Multihop Factual Claim Verification Using Natural Language Prompts

Md. Mezbaur Rahman

Department of Computer Science and Engineering
Islamic University of Technology (IUT)
June, 2023.

Multihop Factual Claim Verification Using Natural Language Prompts

by

Md. Mezbaur Rahman

Supervisor
Dr. Md. Azam Hossain
Assistant Professor
Department of Computer Science and Engineering
Islamic University of Technology

MASTER OF SCIENCE
IN
COMPUTER SCIENCE AND ENGINEERING

Department of Computer Science and Engineering
Islamic University of Technology (IUT)
Board Bazar, Gazipur-1704, Bangladesh.
June, 2023.

CERTIFICATE OF APPROVAL

The thesis titled,"**Multihop Factual Claim Verification Using Natural Language Prompts**" submitted by Md. Mezbaur Rahman, St. No. 191041032 of Academic Year 2019-20 has been found as satisfactory and accepted as partial fulfillment of the requirement for the degree Master of Science in Computer Science and Engineering on

Board of Examiners:

Dr. Md. Azam Hossain — Chairman
Assistant Professor, — (Supervisor)
Department of Computer Science and Engineering,
Islamic University of Technology (IUT), Gazipur.

Dr. Abu Raihan Mostafa Kamal — Member
Professor and Head, — (Ex-Officio)
Department of Computer Science and Engineering,
Islamic University of Technology (IUT), Gazipur.

Dr. Md. Hasanul Kabir — Member
Professor,
Department of Computer Science and Engineering,
Islamic University of Technology (IUT), Gazipur.

Dr. Chowdhury Farhan Ahmed — Member
Professor, — (External)
Department of Computer Science and Engineering,
University of Dhaka, Bangladesh

Declaration of Candidate

This is to certify that the work presented in this thesis is the outcome of the analysis and experiments carried out by **Md. Azam Hossain** under the supervision of **Dr. Md. Azam Hossain**, Assistant Professor, Department of Computer Science and Engineering (CSE), Islamic University of Technology (IUT), Dhaka, Bangladesh. It is also declared that neither this thesis nor any part of it has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others have been acknowledged in the text and a list of references is given.

Dr. Md. Azam Hossain
Assistant Professor
Department of Computer Science and Engineering
Islamic University of Technology (IUT)
Date: June 14, 2023.

Md. Mezbaur Rahman
Student No: 191041032
Date: June 14, 2023.

Dedicated to my parents

Table of Contents

Page

Acknowledgement **ix**

Abstract **x**

1 Introduction **1**
- 1.1 Motivation and Scope . . . 2
- 1.2 Problem Statement . . . 3
- 1.3 Research Challenges . . . 4
- 1.4 Research Contributions . . . 4
- 1.5 Organization . . . 5

2 Background Study **6**
- 2.1 Natural Language Processing . . . 6
  - 2.1.1 Word Representation: . . . 6
  - 2.1.2 Modern Language Models: . . . 8
- 2.2 Fact Extraction and Claim Verification . . . 16
  - 2.2.1 Document Retrieval . . . 17
  - 2.2.2 Sentence Selection . . . 17
  - 2.2.3 Multihop Iterative Sentence Selection: . . . 19
  - 2.2.4 Claim Verification: . . . 20

3 Proposed Methodology **24**
- 3.1 Overview . . . 24
- 3.2 Prompt Tuning for Claim Verification: . . . 24
  - 3.2.1 Closing the Gap between Pretraing and Finetuning?: . . . 25
  - 3.2.2 Openprompt Framework: . . . 29
  - 3.2.3 Supervised Fine-Tuning Scenarios: . . . 31
  - 3.2.4 Zero-Shot Learning Scenarios: . . . 32
  - 3.2.5 Multitasking Ability of Prompt Tuned Models: . . . 32

4 Results and Discussion **34**
4.1 Datasets 34
4.1.1 Combined Train and Dev Set Creation 35
4.2 Experimental Setup 35
4.3 Discussions 36
4.3.1 Performance based on Supervised Fine-Tuning 36
4.3.2 Performance based on Zero-Shot Learning 39
4.3.3 Label Word Probing 39
4.3.4 Error Analysis 41

5 Conclusion **45**
5.1 Summary 45
5.2 Future Works 46

References **46**

List of Publications **52**

List of Figures

Page

1.1 Sample Claim Verification Example 1
1.2 A Sample Multihop Claim Verification 2

2.1 Attention of different words with “it" for the given example 10
2.2 Process of attention calculation 11
2.3 The transformer Model Architecture 12
2.4 Overall pre-training and fine-tuning procedures for BERT 14
2.5 T5 Model Framework 14
2.6 Generative Pretrained Transformer Architecture 15
2.7 ESIM model for Sentence Selection 18
2.8 Pointwise and Pairwise Sentence Retrieval System 19
2.9 Claim Verification via NSMN model 21
2.10 Three-step pipeline evidence extraction and claim verification 21
2.11 Claim Verification using GEAR framework 22

3.1 An overview of our Proposed Prompt-based Fine-Tuning Technique for the Claim Verification Task. 25
3.2 Traditional Finetuning Process of a Language Model 26
3.3 Finetuning Process of Language Models using Task Specific Data . . 27
3.4 Prompt Tuning in Zero Shot Scenario 28
3.5 Prompt Tuning in Few Shot Scenario 29
3.6 The overall architecture of OpenPrompt Framework 30
3.7 Prompt Tuned Model is a better Multitasking Model 33

