xref>] and under normal growth conditions, we found changes in the α1- and α7-subunits of the 20S proteasome in the <italic>rpt6-1 </italic>mutant. One question that remains to be clarified is whether the observed changes are affected at the restrictive temperature. The data presented in figure <xref ref-type="fig" rid="F9">9</xref> indicate that the change in α7 in the <italic>rpt6-1 </italic>mutant grown at 25°C disappeared during the response to cell stress. Interestingly, in parallel with the disappearance, normalization of LLVY-hydrolyzing activity was observed in the absence of SDS (Table <xref ref-type="table" rid="T2">2</xref>). </p>	The rpt6-1 mutant arrests cell division within one cycle at the restrictive temperature [13] and under normal growth conditions, we found changes in the α1- and α7-subunits of the 20S proteasome in the rpt6-1 mutant. One question that remains to be clarified is whether the observed changes are affected at the restrictive temperature. The data presented in figure 9 indicate that the change in α7 in the rpt6-1 mutant grown at 25°C disappeared during the response to cell stress. Interestingly, in parallel with the disappearance, normalization of LLVY-hydrolyzing activity was observed in the absence of SDS (Table 2).
α-Sarcin has been shown to kill cells by apoptotic induction [<xref ref-type="bibr" rid="B5">5</xref>]. To determine whether the cell death we observed by direct expression was due to apoptosis, we measured apoptosis in cells transfected with α-sarcin or each of the mutants. As expected, wild type α-sarcin and both of the mutants induced apoptosis 24 hours after transfection (Figure <xref ref-type="fig" rid="F4">4c</xref>). We did not detect any difference in necrosis between pcDNA3 controls and wild-type α-sarcin or either of the mutants (data not shown).</p>	α-Sarcin has been shown to kill cells by apoptotic induction [5]. To determine whether the cell death we observed by direct expression was due to apoptosis, we measured apoptosis in cells transfected with α-sarcin or each of the mutants. As expected, wild type α-sarcin and both of the mutants induced apoptosis 24 hours after transfection (Figure 4c). We did not detect any difference in necrosis between pcDNA3 controls and wild-type α-sarcin or either of the mutants (data not shown).

Table 3.6: Encoded Labels

Original Label	Encoded Label
Negative	0
Neutral	1
Positive	2

ing artificial neural network models intuitively expect data to be numerical. This improves the performance of these models in terms of accuracy and other evaluative metrics that have been used to assess how the proposed architecture is compared to other research results. One another approach to address this task could be One-Hot Encoding. Both of these approaches serve the same purpose but there are some key differences. One-Hot Encoding creates a dummy variable for each category and adds an extra column which sometimes creates a form of redundancy to the dataset. This also leans towards another problem known as multicollinearity. If there is a relationship between independent variables then this technique can hamper the model's capability of predicting a particular label with a higher confidence value. So, the use of the LabelEncoder library seemed more viable for this research work. This library takes the label column from the dataset and uses it as a parameter of the provided fit() function. This function returns the encoded value of the labels. The Negative category got labeled as 0, Neutral as 1, and Positive as 2. It has been made sure that encoded labels are consistent among all of them. Having uniform encoded labels is also very important for transfer learning as well. All the trained models under this work have been used to test how they perform to an entirely different and unseen dataset but from the same genre. To deal with the issue of redundancy the ordinal label column has been replaced with the new encoded values.

Table 3.6 shows the new encoded values across all the datasets.

After performing label-encoding in all the datasets they were sampled in such a way they have an equal number of samples from each encoded category. With the class-balanced datasets, they have been split into train and test sets. Having the train test separation from a data frame is crucial for any neural network model. The ratio that has been maintained is 80:20 where 80 belongs to training data. The datasets were shuffled so that the models don't tend to find a pattern based on the order of the training samples. This process has been repeated five times for each dataset with a different random state value. This gave me the opportunity to assess the average performance of each model. Stratify has not been used here because all the datasets have an equal number of samples under each category.

A separate class file has been created called as SentimentData for preparing the dataset into a format that can be forwarded to the Transformer architecture-based models directly. This class takes three parameters. They are data frame, tokenizer, and max_length. The data

Table 3.7: Pre-trained Tokenizers corresponding to the Models

Model Name	Tokenizer
BERT	bert-base-cased
Bio-BERT	dmis-lab/biobert-v1.1
RoBERTa	roberta-base
Bio-RoBERTa	roberta-base
ELECTRA	google/electra-large-discriminator
ALBERT	albert-base-v2
SPECTER	allenai/specter

Table 3.8: Pre-trained Tokenizers corresponding to the Models used for paragraphs

Model Name	Tokenizer
Longformer	allenai/longformer-base-4096
Ernie-2.0	nghuyong/ernie-2.0-base-en
BigBird	google/bigbird-roberta-large

frames consist of two columns: Text and Sentiment. After taking the entire data frame text and sentiment are separated and stored separately into two different variables. Every individual word gets split from the provided text sentences and passed to the tokenizer to create a new representation of those words. There are several techniques of tokenization. For this research work, BERT-based tokenizers have been used. BERT-based tokenizers are a combination of token embedding, segment embedding, and positional embedding. This approach is much more complex and carries a lot more information about the input compared to regular LSTM. Tables 3.7 and 3.8 show a list of tokenizers corresponding to each individual model that has been used in this work.

Embeddings play a significant role in NLP-based tasks. Words cannot be directly forwarded to a model and make the model extract features and learn from it. A neural network can only work with numerical figures and so every word needs to be represented in a numerical arrangement. Word embeddings come into the role to address this situation. Embeddings are d-dimensional vectors that represent every unique index. There are many embedding libraries available like Word2Vec and Glove which return different dimensional vectors given a word. Figure 3.2 shows a simple example of indexing and embedding vectors given a sample sentence.

Every single word here is represented with a particular index value and a corresponding d-dimensional vector. BERT [10] uses a word-piece tokenizer concept that puts into account token, segment, and position embedding altogether. The default library of BERT does not mention positional embedding separately because the model learns the positional embedding by itself during the training phase. Under this approach, some words get divided into subwords

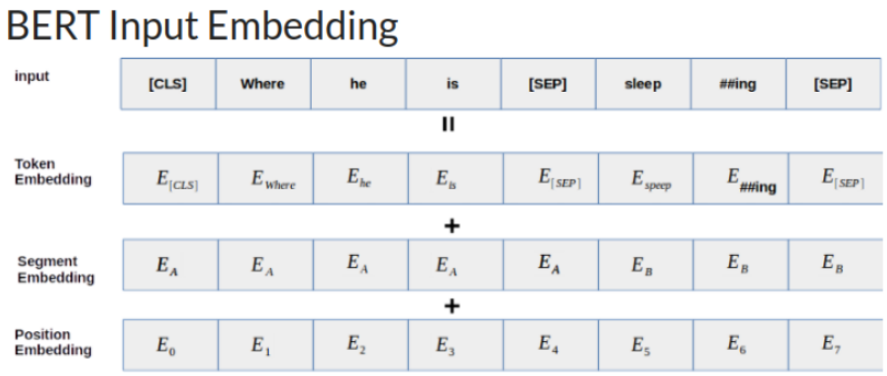

Figure 3.2: BERT Input Embedding [10]

and this helps to enrich the vocabulary because at times some unknown words can be broken down into sub-groups and these subgroups may belong to a known word for the model. As a result, the BERT tokenizer performs really well compared to other tokenization and embedding techniques. Figure 3.2 shows that all three different embeddings are part of the BERT embedding mechanism.

The third parameter is max_length. This value basically shows the longest sequence in a batch. All the inputs under one batch need to be the same. In order to ensure this consistency within a batch padding is mandatory. Not necessarily all sentences will have the same length. So the tokenizers add extra tokens according to necessity. Unless all the inputs are of the same length, BERT cannot be used to extract features that can be passed to an artificial neural network model to train on. The SentimentData class ensures this by setting pad_to_max_length=True. Padding can be applied in both possible directions, either at the of the sentence sequence or at the end. This process is applied to all the datasets. Figure 3.3 shows how padding is applied through the BERT_tokenizer to make sure all inputs are of the right format.

Now from the given texts using the BERT-tokenizer, the object of SentimentData class returns *input_ids*, attention_mask, token_type_ids, and targets as tensors. Before feeding these as inputs to the models some training and testing parameters are needed to be defined such as batch_size, shuffle, num_workers, etc. These parameters are incorporated with the object of SentimentData class using the PyTorch DataLoader library. In order to use these two abstract methods are mandatory: __getitem__ & __len__. SentimentData class already has these methods implemented so they can be inherited now in the DataLoader. This gives the opportunity to iterate through prepared data and initialize the above-mentioned parameters. This is the last step of pre-processing the data and now it is forwarded to the model architecture.

Figure 3.3: BERT Padding Format [10]

3.5 BERT Based Models

Deep learning models (such as transformers) are examples of pre-trained models (PTMs) for natural language processing. These models are trained on a huge dataset to carry out certain NLP tasks. PTMs, when trained on a large corpus, have the ability to acquire universal language representations. This has the potential to be useful for subsequent NLP tasks and eliminates the need to build a new model from the ground up. Pre-trained models can thus be referred to as reusable NLP models, which NLP developers can employ to rapidly build an NLP application using. Transformers offer a collection of deep learning NLP models that have already been pre-trained to perform a variety of NLP tasks, including text classification, question answering, machine translation, and others.

The most recent PTM, also known as the 2nd generation, has been trained to learn contextual word embeddings. This chapter demonstrates how transfer learning and unsupervised learning can benefit from pre-trained language models like BERT [10]. The goal of this study is to portray the workflow of how from the very beginning stage a text is processed and later different models can be trained on them for applying to real-life applications.

3.5.1 BERT

BERT [10] has been proposed to join conditioning from the left and right contexts in all the layers to pre-train deep bidirectional representations. This pre-trained model can easily be

fine-tuned to produce better results for various tasks.

BERT counters the main limitation of the standard language models being unidirectional which restricts the choice of the architectures that can be accessed during pre-training. This has been possible to implement in this work by using a masked language model that randomly masks some of the tokens and the goal is to predict the original vocabulary id of the masked word depending only on its context. Alongside this BERT also uses the "next sentence prediction" task that combined pre-trains text-pair representations. Pre-training and fine-tuning are the two steps that define the framework of BERT. While the model is in the pre-training phase it gets trained on unlabeled over several pre-training tasks. After that in the fine-tuning phase, the model gets initialized with the same pre-trained parameters along with all of the parameters that are fine-tuned using labeled data from downstream tasks. The pre-training corpus that has been used here is a combination of BookCorpus and English Wikipedia.

The deep bidirectional architecture that has been proposed here revolutionized the way of tackling NLP tasks. Transfer learning with language models has shown the fact that meaningful unsupervised pre-training is an inseparable part of language understanding systems. BERT using this idea performed significantly better on almost every single standard NLP task.

3.5.2 Bio-BERT

Bio-BERT is a dedicated model in natural language processing (NLP) for biomedical and clinical applications. It achieves this by addressing the challenges of applying NLP to these domains, including the complexity of biomedical terminology, the need for domain-specific knowledge, and the need for large amounts of annotated data. Several popular PLMs, including BERT, BioBERT, and ClinicalBERT can be adapted and fine-tuned for use in biomedical and clinical NLP. Bio-BERT has been trained on PubMed data and, optionally, MIMIC-III.

3.5.3 RoBERTa

This research work is similar to the study of BERT [10] pretraining given emphasis on the impact of several key hyperparameters and training data size. Claiming that BERT is significantly undertrained this model showed improved results on regular NLP tasks. This work also pointed out the previously overlooked design choices and the source of recently reported improvements. The proposed model name is RoBERTa [21] which is a robust and more optimized version of BERT.

Researchers here presented an evaluation of the impact of hyperparameters and training set size. This model is called RoBERTa and this matched or outperformed all post-BERT models. This process involved training the model longer over more data with a bigger batch

size, removing the Next Sentence Prediction (NSP) objective, training longer sequences, and dynamically changing the masking pattern applied to the training data. To ensure better control over the effects of training size researchers collected a new dataset called CC-News which is of comparable size to other similar datasets.

RoBERTa is trained based on five English-Language corpora of different domains and sizes. These corpora are BOOKCORPUS, CC-NEWS, OPENWEBTEXT, and STORIES.

3.5.4 Bio-RoBERTa

Bio-RoBERTa is a domain-specific model that involves fine-tuning the pre-trained RoBERTa model on the 2.68 million scientific articles from the Semantic Scholar database (the S2ORC dataset maintained by AllenAI) focusing on biomedical papers. For this model complete texts of the articles have been used rather than just abstracts from PubMed, so this dataset contains papers that are not publicly available due to copyright restrictions. This dataset has a total of 7.55 billion tokens and 47 gigabytes of information.

3.5.5 ALBERT

The paper introduces ALBERT [18] (A Lite BERT), a self-supervised language representation learning model that significantly reduces the number of parameters in the original BERT model while maintaining comparable or better performance on various NLP tasks. ALBERT achieves this by using three novel techniques: (1) factorized embedding parameterization, which reduces the size of the embedding matrix by factorizing it into two smaller matrices, (2) cross-layer parameter sharing, which reduces the number of parameters by allowing the parameters of all layers to be shared, and (3) Inter-sentence coherence loss, which mainly focuses modeling of inter-sentence coherence avoiding the topic prediction.

3.5.6 Longformer

Due to the quadratic scaling of self-attention with sequence length, transformer-based models struggle to process long sequences. To overcome this limitation, the Longformer model [4] has been developed with an attention mechanism that scales linearly with sequence length, enabling the processing of documents with thousands of tokens or more. Longformer's attention mechanism combines local windowed attention with task-motivated global attention and can be used as a drop-in replacement for standard self-attention.

The key innovation of Longformer is the use of a sliding window attention mechanism that only attends to a subset of the input tokens at any given time. This approach avoids the

quadratic complexity of traditional self-attention mechanisms while maintaining long-range dependencies across the entire sequence.

The authors also demonstrate the scalability of Longformer by training models on up to 8,000 tokens using a single GPU, a significant improvement over traditional Transformer models which can only handle inputs of 512 as token length.

Overall, Longformer represents a promising approach to handling long documents in natural language processing tasks, with potential applications in areas such as scientific research, legal documents, and financial reports.

3.5.7 Big Bird

This is a research paper that presents a new transformer-based model called Big Bird [37] that is designed to handle longer sequences than previous transformer models. The authors note that traditional transformer models such as BERT are limited in their ability to process long sequences due to their quadratic complexity in both the sequence length and the hidden size of the model.

To address this limitation, the authors introduce two key innovations in the Big Bird model. First, they propose a new sparse attention mechanism that enables the model to attend to only a subset of the sequence, rather than the entire sequence. This mechanism is based on a hierarchical structure that partitions the sequence into segments and applies attention only to the relevant segments. The authors show that this sparse attention mechanism reduces the computational cost of the model from quadratic to linear in the sequence length.

Second, the authors introduce a novel way of initializing the model that is based on a combination of pretraining and random initialization. The model is pre-trained on a smaller subset of the sequence, which allows it to capture global patterns in the data, and then randomly initialized on the remaining portion of the sequence. This approach allows the model to capture both local and global patterns in the data, which the authors argue is necessary for handling longer sequences.

3.6 Other Pre-trained Language Models

3.6.1 ELECTRA

Pre-training methods for masked language modeling (MLM), such as BERT, corrupt the input by replacing some tokens with [MASK] and then train a model to reconstruct the original tokens. While they produce good results when applied to downstream NLP tasks, they typically

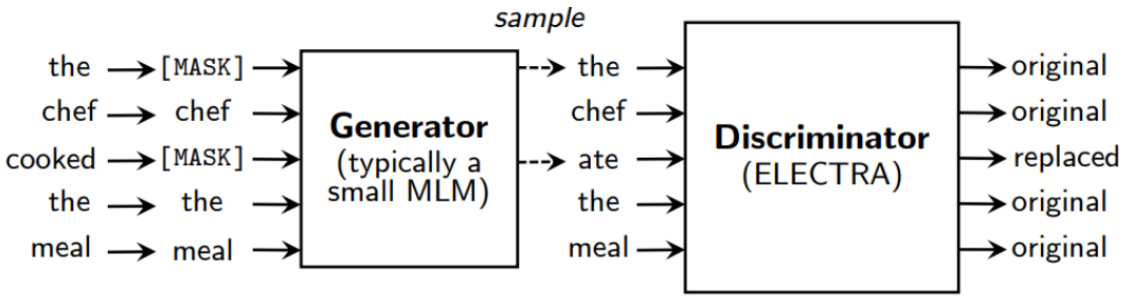

Figure 3.4: A high-level overview of replaced token detection under ELECTRA [7]

require a lot of computing power to be effective. As an alternative, researchers here proposed a replaced token detection system, a more sample-efficient pre-training task named as ELECTRA [7]. Rather than masking the input, they altered it by replacing some tokens with plausible alternatives drawn from a small generator network. Then, rather than training a model to predict the original identities of the corrupted tokens, they trained a discriminative model to predict whether or not each token in the corrupted input was replaced by a generator sample.

3.6.2 SPECTER

To counter the limitation of BERT-based models to address the information based on inter-document relatedness, authors have proposed a novel model named SPECTER [9]. SPECTER in a novel method to produce document-level embedding specializing in scientific documents based on pretraining a Transformer language model on powerful information of document-level relatedness. BERT is powerful but it only emphasizes token and sentence-level training goals. On the contrary, this new SPECTER model considers the document-level representation and can easily be applied to downstream tasks without any task-specific fine-tuning.