List of Tables

Page

4.1 FEVER dataset split for SUPPORTED, REFUTED, NOT ENOUGH INFO classes 34
4.2 HoVer dataset split for SUPPORTED and NOT SUPPORTED classes 34
4.3 Comparison of Performance on the HoVer Dev Set 36
4.4 Comparison of Performance in Cross-Domain Generalization. To evaluate the models' performance, we combined the development set of HoVer and FEVER. 37
4.5 Comparisons of Performance in terms of Few-Shot Learning. *Acc* refers to accuracy. We used three random seeds for each of the experiments and averaged their scores. 38
4.6 Performance based on different Language Prompts used in T5-large model 38
4.7 Comparison of Zero-Shot Performance on the HoVer Dev Set 39
4.8 Results on the Dev Set with Varying Label Words 39
4.9 Sample Cases of misclassification made by T5 model. Here `NS` and `S` stand for NOT SUPPORTED class and SUPPORTED class respectively. 43
4.10 Sample Cases of misclassification made by the `text-davinci-003` model. Here `NS` and `S` stand for NOT SUPPORTED class and SUPPORTED class respectively. 44

List of Abbreviations

BERT	Bidirectional Encoder Representations from Transformers
PLM	Pre-trained Language Model
LLM	Large Language Model
T5	Text-to-Text Transfer Transformer
GPT	Generative Pre-trained Transformers
GRU	Gated Recurrent Unit
LSTM	Long Short-Term Memory
MLP	Multilayer Perceptron
RNN	Recurrent Neural Network

Acknowledgment

My gratitude goes towards the almighty Allah, the supreme ruler of the universe, for enabling me to complete the thesis for the fulfillment of the degree of Master of Science in Engineering in due time by His grace and for granting me sound health and energy to carry out this research work successfully.

I express my deepest sense of gratitude, sincere appreciation, and immense indebtedness to Dr. Md. Azam Hossain, Assistant Professor, Department of Computer Science and Engineering, Islamic University of Technology for his support, careful supervision, scholastic guidance, instructions, encouragement, and constructive criticism during the period of the research work.

I convey my profound respect and heartiest gratitude to all the faculty members and staffs of the Department of Computer Science and Engineering, Islamic University of Technology for their active cooperation and sincere help in carrying out the research work.

Last but not the least, I would like to direct all my appreciation to my beloved parents for their inspiration and endless encouragement.

Abstract

Verifying a claim/statement using facts as evidence can be challenging, especially when the evidence consists of multiple sentences, making it difficult for NLP models to understand long-range dependencies. Most of the existing datasets provide claims that can be verified by single-hop reasoning i.e., relevant evidence to support or deny the claim can be found in a single evidence source. But the task becomes substantially challenging when it is required to mine evidence from multiple sources to correctly reach a verdict about the claim. Successful methods in single-hop verification task struggle to perform at a higher level when provided with a claim that requires multihop evidence in order to be verified. In light of the success of prompt learning in various NLP applications, this thesis introduces prompt learning for the multi-hop claim verification task. Through extensive experimentation, our proposed prompt-based method, which employs manually constructed prompts, has yielded promising results. By fine-tuning language models with prompts, we have achieved an accuracy of 83.9%, along with an enhanced cross-domain generalization performance. Additionally, we conducted experiments in few-shot and zero-shot settings, which demonstrated that prompt-based methods outperformed traditional supervised learning techniques that rely on the fine-tuning paradigm. These results underscore the effectiveness of prompt learning in the realm of claim verification.

Chapter 1

Introduction

As a result of the deluge of data being produced every second, it is no longer practical to carefully verify every piece of information we encounter. Since the invention of the printing press, people have worried about the transmission of false information. However, with the advent of the Internet and the proliferation of social media, the pace of disinformation dissemination has increased dramatically, with potentially catastrophic results. This has caused automated fact-checking to become a challenging research area within the Natural Language Processing research community.

To ensure the validity of a claim or statement, the process of claim verification involves analyzing relevant evidence [1]. Since documents are composed of sentences, one approach to achieve fact-checking is to automatically determine the relationship between sentences, such as whether they support or contradict each other. For this, a certain level of language comprehension and coherence of textual units is required. As a result, the methods developed to solve the problem of fact verification belong to the larger family of Natural Language Inference (NLI) systems. A basic NLI system may not function so well when fact-checking becomes more sophisticated since long-range

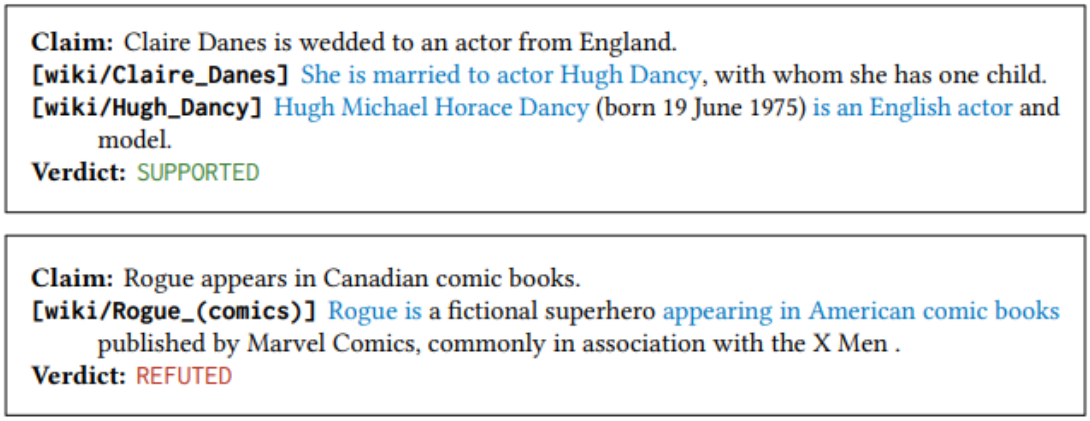

Figure 1.1: Sample Claim Verification Example. (Adapted from [1])

dependencies and the sequence of a combination of sentences come into play. This is particularly true when numerous evidence sentences are required to validate a single claim. In order for a claim verification system to do its job, it must be given both the original claim to be verified and a collection of evidence that may establish the claim as true or false. Figure 1.1 shows a sample claim verification from Wikipedia articles.

1.1 Motivation and Scope

One of the first large-scale benchmark datasets for fact extraction and claim verification is the one introduced by the FEVER shared task [2]. This task has been characterized in the literature as a combination of three subtasks: *Document Retrieval*, *Sentence Retrieval*, and *Claim Verification* [1]. In the Claim Verification module of the FEVER shared task, more than 82% of the data sample requires just one evidence sentence to predict the veracity of a claim [3]. However, in reality, it is often necessary to examine numerous facts, which may originate from numerous source documents, in order to assess the veracity of a claim making the overall verification task more challenging. In some cases, only taking into consideration of a single evidence may lead to wrong conclusion, an example case is described in the figure 1.2.

Claim: People who have recovered from COVID-19 once do not need further vaccine doses.

Evidence 1: People who have recovered from COVID-19 naturally develop immunization against it.

Evidence 2: Taking more vaccine doses for improved safety is recommended since some individuals tend to grow weak natural immunity.
Verdict: Not Supported

Figure 1.2: A sample Multihop Claim Verification

In this specific scenario, if a claim verification system solely relies on the first evidence, it could potentially make an incorrect conclusion that the claim is supported. The first evidence suggests that recovering patients develop immunity, which might lead the system to infer that the existing immunity is sufficient and additional vaccination is unnecessary. However, a more accurate understanding emerges when considering the second evidence, which states that some individuals only develop weak natural

immunity that may not provide adequate protection. Therefore, for the sake of safety, it is still advisable to administer further vaccine doses.

To overcome this challenge, a new dataset titled HoVer [4] was developed, with claim verification requiring the accumulation of multiple evidence sentences from up to four Wikipedia articles. The necessity for multi-sentence inference makes this a challenge for both the retrieval process as well as for the claim verification.

In recent years, researchers have approached the task's final module—claim verification—as a sequence classification problem and achieved state-of-the-art performance via fine-tuning pre-trained language models (PLMs) [3, 5, 6]. However, this pre-training and fine-tuning paradigm of language models require large task-specific datasets for the downstream tasks due to the difference between the pre-training and fine-tuning objectives [7]. Very recently, a new approach to pre-train language models has emerged that leverages natural-language prompts along with task demonstrations as context, keeping the objective of the downstream task similar to the pre-training task [8–11]. This is achieved via appending an additional piece of text known as a prompt to the input sample. This new method of prompt-based language model training has outperformed the widely used language model pre-training and task-specific fine-tuning approaches, along with significantly better performance in few-shot and zero-shot learning scenarios [7–11].

1.2 Problem Statement

Based on the discussion above, this research aims to develop a system that can accurately verify a given claim by learning multihop reasoning between several evidence sentences with the help of enabling the prompt learning technique in state-of-the-art language models.

The specific objectives of this research are:

1. Formulating the claim verification task to a masked word prediction problem in order to harness the vast knowledge base of state-of-the-art PLMs (Pre-trained Language Models).

2. Filtering out the correct template prompt to use as input to the PLMs and also searching for the best label words for different classes.

3. Training a robust architecture capable of capturing long-range dependencies between the evidence chains and correctly determining the veracity of the target claim.

1.3 Research Challenges

Creating a prompt-based claim verification system poses various challenges. Firstly, the language models used for input must be reconfigured to effectively comprehend the given prompts. Additionally, to leverage the vast pre-existing knowledge of these models, the prompts should be organic and similar to original English passages, as the models have been pre-trained on such articles.

Once a prompt is provided, the language model will produce one or multiple words in place of a masked token. The subsequent step is to determine the correct category of the original input claim using the predicted word. This involves selecting the most suitable label words for each category, which demands careful experimentation and filtering. Ultimately, the system must be robust enough to perform well in claim verification across various domains, rather than being limited to excelling in a single dataset.

1.4 Research Contributions

The key contributions of this research can be summarized as follows:

- We introduce the prompt learning technique in the domain of fact extraction and claim verification. Our proposed approach effectively harnesses extensive knowledge from PLMs, resulting in a huge performance gain in the veracity prediction of claims.
- We conduct extensive experiments with our proposed approach to demonstrate its substantial generalization potential in a cross-domain scenario as well as in few-shot learning scenarios. Moreover, we investigate the best natural language template for the task of verifying claims in a prompt learning setup.
- We assess the zero-shot capability of ChatGPT[1]-like model (i.e., `text-davinci-003`[2]) from OpenAI, compared to other transformer architectures, i.e., BERT, T5 etc., to set a new benchmark for zero-shot evaluation on the claim verification task. We further analyze the quality of the responses provided by `text-davinci-003` (the most competent GPT-3 model to date) to study its potential biases and hallucinations, as well as the generation of misleading information, etc.

[1] `https://openai.com/blog/chatgpt/`
[2] `https://platform.openai.com/docs/models/overview`

1.5 Organization

The rest of the dissertation is organized as follows. Chapter 2 discusses the background and motivation for gait research. It also identifies the problems persistent in the existing literature. Chapter 3 presents a new gait recognition pipeline that is able to utilize multi-scale, multi-stream features while avoiding overfitting. Chapter 4 analyzes the performance of the proposed pipeline and compares it with other state-of-the-art systems. Chapter 5 concludes our discussion and provides direction for future research scope.

Chapter 2

Background Study

In this section, we first review the hisotorical developement of Natural Language Processing(NLP) and claim verification as a NLP task , followed by reviewing the recent studies on prompt-based language models.

2.1 Natural Language Processing

The interpretation of human or natural language by a computer program is the focus of the subfield of Artificial Intelligence known as Natural Language Processing, or NLP for short. It is a subfield of linguistics that investigates the intersection between data science and the study of human language. Today, natural language processing (NLP) is prospering as a consequence of significant advances in data availability and computer capability. NLP can aid us with a broad variety of activities, and the domains of application appear to be developing on a daily basis. For example: text classification, text segmentation, Named-Entity recognition, sentiment analysis, fact verification, detection of illness, question aswering, machine translation, text summarization etc. When we perform NLP on a text,initially it turns the input raw data into tokens which we call tokenization, and then the tokens go through a number of procedures such as stop word removal, lemmatization etc. After that, we take use of several word embedding approaches in order to extract features from the processed data. After that, it monitors the integrity of our input data by using trained pipelines. Finally, following assessment we deploy the related job of NLP.