The researchers have particularly used citations as a naturally occurring, inter-document incidental signal showing which documents are most related and structured the signal into a triple loss pretraining objective. This model does not require any citation information during the inference.

The choice of negative paper is highly significant in terms of training this model. In order to give a more nuanced training signal they augmented the randomly drawn negatives and prepared a set called “hard negative”. Hard negatives are the papers that are not cited by the query paper but are cited by a paper cited by the query paper. Their experiments show that involving hard negatives results in more accurate embeddings. During the time of inference, a paper P is given to the model and it produces output as SPECTER’s Transformer pooled output activation. SPECTER only requires the title and abstract and does not need any citation

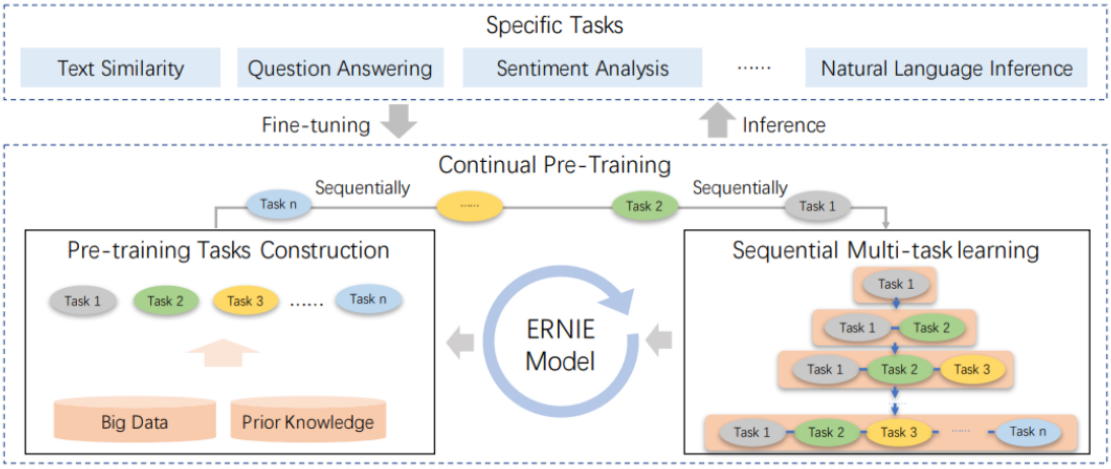

Figure 3.5: Structure of Ernie2.0 model [31]

information. This means that these embeddings can work on new papers that are yet to be cited.

The results are really promising after all the experiments done on the above-mentioned tasks. Overall, this paper has shown that average performance of 80.00 across all metrics on all tasks which is significantly better compared to the next best baseline. SGC only outperforms SPECTER on the Cite task's MAP and nDCG score.

3.6.3 Ernie 2.0

ERNIE 2.0 is a continuous pre-training framework that supports customized training tasks and continuous multi-task learning in an incremental manner. Researchers here have created three different unsupervised language processing tasks to assess the performance of the proposed model. The ongoing multi-task learning approach efficiently trains the newly introduced tasks alongside the old tasks when one or more new tasks are offered, without discarding previously learned information. On the basis of the previously trained parameters that it understood, the new framework may then incrementally train the distributed representations. The same encoding across all the tasks enables the encoding of lexical, syntactic, and semantic information and evaluation of them.

The above-mentioned architecture is built on two stages. The first is the pre-training phase and the second is the fine-tuning. The difference here compared to other pretraining processes is that it is continual. Instead of just training with a small number of pre-training objectives. To enable the model effectively learn the lexical, syntactic, and semantic representations, it might continuously introduce a wide variety of pretraining challenges. Three types of pretraining tasks have been mentioned in this research work for pre-training. They are word-aware tasks,

structure-aware tasks, and finally semantic-aware tasks.

3.7 Model Architecture

The model architecture starts with tokenizing the citation instances. Every pre-trained language model provides their own tokenizer. The tokenized inputs are then forwarded to the model which is a set of encoders. The number of encoders in the model stack also varies from model to model. These encoders, in the end, return the value of the hidden state. Hidden state values are then passed to a dense layer which has been named a pre-classifier. The output from the pre-classifier is used to pass through an activation function. For my work, I have used ReLU as the activation function. ReLU stands for Rectified Linear Unit. Under this activation function, the neurons perform a linear transformation on this input using the weights and biases. In the end, the output from the activation function is sent to the subsequent hidden layer, where the process starts all over again. The term "forward propagation" refers to the transfer of information in a forward direction. In most cases, the value that is actually being used is very different from the output that is being generated. The error is estimated by making use of the output from the forward propagation. Updates are made to the weights and biases of the neurons whenever this error value is calculated and used. Backpropagation is the name given to this particular mechanism. Before, performing the final linear transformation I have one dropout layer. The fact that the dropout layer stops all neurons in a layer from improving their weights at the same time simultaneously is the primary benefit of using this strategy. This adaptation, which is carried out in random groups, prevents all of the neurons from converging to the same goal, and as a result, the weights are decorrelated. Finally, after the dropout, the neurons go through the final linear layer with an output dimension of 3. The reason behind having 3 as the final output of the linear layer is because in all the datasets in this work have three labels. The goal is to predict the correct label from a given citation instance.

Each model has been trained on 40 epochs. Under this, I have used cross-entropy as the loss function. The concept of cross-entropy originates in the discipline of information theory. It is a measure that expands upon the concept of entropy and, in general, calculates the difference between two probability distributions. Cross-entropy can be thought of as calculating the total entropy between the distributions, in contrast to KL divergence, which calculates the relative entropy between two probability distributions. Cross-entropy is closely related to but distinct from KL divergence, which calculates the relative entropy between two probability distributions. Cross-entropy and its close relative, logistic loss, sometimes known as log loss, are two concepts that are frequently mistaken for one another. When employed as loss functions for classification models, both measures calculate the same quantity and can be utilized

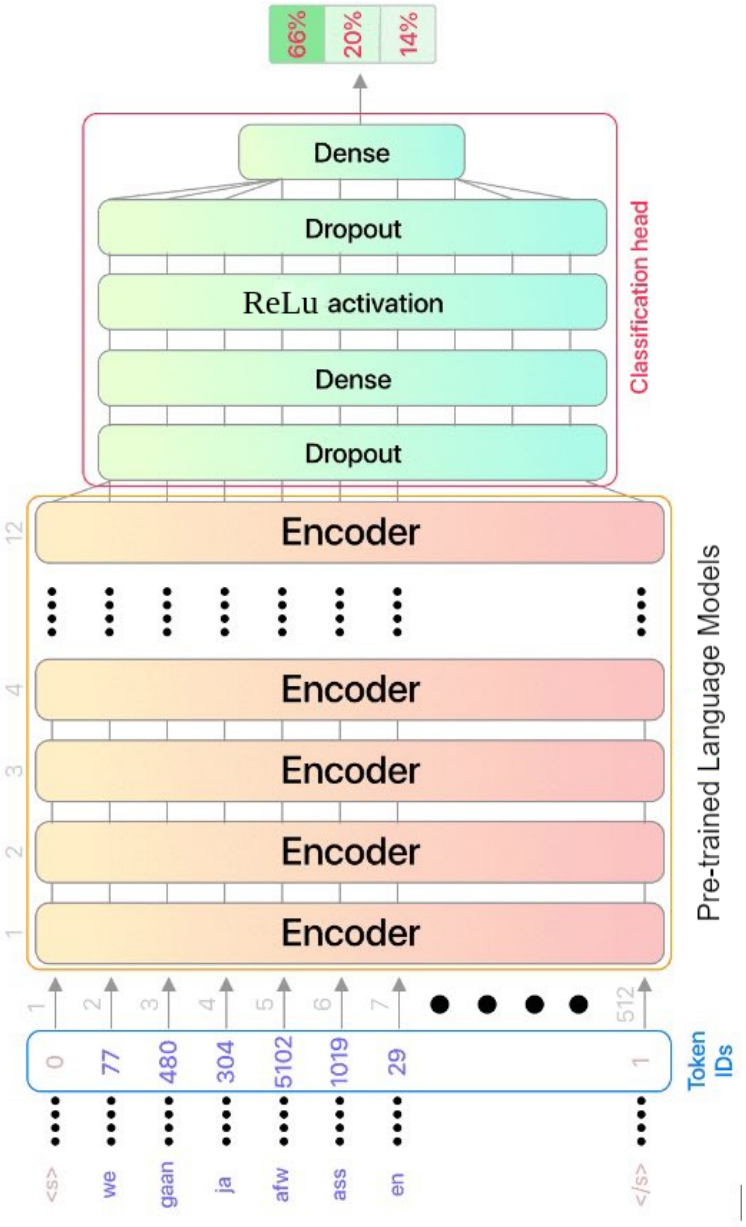

Figure 3.6: Proposed Model Architecture

interchangeably, despite the fact that the two measures are derived from separate sources [24]. Besides this, I have used Adam as the optimizer. The Adam optimizer is superior to all other optimization algorithms in several ways, including the fact that it has a shorter learning curve, a shorter calculation time, and fewer tuning parameters. Because of all of these factors, Adam is the optimizer that is advised to be used by the majority of the applications and also used for training the proposed model here in this work. [16]