2.1.1 Word Representation:

Word representation, also known as word embedding, is the process of converting text into numerical representations that machine learning algorithms and deep learning architectures can interpret. Several techniques have been developed to achieve word

embedding, each with its own characteristics and applications. This section provides a brief overview of some popular word embedding techniques.

- Count Vector: The method known as the count vector technique requires the construction of a matrix that illustrates the number of times each word appears in a given text. The count vector matrix has dimensions of D by T, where D is the number of documents and T is the size of the vocabulary. The dimensions of the count vector matrix. Although this method is straightforward, it does not convey the semantic meaning of the words in an appropriate manner.

- TF-IDF vectorization: This method re-weights the count features by taking into account the number of times a word appears in a single document and the number of times it appears in the whole corpus. In order to provide representations that are richer in meaning, this method takes into consideration both the term frequency (TF) and the inverse document frequency (IDF).

- Continuous Bag of Words (CBOW): this model was introduced by [12], and focuses on predicting the target word by estimating its likelihood given the context. CBOW takes either a single word or a set of words from the context and uses them to predict the target word. This approach simplifies the prediction task by leveraging the context information to generate accurate word representations. CBOW is particularly useful for applications where the context plays a significant role in determining the meaning of a word.

- Skip-Gram model: this model was also introduced by [12], and takes a different approach by attempting to predict the context given a word. It aims to generate word representations that capture the relationships between words in terms of their contextual usage. By predicting the surrounding context words based on a given target word, the Skip-Gram model effectively captures words' semantic meaning and syntactic patterns.

- GloVe: the GloVe model proposed by [13] takes a different perspective on word embedding. It utilizes co-occurrence statistics of words in a corpus to generate vector representations. GloVe captures the semantic relationships between words and creates meaningful word representations by considering the global statistics of word co-occurrences. This technique enables the identification of similar words based on their distributional patterns and facilitates understanding semantic relationships between words.

2.1.2 Modern Language Models:

The field of Natural Language Processing (NLP) has witnessed significant advancements in recent years, leading to the development of more sophisticated and powerful models. These models have revolutionized the way we process, understand, and generate natural language text. The evolution of language models can be decribed in following sequences:

Neural Language Models: The neural language model employs a neural network architecture to effectively model language patterns. Neural networks represent a machine learning approach inspired by the intricate functioning of neurons in the human brain. They consist of interconnected layers of neurons that are specifically designed to process complex data and extract meaningful information. In 2003, Bengio et al. [14] proposed "A Neural Probabilistic Language Model," which demonstrated a notable 10-20% improvement in performance compared to the Trigram algorithm, as assessed by perplexity score. Perplexity serves as a measure of the model's ability to predict a sequence of words accurately. This seminal work marked a significant advancement in the field of language modeling. Subsequently, the development of the Recurrent Neural Network (RNN) emerged as a pivotal phase in neural network language modeling. RNNs have become the prevailing approach for sequence-to-sequence (seq2seq) tasks, finding applications in various domains such as natural language processing (NLP), computer vision, and speech recognition. Notably, variants of RNNs [15], such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) [16], have garnered widespread adoption due to their effectiveness in capturing sequential dependencies and mitigating the vanishing gradient problem. These advancements, collectively referred to as deep learning, have transformed the landscape of NLP and other related fields. Deep learning techniques, powered by neural networks, have become the de facto standard for tackling complex tasks in language processing, computer vision, and speech recognition. Their ability to learn hierarchical representations and model intricate patterns has significantly elevated the performance and capabilities of systems operating in these domains.

In summary, the neural language model, based on neural network architectures, has revolutionized the field of NLP. The transition from traditional approaches to deep learning, facilitated by techniques such as RNNs with LSTM and GRU, has significantly enhanced the understanding and processing of language, leading to substantial advancements in various applications.

Transformer Models: When it comes to the processing of sequence data, recurrent neural networks (RNNs) [15] are absolutely necessary since these networks make it

possible for the previous timestamp output of the network to be used as the input for the upcoming processing step. Sequence-to-sequence processing, also known as seq2seq processing, is one of the most important tasks in the field of natural language processing (NLP). In this kind of processing, the input is a series of words or frames, and the output is also a sequence of words or frames. Advanced RNN variations that include attention mechanisms, such as Long Short-Term Memory (LSTM) [17] and Gated Recurrent Units (GRU) have been used to address such issues. LSTM [17] and GRU are two examples. However, these designs have difficulties in maintaining the audience's interest across lengthy sequences, and they often struggle to come up with fresh phrase combinations. In addition, the sequential structure of these recurrent units makes parallelization difficult, which in turn causes the training process to move at a snail's pace.

The next big breakthrough came with the introduction of Google's transformer architecture. [18] The problems that were highlighted before that were linked with GRU and LSTM-based designs may now be solved thanks to the newly suggested Transformer architecture. If there are enough resources available, the Transformer design can provide attention over extended word sequences, enabling it to forecast the importance between words. Transformer architecture proposes three new ideas: Self Attention, Multi Headed Attention and Positional Embedding.

Self Attention: Self-attention, despite its name, functions in a somewhat different way than other types of attention. Self-attention, in its most basic form, is characterized by the capacity to permit the examination of all words included inside a particular phrase. In the context of this discussion, "attention" refers to the degree to which a certain target word is relevant to each individual word in the phrase. The computation of the relevance score for each word in relation to the target word is made easier by self-attention. Consider the following line as an illustration of this idea: "The animal didn't cross the street because it was too tired." The pronoun "it" raises the issue of which component of the phrase it relates to, either "The animal" or "the street." The answer to this question is not immediately clear. We are given a score for each word in the phrase by self-attention, which indicates how relevant each word is to the target word. The attention scores that were acquired following training are shown in Figure 2.1, with an emphasis placed on the words that were given the most attention. The score for "The animal" that represents the greatest level of attention is shown in the Figure 2.1.

Figure 2.2 provides a more in-depth illustration of the method used to generate attention ratings. It illustrates the participation of three separate vectors—the Query, the Key, and the Value—in the process. The Query, Key, and Value vectors are obtained by multiplying the word embedding by one of three weight vectors (WQ, WA, or WV),

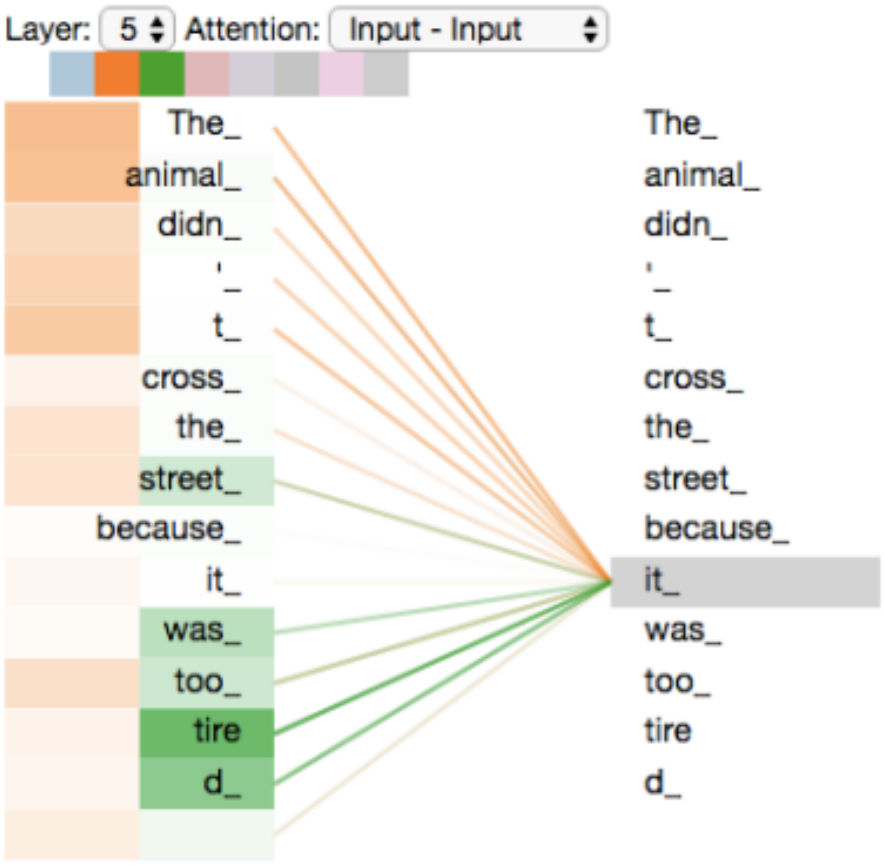

Figure 2.1: Attention of different words with “it" for the given example

and the results are then sorted in descending order. A dot product operation is applied to the Query and Key vectors before calculating the score for each word.

$$Score_t = Q_t * k_t \tag{2.1}$$

The soft-max over each of the words is being computed right now. In order to stop the gradient from exploding, the value of the soft maximum is then divided by the square root of the dimension of the key vector. After that, the value of the softmax is multiplied with the value that is determined for each word, and the results are totaled. At that particular pivot point, the total represents the output of the self-attention layer.

$$Score_t = softmax(\frac{Q * K^T}{\sqrt{d_k}}).V \tag{2.2}$$

where, Q, K and V are the query, key and value vector respectively. Then the output of self attention layer is sent to feed forward neural network.