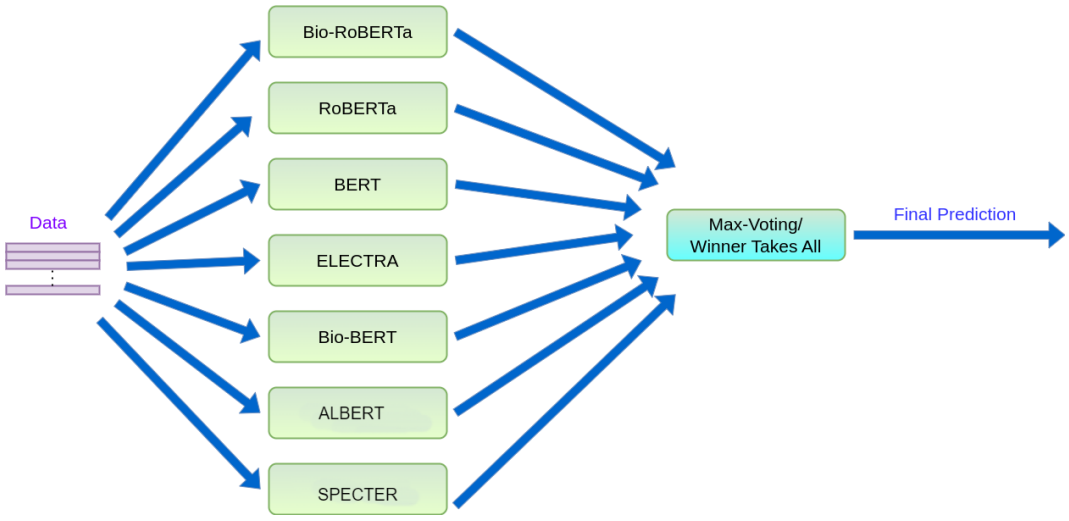

Figure 3.7: Ensemble architecture in this thesis

3.8 Ensembling

The process of combining multiple models into a single, comprehensive, and trustworthy predictive model is referred to as "ensembling". A powerful deep learning model that may be utilized in a variety of contexts is known as a convolutional neural network. On the other hand, using an ensemble of the same deep learning model produces results that are more accurate and more robust when applied to different datasets. When compared to the fundamental deep learning models, ensemble models are characterized by greater reliability and robustness. Additional comparisons can be made between the two models by applying them to larger datasets. There are more deep learning models that, in order to perform classifications, can be ensembled with either the same or distinct models.

There are several ensembling techniques. In Figure 3.7, the meta learner can be designed in multiple ways following the ensembling techniques. For this work, I have used Max-voting and Winner-take-all.

Max-voting: One of the most used and convenient methods for combining the predictions produced by different machine learning algorithms is called max-voting. This method is typically applied when dealing with classification issues. In all the datasets here there are three labels (Positive, Negative, and Neutral). So, this method has been implemented initially for this task. During the process of max-voting, every base model generates a forecast and casts a vote for each sample. Only the sample class that received the most votes is considered for inclusion in the final predictive class. For illustration's sake, let's imagine that we have one citation from a biomedical journal that can be predicted into three distinct labels. We are able to make the reasonable assumption that some models will give a label value of positive while

others may choose from negative and neutral. When determining the final label or class, if the majority of models, or more than half of them, provide a rating of a particular label, then that is the label that is returned and checked for validation.

Winner-take-all: Winner-take-all networks are an example of competitive learning in recurrent neural networks, which is a concept that is central to the theory underlying artificial neural networks. The network's output nodes mutually inhibit one another while at the same time activating themselves through reflexive connections. After a certain amount of time has passed, there will be only one active node in the output layer, and that node will be the one that corresponds to the input that is the strongest. Therefore, the network employs nonlinear inhibition in order to select the most significant input from among a group of candidates. Winner-take-all is a universal computational primitive that can be implemented using several kinds of neural networks, such as continuous-time networks, as well as spiking networks [22, 25]. In computer models of the brain, winner-take-all networks are frequently employed, notably for distributed decision-making or action selection in the cortex. In this work, I have used the output prediction of different models replacing the idea of neurons with them. I have used several language models to predict correct labels on all the datasets. Every single model predicted a particular label for a given citation example with a different probability value also can be called confidence. In contrast to the max voting approach, I didn't consider the max number of votes for a prediction but gave emphasis to the model which identified a label with more confidence. Several models can predict something the same with less confidence but a single model might predict a different label with much more confidence and that can match the target efficiently resulting in better performance.

Chapter 4

Experimental Setup and Results Analysis

This chapter outlines the experiments that I conducted for my thesis project. The two previous works that researched citation polarity using gold standard datasets looked at two text genres: biomedical text [15] and computational linguistic text [2]. Both of these works used traditional machine-learning techniques. My motivation was to investigate deep learning techniques for citation polarity using these two gold-standard datasets. In addition, I was interested in working with the CORD-19 silver standard corpus.

4.1 Description of Experiments

As a part of our research work, we opted for several experiment setups and used various techniques for initializing and using the models. The experiments are described as below:

- Experiment 1: We trained all the models based on Jia's dataset and Athar's dataset and tested on the unseen samples from the original datasets, respectively. The motivation behind this setup is to see how pre-trained language models perform on the original datasets and compare them to the results mentioned in the actual papers. All the parameters during the training were the same so that the results remain consistent.

- Experiment 2: Under this setup, we used the augmented datasets which were created using paraphrasing, downsampling, and class-balancing techniques. This time the models were trained using these new proposed datasets but were tested on the same unseen sample from Experiment 1. The goal is to evaluate if having a class-balanced dataset contributes to the accuracy of the models or not.

- Experiment 3: A merged dataset combining the augmented datasets of Jia and Athar has been used here for training the models. Previously, the experiments were done on an

individual dataset but this time the approach is to see if merging data from two different sources belonging to different genres improve the prediction capability of the models or not. This time the models were tested on Jia's original dataset.

- Experiment 4: We used the new paragraph dataset which we created from Jia's work to train the models that work well with long sequences of sentences. The incentive behind this is to see if having neighboring sentences merged into a paragraph contains more information about the polarity of a citation rather than just a single sentence.

- Experiment 5: This time we used one dataset to train the models and tested it on an entirely different dataset to explore the potential of transfer learning. For all three datasets, we did this experiment to show how well they perform on each other and used the same models for training so that the new results can be compared to the experiments mentioned earlier.

4.2 Parameters and Resources

I have used multiple tokenizers to create the word embeddings. Table 3.7 in Chapter 3 shows all the pre-trained models and their corresponding tokenizers that have been used. The maximum length of a sentence here has been set to 512 for all the models since this is a limitation of all BERT-like models. The batch size is fixed to 4 for both the training and validation datasets. The learning rate is a crucial factor in optimizing a neural network. It dictates the momentum and helps the optimizer to perform efficiently. The learning rate has been set to 1e-05 for all the experiments. All models have been trained for 40 epochs using the NVIDIA RTX A6000 GPU having 48 gb ram available.

4.3 Results and Evaluation

4.3.1 Sentence-based Datasets

After training the models separately on Jia's [15] dataset, Athar's [2] dataset, a merged dataset composed of Jia's and Athar's dataset, and the CORD-19 [35] dataset I have obtained the following results.

First, Table 4.1 shows the results based on the initial dataset proposed by Jia [15]. SPECTER and both of the ensembling methods produce an accuracy of 0.86. However, in this work, I have focused on precision, recall, and F1-scores as well. This is where this initially proposed dataset lacks since it's highly class-imbalanced. In case of BERT and ELECTRA we observe that these

Table 4.1: Result of training and testing on Jia's Original dataset

Model Name	Label	Precision	Recall	F1-score	Accuracy
BERT	0	0.00	0.00	0.00	0.69
	1	0.87	0.76	0.81	
	2	0.35	0.66	0.46	
Bio-BERT	0	1.00	0.10	0.18	0.82
	1	0.88	0.91	0.89	
	2	0.57	0.69	0.62	
RoBERTa	0	0.57	0.80	0.67	0.83
	1	0.92	0.84	0.88	
	2	0.64	0.79	0.71	
Bio-RoBERTa	0	0.67	0.60	0.63	0.84
	1	0.93	0.86	0.89	
	2	0.63	0.83	0.72	
ELECTRA	0	0.00	0.00	0.00	0.75
	1	0.75	1.00	0.86	
	2	0.00	0.00	0.00	
ALBERT	0	0.20	0.30	0.24	0.74
	1	0.83	0.91	0.87	
	2	0.85	0.38	0.52	
SPECTER	0	0.55	0.60	0.57	0.85
	1	0.94	0.88	0.91	
	2	0.69	0.83	0.75	
Max Voting	0	0.83	0.50	0.62	0.86
	1	0.89	0.93	0.91	
	2	0.71	0.69	0.70	
Winner Takes All	0	0.80	0.40	0.53	0.86
	1	0.89	0.95	0.92	
	2	0.77	0.69	0.73	

models could not detect negative label as the precision, recall and F1-score value is 0.00. Both Bio-BERT and Bio-RoBERTa outperfom their base models in terms of all the matrices.