Multi Headed Attention: Instead of using a single module for each word, the de-

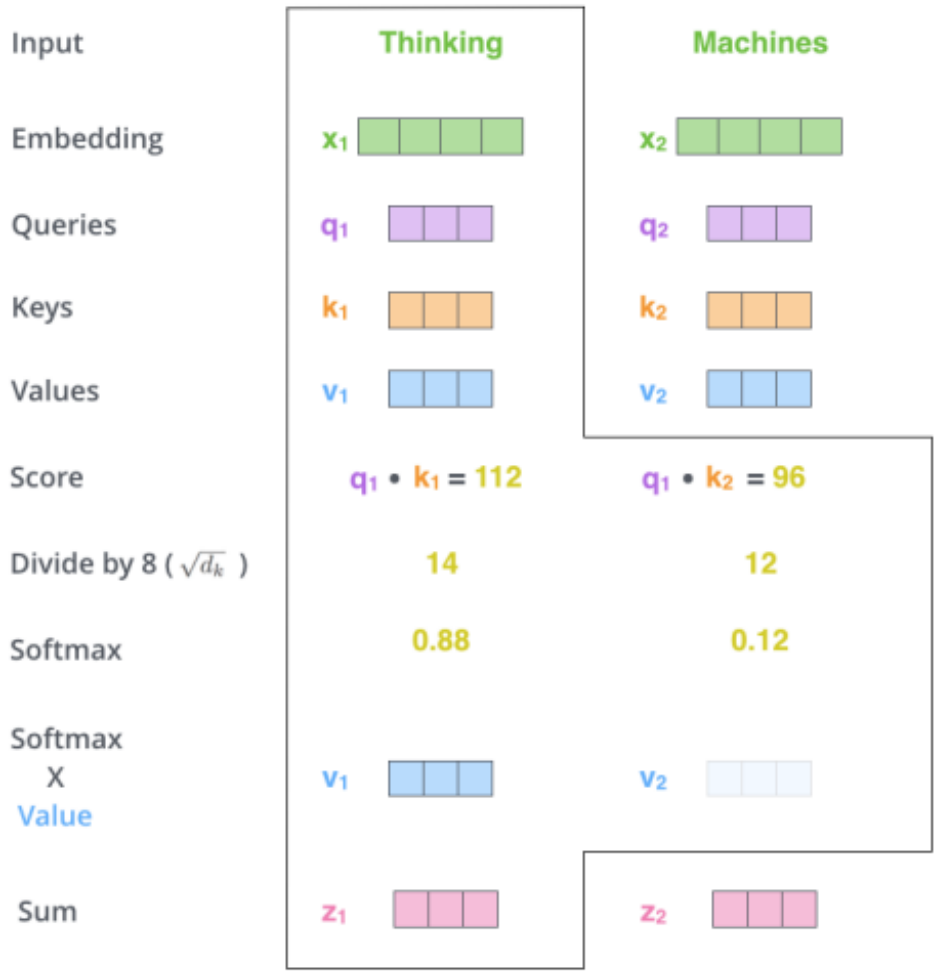

Figure 2.2: Process of attention calculation

velopers of the Transformer architecture came up with a novel strategy that included using several self-attention modules. This was done in place of the traditional method of using a single module for each word. When the outputs of each self-attention layer are concatenated, this innovation produces representations that are more accurate, as the authors showed in their work. Figure 2.1 illustrates the output for the word "it" received from multiple attentions, highlighting its importance to other words in the phrase. This example expands upon the information shown in the preceding illustration. The authors of the study integrated a total of eight multi-headed attention processes across all of the different layers.

Figure 2.3 illustrates the structure of the Transformer model so that a visual representation of the model may be provided. During the processing that takes place in an encoder block, the input embedding travels via the multiple-headed attention layer. After that, information goes through what's known as a feed-forward neural network.

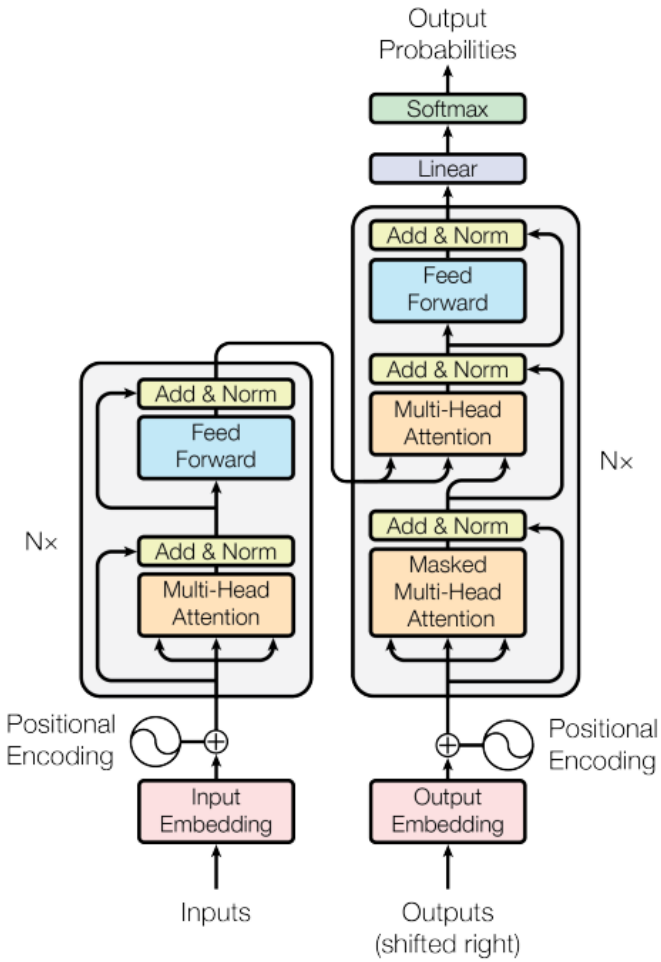

Figure 2.3: The transformer Model Architecture(Courtesy of [18]

In between each layer, the output is first normalized, then it is supplemented with the output from the layer below it by employing skip connections. After that, the information that was produced by the encoder layer is sent to the encoder-decoder attention block that is included inside the decoder. In its turn, the decoder takes the contextual information and decodes it in order to build a new sequence.

Overall, the Transformer architecture makes use of numerous self-attention modules, which results in enhanced representations and a full grasp of contextual connections. This is achieved by the employment of multiple self-attention modules. The model includes encoder blocks, which include multi-headed attention and feed-forward neural networks, as well as encoder-decoder attention blocks inside the decoder to allow successful context decoding. Additionally, the model includes encoder-decoder attention blocks within the encoder.

Positional Embedding: In the fundamental design of the Transformer, there are no recurrent units, which may appear counterintuitive given the common demand for se-

quence data processing. Nevertheless, the authors of [18] proposed a surprising solution to this problem that may be implemented. They came up with a technique that is now known as positional encoding. This encoding essentially specifies where each word is located inside the phrase. The representation of the input sequence is made more accurate thanks to this positional encoding approach. To explain, positional encoding is a way to include positional information into the Transformer model. This information may be encoded in a vector. The model is able to acquire a grasp of the relative locations and dependencies among the words in the sequence once the unique positional embeddings have been assigned to each word in the sequence. This is very necessary in order to capture the sequential nature of the incoming data successfully. Because of the positional encoding strategy, the Transformer model is able to get around the problem of having insufficient explicitly recurring units. This is because the technique enables the model to implicitly capture and capitalize on the sequential order of the words included inside a phrase. As a consequence of this, the positional encoding system makes a contribution to a representation of the input sequence that is more complete and efficient, which in turn makes it easier to make accurate and context-aware predictions. In conclusion, the authors of the study introduced positional encoding as a solution to the problem of the lack of recurrent units in the Transformer design. This approach stores the positional information of each word in the phrase, which improves the model's capacity to capture sequential dependencies and enables it to handle sequence data efficiently. Additionally, the model's ability to process sequence data is enhanced.

BERT: Bi Directional Encoder Representation of Transformers: Following the emergence of transformer architecture in [18], there has been a significant advancement in the field of Natural Language Processing (NLP). However, during that period, the prevailing models were predominantly focused on specific tasks. In 2018, a pivotal development was introduced by [19] in the form of BERT (Bidirectional Encoder Representation of Transformer). BERT revolutionized NLP by introducing a common pretraining approach. It employed Masked Language Modeling and Next Sentence Prediction techniques during pretraining. The Masked Language Modeling approach enabled BERT to effectively capture context from both directions, while Next Sentence Prediction enhanced its ability to comprehend sentence context. With the advent of BERT, the concept of a generalized model design was introduced. After undergoing pretraining, BERT became adaptable to fine-tuning for various downstream tasks. Figure 2.4 depicts a generalized architecture of BERT.

Text-To-Text Transfer Transformer: The Text-To-Text Transfer Transformer, more often known as the T5 model, is an effective framework for natural language processing

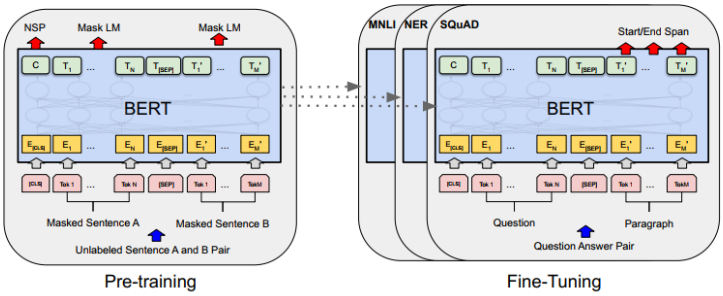

Figure 2.4: Overall pre-training and fine-tuning procedures for BERT. Apart from output layers, the same architectures are used in both pre-training and fine-tuning. The same pre-trained model parameters are used to initialize models for different down-stream tasks. During fine-tuning, all parameters are fine-tuned. [CLS] is a special symbol added in front of every input example, and [SEP] is a special separator token (e.g. separating questions/answers).(Courtesy of [19])

(NLP) activities. It does this by converting the problems that are text-based into a format that is typical for text-to-text communication. This makes it possible to solve a broad variety of text-based language issues. The fundamental concept behind T5 is referred to as "transfer learning," in which the model is originally pre-trained on a large dataset known as the "Colossal Clean Crawled Corpus." During this pre-training phase, the model is allowed to learn generic representations of language and recognize underlying patterns in the data.

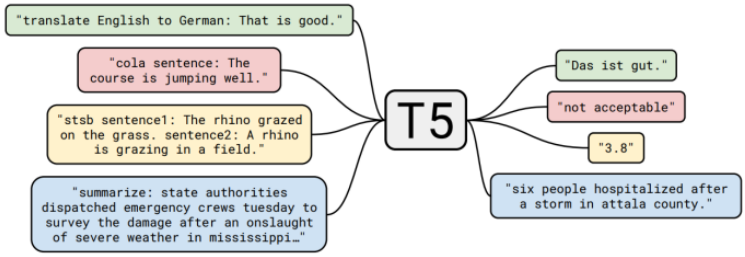

Figure 2.5: Diagram explaining T5's text-to-text framework..(Courtesy of [20])

In the T5 framework, input text is formatted such that it includes a prefix that indicates the intended job, such as "translate to English," followed by the actual text on which the operation should be done. For example, "translate to English." Because of this, T5 is able to perform a wide range of jobs, such as translation, summarization, question answering, and many more. The outcome of carrying out the particular operation on the text that was provided as input is what is known as the model's output. The adaptability and scalability of T5 are two of its most remarkable characteristics. It is simple to alter the task-specific prefix and train the model on data relevant to the particular NLP job in order to make it readily adaptable and extensible to a variety of NLP

applications. Because of this, T5 is a flexible tool that may be used to solve a broad variety of issues that are linked to language. The T5 model has garnered widespread support from members of the NLP community because to its outstanding performance on a number of different benchmarks. It provides a consistent and efficient method to transfer learning, which enables academics and practitioners to harness its capabilities for a broad range of linguistic problems. This is made possible by the fact that it offers a transfer learning framework.

GPT-3: Significant progress has been achieved in natural language processing (NLP) thanks in large part to the GPT-3 (Generative Pre-trained Transformer 3) model, an advanced language model. It has the transformer design of the older GPT-2 but is far bigger and more powerful. The original GPT architecture is described in Figure 2.6

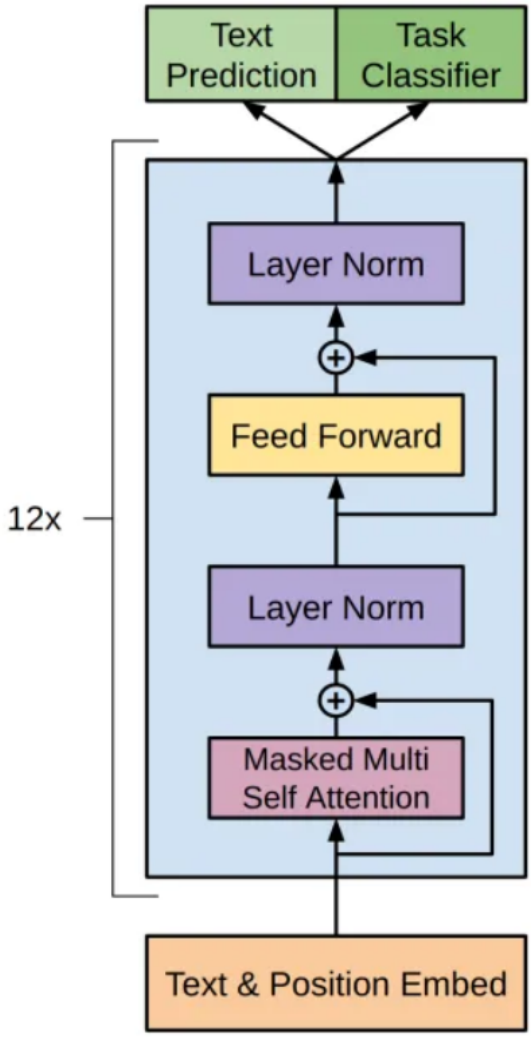

Figure 2.6: The original GPT architecture (Courtesy of [21])