Table 4.2 shows the result where the model was trained on class-balanced Jia's dataset and it was also tested on the same dataset as Table 4.1 but with unseen citation instances. From the pre-trained models, SPECTER showed the highest overall accuracy of 0.9225. In terms of the F1-score, for label 0, BERT performs the best with a score of 0.95 and SPECTER produces the best score for labels 1 and 2. In this scenario BERT outperforms Bio-Bert but Bio-RoBERTa performed significantly well compared to base RoBERTa. One interesting finding for ELECTRA under this setup is that the accuracy went down compared to Table 4.1 whereas precision, recall and F1-score for all three labels improved. This shows that class-balancing helps the models to learn all three labels better and produce better polarity detection. The ensembling

Table 4.2: Result of training on Jia's Class-balanced dataset and testing on Jia's Test dataset

Model Name	Label	Precision	Recall	F1-score	Accuracy
BERT	0	0.91	1.00	0.95	0.9161
	1	0.95	0.94	0.94	
	2	0.79	0.79	0.79	
Bio-BERT	0	0.67	1.00	0.80	0.9096
	1	0.96	0.92	0.94	
	2	0.86	0.83	0.84	
RoBERTa	0	0.77	1.00	0.87	0.856
	1	0.94	0.93	0.94	
	2	0.81	0.76	0.79	
Bio-RoBERTa	0	0.71	1.00	0.83	0.9032
	1	0.93	0.94	0.94	
	2	0.88	0.72	0.79	
ELECTRA	0	0.10	0.50	0.17	0.5354
	1	0.75	0.66	0.70	
	2	0.25	0.03	0.06	
ALBERT	0	0.67	1.00	0.80	0.8516
	1	0.91	0.90	0.90	
	2	0.69	0.62	0.65	
SPECTER	0	0.71	1.00	0.83	0.9225
	1	0.95	0.95	0.95	
	2	0.92	0.79	0.85	
Max Voting	0	0.83	1.00	0.91	0.9161
	1	0.93	0.96	0.94	
	2	0.88	0.72	0.79	
Winner Takes All	0	0.83	1.00	0.91	0.9225
	1	0.94	0.96	0.95	
	2	0.88	0.76	0.81	

techniques even push the performance of these models. Using the Winner-Takes-All mechanism the highest average accuracy of 0.9225 has been achieved as well as this technique has the most impactful F1, precision, and recall value for all three labels compared to the other models.

After that, Table 4.3 shows the result where we trained the model on Athar's original dataset and also tested the model on the unseen test data from the same corpus. RoBERTa and Bio-RoBERTa both produce the max accuracy of 0.89 but the F1-score for each label here is not satisfactory. The main reason behind this is that the dataset is hugely class imbalanced. ELECTRA gave an accuracy of 0.88 which is pretty well but it is only detecting one label and that is neutral. If we look at the F1-score, for labels 0 and 2 the value is 0.00. This means the model is labeling everything as neutral and since the dataset is skewed toward neutral labels we got such

Table 4.3: Result of training and testing on Athar's Original dataset

Model Name	Label	Precision	Recall	F1-score	Accuracy
BERT	0	0.38	0.09	0.14	0.87
	1	0.95	0.92	0.94	
	2	0.39	0.63	0.48	
Bio-BERT	0	0.46	0.32	0.38	0.89
	1	0.94	0.95	0.95	
	2	0.52	0.49	0.51	
RoBERTa	0	0.53	0.38	0.44	0.891
	1	0.95	0.94	0.94	
	2	0.48	0.62	0.55	
Bio-RoBERTa	0	0.44	0.39	0.42	0.894
	1	0.95	0.94	0.95	
	2	0.51	0.61	0.56	
ELECTRA	0	0.00	0.00	0.00	0.88
	1	0.89	1.00	0.94	
	2	0.00	0.00	0.00	
ALBERT	0	0.22	0.39	0.28	0.875
	1	0.94	0.94	0.94	
	2	0.58	0.37	0.45	
SPECTER	0	0.38	0.48	0.43	0.88
	1	0.96	0.93	0.94	
	2	0.50	0.60	0.55	
Max Voting	0	0.50	0.38	0.43	0.907
	1	0.95	0.96	0.95	
	2	0.58	0.54	0.56	
Winner Takes All	0	0.48	0.21	0.30	0.906
	1	0.94	0.97	0.95	
	2	0.58	0.51	0.54	

high accuracy which actually does not contribute to the quality of polarity detection. Just like Jia's original dataset both Bio-BERT and Bio-RoBERTa outperform their base models in terms of all the matrices. We found a pattern that for augmented datasets the base models perform better than the bio-based ones and vice-versa. Max voting ensembling technique produced the best result with an accuracy of 0.907 but still, it was lacking in terms of precision, recall, and F1-score for negative and positive labels.

Then, we have included Table 4.4 where we trained the models on Athar's class-balanced dataset with paraphrases and tested on the same dataset as used in Table 4.3. This table shows that even though the highest accuracy is 0.821 from the Max Voting technique, the F1-scores for each label have improved significantly. RoBERTa and Bio-RoBERTa performed really well as stand-alone models with an accuracy of 0.816 and 0.814 respectively. Even though

Table 4.4: Result of training on Athar's Class-Balanced dataset and testing on Athar's Test dataset

Model Name	Label	Precision	Recall	F1-score	Accuracy
BERT	0	0.78	0.82	0.80	0.754
	1	0.81	0.78	0.80	
	2	0.67	0.67	0.67	
Bio-BERT	0	0.76	0.78	0.77	0.737
	1	0.82	0.76	0.79	
	2	0.64	0.67	0.66	
RoBERTa	0	0.76	0.89	0.82	0.816
	1	0.88	0.83	0.85	
	2	0.84	0.73	0.78	
Bio-RoBERTa	0	0.79	0.86	0.82	0.814
	1	0.97	0.77	0.86	
	2	0.73	0.81	0.77	
ELECTRA	0	0.36	0.36	0.36	0.356
	1	0.35	0.56	0.43	
	2	0.39	0.15	0.22	
ALBERT	0	0.83	0.67	0.74	0.74
	1	0.73	0.81	0.77	
	2	0.68	0.74	0.71	
SPECTER	0	0.74	0.88	0.80	0.772
	1	0.86	0.73	0.79	
	2	0.73	0.71	0.72	
Max Voting	0	0.78	0.88	0.83	0.821
	1	0.90	0.84	0.87	
	2	0.79	0.74	0.76	
Winner Takes All	0	0.78	0.81	0.79	0.792
	1	0.87	0.84	0.86	
	2	0.73	0.72	0.72	

ELECTRA still is unable to predict the labels correctly, the performance improved after class-balancing the dataset and we see almost similar F1-score values for all three labels. The goal of this research work is to propose a mechanism that can detect each label with almost the same accuracy. In the original work, Athar [2] showed that by using dependencies and negation he received the highest average *macro*-F1 value of 0.764. According to my approach using the max voting ensembling technique, I got the *macro*-F1 value of **0.82** which is **5.6 percentage points** higher than the original reported result.

Results mentioned in Table 4.5 show that the highest average accuracy on the down-sampled yet class-balanced COVID dataset has been achieved with the Max-voting ensembling technique with a value of 0.9521. Even though Max-voting produces the overall best precision,

Table 4.5: Result of training on the Class-balanced CORD-19 dataset and testing on the Original CORD-19 Test dataset

Model Name	Label	Precision	Recall	F1-score	Accuracy
BERT	0	0.92	0.96	0.94	0.934
	1	0.97	0.93	0.95	
	2	0.92	0.91	0.91	
Bio-BERT	0	0.93	0.97	0.95	0.945
	1	0.98	0.92	0.95	
	2	0.92	0.95	0.93	
RoBERTa	0	0.91	0.97	0.94	0.929
	1	0.97	0.88	0.93	
	2	0.91	0.94	0.92	
Bio-RoBERTa	0	0.94	0.98	0.96	0.947
	1	0.98	0.92	0.95	
	2	0.92	0.95	0.94	
ELECTRA	0	0.53	0.39	0.45	0.4
	1	0.38	0.78	0.51	
	2	0.28	0.21	0.13	
ALBERT	0	0.93	0.93	0.93	0.911
	1	0.89	0.94	0.91	
	2	0.91	0.86	0.89	
SPECTER	0	0.93	0.98	0.96	0.946
	1	0.98	0.92	0.95	
	2	0.92	0.94	0.93	
Max Voting	0	0.94	0.98	0.96	0.9521
	1	0.98	0.93	0.96	
	2	0.93	0.95	0.94	
Winner Takes All	0	0.95	0.97	0.96	0.948
	1	0.98	0.93	0.95	
	2	0.92	0.94	0.93	

recall, and F1-scores other models like Bio-RoBERTa , Bio-BERT and SPECTER performed comparatively well, too. This time both the training and test dataset had equal number of samples from all the labels.

After this, Table 4.6 shows the result of the above-mentioned Experiment 3. We used the merged dataset to train all the models and then tested on Jia's original dataset. From the single models, RoBERTa showed the best accuracy of 0.941 as well as for the precision, recall and F1-score values. If we compare this result with Tables 4.1 and 4.2 it shows that all the models perform better when these two datasets were merged. Even though they belong to two different genres they are contributing to understanding the polarity better. Two other interesting results are: 1) for BERT, Bio-BERT, RoBERTa, Bio-RoBERTa, and SPECTER,

Table 4.6: Result of training on a Merged dataset (Athar and Jia) and testing on Jia's Test dataset

Model Name	Label	Precision	Recall	F1-score	Accuracy
BERT	0	0.77	1.00	0.87	0.9161
	1	0.96	0.92	0.94	
	2	0.81	0.86	0.83	
Bio-BERT	0	0.83	1.00	0.91	0.935
	1	0.97	0.94	0.96	
	2	0.84	0.90	0.87	
RoBERTa	0	0.83	1.00	0.91	0.941
	1	0.97	0.95	0.96	
	2	0.87	0.90	0.88	
Bio-RoBERTa	0	0.91	1.00	0.95	0.9354
	1	0.96	0.96	0.96	
	2	0.86	0.83	0.84	
ELECTRA	0	0.11	0.90	0.19	0.3612
	1	0.75	0.41	0.53	
	2	0.00	0.00	0.00	
ALBERT	0	0.83	0.67	0.74	0.74
	1	0.73	0.81	0.77	
	2	0.68	0.74	0.71	
SPECTER	0	0.71	1.00	0.83	0.9032
	1	0.95	0.92	0.93	
	2	0.82	0.79	0.81	
Max Voting	0	0.77	1.00	0.87	0.9354
	1	0.96	0.96	0.96	
	2	0.92	0.83	0.87	
Winner Takes All	0	0.77	1.00	0.87	0.9290
	1	0.96	0.95	0.95	
	2	0.89	0.83	0.86	

all negative labeled sentences were found with reasonable precision, and 2) for all models (except ALBERT) the positive citations were the most difficult to classify. One may speculate that negative citations have a few linguistic styles that are easy to detect, whereas positive citations have less discernible or many more types of features. The results from Table 4.5 would suggest that the latter may be the case since with many more positive class training examples in the class-balanced CORD-19 dataset, the positive class was easier to classify. The results for ELECTRA show that this time it failed to detect positive labels entirely. The training process of ELECTRA is fundamentally different from the other models since it uses a method to corrupt the input while training instead of concealing it by swapping out some input tokens for believable substitutes drawn from an SGN. This contributed to producing embeddings that

Table 4.7: Result of Transfer Learning: Training on Athar's dataset and Testing on Jia's dataset

Model Name	Label	Precision	Recall	F1-score	Accuracy
BERT	0	0.71	0.39	0.51	0.512
	1	0.45	0.78	0.57	
	2	0.51	0.39	0.44	
Bio-BERT	0	0.58	0.61	0.60	0.487
	1	0.41	0.42	0.42	
	2	0.46	0.44	0.45	
RoBERTa	0	0.56	0.27	0.37	0.45
	1	0.43	1.00	0.60	
	2	0.42	0.14	0.21	
Bio-RoBERTa	0	0.73	0.37	0.49	0.475
	1	0.40	0.74	0.52	
	2	0.48	0.34	0.40	
ELECTRA	0	0.53	0.39	0.45	0.4
	1	0.38	0.78	0.51	
	2	0.28	0.08	0.13	
ALBERT	0	0.50	0.08	0.14	0.4
	1	0.37	1.00	0.54	
	2	0.56	0.17	0.26	
SPECTER	0	0.64	0.80	0.71	0.631
	1	0.61	0.62	0.61	
	2	0.64	0.49	0.56	
Max Voting	0	0.75	0.47	0.58	0.531
	1	0.44	0.94	0.60	
	2	0.64	0.24	0.35	
Winner Takes All	0	0.70	0.41	0.52	0.506
	1	0.43	0.90	0.58	
	2	0.58	0.25	0.35	

do not perform well for this particular task.

Then, we wanted to explore transfer learning from the above-trained models. Table 4.7 shows that using Athar's dataset highest accuracy of 0.631 has been achieved by SPECTER. Compared to all the models this model produced comparatively better results. Unlike the previous results, ensemble techniques don't give the best results. Ensembling techniques show relatively poor results compared to SPECTER. The possible explanation for this scenario is that even though SPECTER is predicting more correct labels but it is doing with less confidence and other models are choosing the wrong label with a higher probability value. As a result, the proposed max voting and winner takes all approach cannot contribute much. The same scenario has been observed in Table 4.8 and Table 4.10. Here, RoBERTa performed better on the test dataset compared to the other models whereas ELECTRA surprisingly underperformed

Table 4.8: Result of Transfer Learning: Training on CORD-19 dataset and Testing on Jia's dataset

Model Name	Label	Precision	Recall	F1-score	Accuracy
BERT	0	0.83	0.37	0.51	0.537
	1	0.40	0.78	0.53	
	2	0.70	0.47	0.57	
Bio-BERT	0	0.86	0.35	0.50	0.562
	1	0.42	0.84	0.56	
	2	0.77	0.51	0.61	
RoBERTa	0	0.85	0.43	0.57	0.6
	1	0.45	0.84	0.59	
	2	0.78	0.54	0.64	
Bio-RoBERTa	0	0.82	0.35	0.49	0.568
	1	0.42	0.82	0.56	
	2	0.78	0.54	0.64	
ELECTRA	0	0.00	0.00	0.00	0.3625
	1	0.00	0.00	0.00	
	2	0.38	0.98	0.55	
ALBERT	0	0.85	0.22	0.34	0.512
	1	0.39	0.88	0.54	
	2	0.77	0.46	0.57	
SPECTER	0	0.86	0.35	0.50	0.562
	1	0.42	0.82	0.56	
	2	0.74	0.53	0.61	
Max Voting	0	0.85	0.33	0.48	0.568
	1	0.42	0.82	0.56	
	2	0.77	0.56	0.65	
Winner Takes All	0	0.85	0.33	0.48	0.556
	1	0.41	0.82	0.55	
	2	0.76	0.53	0.62	

for the neutral label as well as for negative labels having F1-score, precision, and recall of 0.00.

When we used the trained models based on Jia's dataset and tested on Athar's and COVID dataset, we see in Table 4.11 and Table 4.12 that the accuracy and other metrics are not satisfying. SPECTER showed the highest value of accuracy of 0.7309 whereas other models even including ensemble methods significantly underperformed. Meng Jia's dataset consists of around seven hundred entries and has information about citations only from biomedical journals. Whereas, Athar's dataset has information about multiple genres and this might be a reason why only training on Jia's data does not perform well on the other two datasets. Table 4.12 also shows that this time models except SPECTER struggled to predict the neutral label since label 1 has the lowest F1-score for all the models even the ensemble methods. When we merged the

Table 4.9: Result of Transfer Learning: Training on Athar's dataset and Testing on the CORD-19 dataset

Model Name	Label	Precision	Recall	F1-score	Accuracy
BERT	0	0.57	0.34	0.43	0.415
	1	0.43	0.50	0.46	
	2	0.33	0.40	0.36	
Bio-BERT	0	0.55	0.64	0.60	0.410
	1	0.29	0.14	0.19	
	2	0.33	0.46	0.39	
RoBERTa	0	0.57	0.69	0.62	0.499
	1	0.46	0.54	0.50	
	2	0.44	0.26	0.33	
Bio-RoBERTa	0	0.61	0.76	0.68	0.537
	1	0.60	0.17	0.26	
	2	0.46	0.69	0.55	
ELECTRA	0	0.31	0.42	0.36	0.329
	1	0.35	0.55	0.42	
	2	0.18	0.01	0.02	
ALBERT	0	0.59	0.41	0.49	0.439
	1	0.39	0.40	0.39	
	2	0.39	0.51	0.44	
SPECTER	0	0.57	0.77	0.65	0.514
	1	0.50	0.45	0.47	
	2	0.43	0.33	0.37	
Max Voting	0	0.58	0.74	0.65	0.496
	1	0.45	0.41	0.43	
	2	0.41	0.34	0.37	
Winner Takes All	0	0.62	0.64	0.63	0.475
	1	0.43	0.43	0.43	
	2	0.37	0.36	0.37	

augmented datasets of Jia and Athar we saw the better performance on Jia's original dataset because they complemented each other and thus the models could perform well. However, in transfer learning, we observed poor results because of the lack of similar genre samples in the datasets. Athar's dataset contains annotations from conference and journal papers in natural language processing and computational linguistics. The CORD-19 dataset is all clinical sentences related to a specific disease (COVID-19) and scenarios regarding it taken from articles from Virology, Immunology, Molecular biology, Genetics, and Intensive care medicine journals. Jia's dataset is drawn from three biomedical journals, Acta Veterinaria Scandinavica, AIDS Research and Therapy, and BMC Biochemistry. Exploring transfer learning gave us the opportunity to explore this insight that in which scenario approach works the best.

Table 4.10: Result of Transfer Learning: Training on the CORD-19 dataset and Testing on Athar's dataset

Model Name	Label	Precision	Recall	F1-score	Accuracy
BERT	0	0.93	0.10	0.19	0.363
	1	0.35	0.93	0.51	
	2	0.28	0.05	0.09	
Bio-BERT	0	0.82	0.07	0.12	0.356
	1	0.35	0.96	0.51	
	2	0.25	0.04	0.07	
RoBERTa	0	0.75	0.13	0.23	0.361
	1	0.35	0.93	0.51	
	2	0.12	0.02	0.03	
Bio-RoBERTa	0	0.84	0.12	0.21	0.368
	1	0.35	0.96	0.51	
	2	0.27	0.03	0.05	
ELECTRA	0	0.00	0.00	0.00	0.363
	1	0.47	0.21	0.29	
	2	0.35	0.89	0.50	
ALBERT	0	0.86	0.04	0.08	0.331
	1	0.33	0.94	0.49	
	2	0.07	0.01	0.01	
SPECTER	0	0.89	0.12	0.21	0.376
	1	0.35	0.98	0.52	
	2	0.33	0.03	0.06	
Max Voting	0	0.87	0.10	0.17	0.358
	1	0.34	0.96	0.51	
	2	0.20	0.02	0.04	
Winner Takes All	0	0.85	0.08	0.15	0.361
	1	0.35	0.96	0.51	
	2	0.29	0.04	0.07	

The poor transfer performance strongly indicates that different genres have different citation styles, and interestingly, biomedicine may not be a homogeneous genre from the citation style point of view. However, merging genres, as suggested by the results shown in Table 4.6 may have positive performance outcomes.

There have been some observations that were present during most of the experiments. Regular non-bio models like RoBERTa, ALBERT, and SPECTER perform well compared to the bio-models, Bio-RoBERTa, and Bio-BERT because the datasets that have been used here belong to many categories. For example, Athar's dataset contains annotations from conference and journal papers in natural language processing and computational linguistics. The CORD-19 dataset is all clinical sentences related to a specific disease and scenarios regarding it. As

Table 4.11: Result of Transfer Learning: Training on Jia's dataset and Testing on Athar's dataset

Model Name	Label	Precision	Recall	F1-score	Accuracy
BERT	0	0.15	0.29	0.19	0.170
	1	0.88	0.11	0.19	
	2	0.08	0.80	0.14	
Bio-BERT	0	0.12	0.64	0.20	0.111
	1	0.84	0.03	0.07	
	2	0.08	0.74	0.14	
RoBERTa	0	0.16	0.59	0.26	0.0944
	1	1.00	0.00	0.00	
	2	0.08	0.90	0.15	
Bio-RoBERTa	0	0.11	0.50	0.18	0.089
	1	0.89	0.01	0.01	
	2	0.08	0.84	0.15	
ELECTRA	0	0.04	0.54	0.07	0.4733
	1	0.89	0.52	0.65	
	2	0.00	0.00	0.00	
ALBERT	0	0.46	0.38	0.41	0.3811
	1	0.40	0.01	0.03	
	2	0.35	0.76	0.48	
SPECTER	0	0.17	0.52	0.26	0.7309
	1	0.90	0.80	0.85	
	2	0.05	0.07	0.06	
Max Voting	0	0.24	0.11	0.15	0.311
	1	0.25	0.03	0.05	
	2	0.33	0.80	0.47	
Winner Takes All	0	0.27	0.10	0.15	0.3341
	1	0.36	0.36	0.36	
	2	0.33	0.55	0.41	

a result, generic models generate better embeddings context-wise and it reflects in the result tables mentioned in this chapter. After that, we observed ELECTRA performed significantly poorly across all the experiments. The training of ELECTRA is different from the other BERT-based models. In BERT, some tokens are substituted for the input using [MASK], and a model is trained to recreate the original tokens. In ELECTRA, the method corrupts the input instead of concealing it by swapping out some input tokens for believable substitutes drawn from an SGN. This non-identical training reflects in the results and we come to the conclusion that ELECTRA does not perform well for these kinds of datasets and tasks.

Returning to my interest in the improved performance of the deep neural net models on Jia's gold standard annotated dataset, I have made a comparison with Jia's reported results

Table 4.12: Result of Transfer Learning: Training on Jia's dataset and Testing on the CORD-19 dataset

Model Name	Label	Precision	Recall	F1-score	Accuracy
BERT	0	0.68	0.54	0.60	0.4667
	1	0.87	0.08	0.14	
	2	0.37	0.81	0.51	
Bio-BERT	0	0.65	0.87	0.74	0.556
	1	0.00	0.00	0.00	
	2	0.49	0.83	0.61	
RoBERTa	0	0.70	0.79	0.74	0.563
	1	1.00	0.00	0.01	
	2	0.48	0.93	0.63	
Bio-RoBERTa	0	0.68	0.83	0.75	0.5669
	1	1.00	0.02	0.03	
	2	0.49	0.88	0.63	
ELECTRA	0	0.33	0.98	0.50	0.337
	1	0.54	0.04	0.07	
	2	0.00	0.00	0.00	
ALBERT	0	0.42	0.73	0.53	0.411
	1	0.40	0.01	0.01	
	2	0.41	0.52	0.46	
SPECTER	0	0.71	0.67	0.69	0.648
	1	0.73	0.50	0.59	
	2	0.57	0.79	0.66	
Max Voting	0	0.64	0.88	0.74	0.5757
	1	1.00	0.01	0.02	
	2	0.52	0.87	0.65	
Winner Takes All	0	0.68	0.80	0.73	0.5943
	1	0.87	0.15	0.26	
	2	0.51	0.86	0.64	

from her MSc thesis using an SVM classifier and summarized it here. Table 4.13 shows that we have achieved a better F1-score for all three labels compared to Jia's reported results. The ensemble model trained on the class-balanced dataset produces a 14.1% better macro average F1-score. The recall for label 0 is much higher than Jia's result. The perfect recall with good precision means the model can correctly identify this label as belonging to the proper class of interest with few false positives. Using the merged dataset we observed that the F1-score got another boost due to more label 2 examples being correctly identified, this time by the model using the RoBERTa embeddings. The performance improvement has been a 15.38% increase in the macro average F1-score.

Table 4.13: Comparison between Jia's SVM Classifier and Two Best-performing Proposed Deep Neural Net Models Tested on Jia's Test Set

Models	Labels	Precision	Recall	F1
SVM Classifier [15]	0	0.959	0.66	0.782
	1	0.806	0.9311	0.838
	2	0.825	0.630	0.714
Proposed Ensemble Model (Trained on Class-balanced Jia's Dataset)	0	0.83	1.00	0.91
	1	0.94	0.96	0.95
	2	0.88	0.76	0.81
Best Performing Model (Trained on the Merged Dataset)	0	0.83	1.00	0.91
	1	0.97	0.95	0.96
	2	0.87	0.90	0.88

Table 4.14: Result of training and testing on the Paragraph dataset

Model Name	**Label**	**Precision**	**Recall**	**F1-score**	**Accuracy**
ERNIE-2.0	0	0.51	0.97	0.67	0.595
	1	0.75	0.88	0.81	
	2	0.00	0.00	0.00	
LongFormer	0	0.51	0.96	0.67	0.601
	1	0.75	0.91	0.82	
	2	0.00	0.00	0.00	
BigBird	0	0.50	0.96	0.65	0.589
	1	0.74	0.90	0.80	
	2	0.00	0.00	0.00	
Max Voting	0	0.51	0.97	0.67	0.595
	1	0.75	0.89	0.81	
	2	0.00	0.00	0.00	
Winner Takes All	0	0.51	0.97	0.67	0.6018
	1	0.75	0.90	0.82	
	2	0.00	0.00	0.00	

4.3.2 Paragraph-based Dataset

Finally, Table 4.14 shows the results when we used the newly created dataset for training the models. Among the models that work well with long sequences of sentences, LongFormer performed comparatively better than the others with an accuracy of 0.601. Even though this model can predict neutral labels with an F1-score of 0.82, it performs significantly poorly for positive labels. For all the models we have seen positive label has a very poor precision, recall, and F1-score. A probable reason behind this might be that when we are making paragraphs and multiple sentences are involved, it becomes ambiguous between neutral and positive. As a result, models put the positives into the other two labels and thus we have observed such poor

results. In the future, if this paragraph dataset is properly annotated once again only having relevant sentences of a citation entry then we might observe better results.

Chapter 5

Conclusions and Future Work

Citations connect research articles to form a network. A citation in the context of scientific research articles refers to the source of the idea stated in the citing sentence. This thesis presented a citation polarity classification method. Citations carry a lot of value in terms of assessing the contribution of research, its effectiveness, and mostly the impact. The previous chapters demonstrate the workflow and inspiration behind this work. This final chapter summarises the thesis' contributions, identifies the study's shortcomings, and provides recommendations for future research.

5.1 Conclusions

We have worked with three different datasets that consist of citations belonging to different research areas. These datasets have been annotated and labeled into three unique classes. The main motivation has been to surpass the accuracy metrics for all these datasets using proper word embedding, pre-trained language models, and incorporating ensemble techniques. Using the data processing methods and methodology the results that have been generated reflect the fact that, the proposed models here outperform previous results from the original papers in precision, recall, and F1-score. Our proposed models have shown better results than what Jia (**15.38%** increase in macro-average F1-score), and Athar (**5.6%** increase in macro-average F1-score) mentioned in their work. We have also obtained good performance with the CORD-19 dataset. We also experimented with merging datasets and how they perform. In addition, transfer learning also showed results that justify the fact of whether learning from one corpus can carry information if tested on an entirely new dataset or not.

To conclude, the key contributions of this work are as follows:

- We have used several data augmentation techniques like paraphrasing, and downsam-

pling to keep the datasets consistent, and class-balanced, and generate improved outcomes.

- Here, the features have been prepared using a number of pre-trained language models before being sent to the suggested artificial neural network model to categorize the proper polarity labels of citation occurrences.
- To train all the models and evaluate them against Jia's real work, a merged dataset has been created that combines Jia's and Athar's augmented datasets.
- We created a new dataset that includes the paragraphs from each reference item in Jia's original work and showed the results using pre-trained long-sequence models.
- Along with transfer learning, ensemble techniques were used to combine the results from different models and create a voting system to select the most appropriate polarity of a given citation instance.

5.2 Future Work

First of all, the paragraph dataset that has been used here can be annotated in a better way. Meng Jia [15] only worked with citation sentences and for this work, the entire paragraph that belonged to the citation sentence was extracted. However, a single paragraph can have multiple citation instances. As a result, the current dataset has many duplicates and not every single sentence in the paragraph is needed to express the polarity of a citation. So, a second layer of annotation can be done to this dataset and that will generate a smaller and more meaningful paragraph corpus. Secondly, more pre-trained language models like OpenAI's GPT-3 and others can be used to assess their performance. Finally, the CORD-19 dataset was initially annotated by machine-learning models. It has a mixture of sentences and paragraphs as citations. So, a further annotated version of this dataset or other COVID-based datasets can be used to train the proposed models and observe the impact on the performance of transfer learning.

Bibliography

[1] Awais Athar. Sentiment analysis of scientific citations. Technical report, University of Cambridge, Computer Laboratory, 2014.

[2] Awais Athar and Simone Teufel. Context-enhanced citation sentiment detection. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 597–601, 2012.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[4] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *CoRR*, abs/2004.05150, 2020.

[5] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[6] Kathi Canese and Sarah Weis. PubMed: The bibliographic database. *The NCBI handbook*, 2(1), 2013.

[7] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[8] Arman Cohan, Waleed Ammar, Madeleine van Zuylen, and Field Cady. Structural scaffolds for citation intent classification in scientific publications. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language*

Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pages 3586–3596. Association for Computational Linguistics, 2019.

[9] Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S. Weld. SPECTER: Document-level representation learning using citation-informed transformers. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online*, pages 2270–2282. Association for Computational Linguistics, 2020.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[11] Eugene Garfield. "Science Citation Index"—A new dimension in indexing. *Science*, 144(3619):649–654, 1964.

[12] Mark Garzone and Robert E Mercer. Towards an automated citation classifier. In *Conference of the Canadian Society for Computational Studies of Intelligence*, pages 337–346. Springer, 2000.

[13] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 855–864. ACM, 2016.

[14] Kokou Hospice Houngbo. *Investigating citation linkage between research articles*. PhD thesis, University of Western Ontario, 2017.

[15] Meng Jia. Citation function and polarity classification in biomedical papers. Master's thesis, University of Western Ontario, 2018.

[16] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

[17] Klaus Krippendorff. Validity in content analysis. In E. Mochmann, editor, *Computerstrategien für die kommunikationsanalyse*, pages 69–112. Campus Verlag, 1980.

[18] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020*. OpenReview.net, 2020.

[19] J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174, 1977.

[20] Anne Lauscher, Brandon Ko, Bailey Kuehl, Sophie Johnson, Arman Cohan, David Jurgens, and Kyle Lo. Multicite: Modeling realistic citations requires moving beyond the single-sentence single-label setting. In Marine Carpuat, Marie-Catherine de Marneffe, and Iván Vladimir Meza Ruíz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022*, pages 1875–1889. Association for Computational Linguistics, 2022.

[21] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

[22] Wolfgang Maass. On the computational power of winner-take-all. *Neural Computation*, 12(11):2519–2535, 2000.

[23] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Yoshua Bengio and Yann LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Workshop Track Proceedings*, 2013.

[24] Kevin P Murphy. *Machine learning: A probabilistic perspective*. The MIT Press: London, UK, 2018.

[25] Matthias Oster, Rodney Douglas, and Shih-Chii Liu. Computation with spikes in a winner-take-all network. *Neural Computation*, 21(9):2437–2465, 2009.

[26] Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. Unsupervised Learning of Sentence Embeddings using Compositional n-Gram Features. In *NAACL 2018 - Conference of the North American Chapter of the Association for Computational Linguistics*, pages 528–540, 2018.

[27] Hemant Palivela. Optimization of paraphrase generation and identification using language models in natural language processing. *International Journal of Information Management Data Insights*, 1(2):100025, 2021.

[28] Richard J Roberts. Pubmed central: The genbank of the published literature, 2001.

[29] Justyna Sarzynska-Wawer, Aleksander Wawer, Aleksandra Pawlak, Julia Szymanowska, Izabela Stefaniak, Michal Jarkiewicz, and Lukasz Okruszek. Detecting formal thought disorder by deep contextualized word representations. *Psychiatry Research*, 304:114135, 2021.

[30] Ulrich Schäfer, Christian Spurk, and Jörg Steffen. A fully coreference-annotated corpus of scholarly papers from the acl anthology. In *Proceedings of COLING 2012: Posters*, pages 1059–1070, 2012.

[31] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. Ernie 2.0: A continual pre-training framework for language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8968–8975, 2020.

[32] Simone Teufel, Advaith Siddharthan, and Dan Tidhar. Automatic classification of citation function. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 103–110, 2006.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[34] Henry Voos and Katherine S Dagaev. Are all citations equal? or, did we op. cit. your idem?. *Journal of Academic Librarianship*, 1(6):19–21, 1976.

[35] Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Darrin Eide, Kathryn Funk, Rodney Kinney, Ziyang Liu, William Merrill, Paul Mooney, Dewey A. Murdick, Devvret Rishi, Jerry Sheehan, Zhihong Shen, Brandon Stilson, Alex D. Wade, Kuansan Wang, Chris Wilhelm, Boya Xie, Douglas Raymond, Daniel S. Weld, Oren Etzioni, and Sebastian Kohlmeier. CORD-19: The covid-19 open research dataset. *CoRR*, abs/2004.10706, 2020.

[36] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, 2016.

[37] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big Bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33:17283–17297, 2020.

[38] Yijia Zhang, Qingyu Chen, Zhihao Yang, Hongfei Lin, and Zhiyong Lu. BioWordVec, improving biomedical word embeddings with subword information and MeSH. *Scientific Data*, 6(1):52, 2019.

Curriculum Vitae

Name:	Souvik Kundu
Post-Secondary Education and Degrees:	B.Sc. in Computer Science BRAC University 2016 - 2019
Honours and Awards:	Western Graduate Research Scholarship 2021-2022
Related Work Experience:	Teaching Assistant The University of Western Ontario 2021 - 2022 Research Assistant The University of Western Ontario 2021 - 2022 Teaching Assistant BRAC University 2018 - 2019

Publications:

- Vladimir Araujo, Andrés Carvallo, **Souvik Kundu**, José Cañete, Marcelo Mendoza, Robert E. Mercer, Felipe Bravo-Marquez, Marie-Francine Moens, Alvaro Soto, Evaluation Benchmarks for Spanish Sentence Representations, In Proceedings of the Thirteenth Language Resources and Evaluation Conference, pages 6024–6034, Marseille, France. European Language Resources.