The large number of parameters in GPT-3's design is what gives it its reputation for

excellence in language processing and pattern recognition. GPT-3 has been pre-trained on a vast trove of different text material, allowing it to pick up on the subtleties and intricacies of language. The model makes use of a series of self-attention transformer layers. These processes allow GPT-3 to create coherent and contextually relevant text by successfully capturing linkages and dependencies between words. When it comes to language-related tasks, GPT-3 shines. This includes, but is not limited to, text completion, translation, summarization, question answering, and more. It has shown exceptional performance in various tasks, often outperforming state-of-the-art results and benchmarks. The capability of few-shot and zero-shot learning is one of GPT-3's defining characteristics. The capacity to learn a new task with a limited number of examples is called "few-shot learning," whereas "zero-shot learning" describes a model's ability to create replies or complete tasks without any prior training in that area. Because of its adaptability and versatility, GPT-3 has found broad use in fields as diverse as virtual assistance, content production, language translation, and even creative writing. Its extensive vocabulary and comprehension of language make it a potent instrument for solving a variety of natural language processing problems.

2.2 Fact Extraction and Claim Verification

The FEVER shared task [2] has introduced one of the first large-scale benchmark datasets for fact extraction and claim verification. Most of the literature divided the task into a three module pipeline: *Document Retrieval*, *Sentence Selection* and *Claim Verification*. The baseline system proposed by [2] evaluated each of these three submodule separately and reported the final accuaracy on the test set. For the first step, they filtered out k nearest relevant document in terms of the claim whose veracity needs to be identified, with the help of the DrQA system proposed by [22], where they rank the documents using cosine similarity between TF-IDF vectors. Next from the filtered documents, the most relevant sentences are also selected using TF-IDF similarity. Following this, a number of enhanced approaches were implemented for the entire pipeline; these will be described in the following sections.

Initially, Thorne et al. [2] formulated the claim verification subtask as a Natural Language Inference (NLI) that enabled them to concatenate all the evidence and the claim together and pass it to an NLI model to get the final prediction. Following this, Hanselowski et al. [23] proposed a method where they tackled the textual entailment between the claim and the evidence sentence pairs by adapting the Enhanced Sequential Interface Model (ESIM) [24]. Later on, most of the contemporary works have focused on employing language models such as BERT [19], GPT [21], etc., that are

pre-trained on large datasets, to solve the claim verification subtask particularly. All of the aforementioned approaches mainly focused on developing a claim verification system based on the dataset introduced in FEVER, where most claims require a single evidence sentence to be verified. Contrary to most prior work, in this paper, we also utilize the HoVer [4] dataset that is more relevant to real-world scenarios where the claims/statements need more than a single sentence to be authenticated.

2.2.1 Document Retrieval

An important opportunity for other academics to work on better methodology was presented by the first baseline approach, which relied on the conventional cosine similarity method to filter out relevant documents for review. The document retrieval baseline achieved about 70% accuracy. Then the first breakthrough was proposed by [23], where they were able to gain about 23% accuracy boost. This work describes an ad-hoc entity linking approach that uses Wikipedia as a knowledge source. The goal is to match entities mentioned in natural language claims to corresponding Wikipedia articles. The approach involves three main steps: mention extraction, candidate article search, and candidate filtering. Regular named entity extraction methods focus on basic entity types such as Location, Organization, Person, etc., but in order to find relevant Wikipedia documents, various categories of entities such as movie titles are of our interest. That is why mention extraction is carried out using a constituency parser and a heuristic that considers words before the main verb and the entire claim as potential entity mentions. For example, a claim "*Down With Love is a 2003 comedy film.*" contains the noun phrases '*a 2003 comedy film*' and '*Love*'. Neither of the noun phrases constitutes an entity mention, but the tokens before the main verb, '*Down With Love*', form an entity. The next phase of document retrieval is Candidate article search; it is performed using the MediaWiki API[1] to search for matches between potential entity mentions and Wikipedia article titles. Similar to previous work on entity linking by [25], candidate filtering removes articles that are longer than the entity mention and do not overlap with the rest of the claim. The retrieved Wikipedia articles are supplied to the next step in the pipeline. The system was evaluated on the development data and was found to be effective.

2.2.2 Sentence Selection

After filtering a set of Wikipedia documents relevant to a claim, the subsequent module of the pipeline involves selecting the most pertinent sentences from those documents. The baseline approach, proposed by [2], uses a ranking algorithm that ranks

[1] `https://www.mediawiki.org/wiki/API:Main_page`

all sentences based on the TF-IDF similarity with the claim. To determine the textual entailment between the claim and other sentences, an RTE model is utilized. The RTE model is essentially a multi-layer perceptron that employs a single hidden layer and term frequency and TF-IDF cosine similarity between the claim and evidence as features. The next big breakthrough in sentence selection module was achieved by *Enhanced Sequential Inference Model (ESIM)*. ESIM [22] with minor modifications were used in [23] [26]. The modified ESIM (Enhanced Sequential Inference Model) ranks sentences based on their relevance to a given claim. It takes a claim and a sentence as input. The last hidden state of the ESIM is passed through a hidden layer connected to a single neuron to generate a ranking score. This score is used to rank all sentences from retrieved documents. To identify potential evidence, the five highest-ranked sentences are selected. During training, the modified ESIM takes a claim and a set of

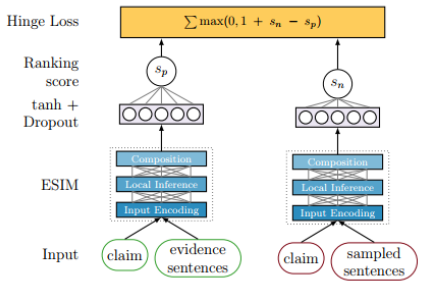

Figure 2.7: Sentence Selection Model (Courtesy of [23])

concatenated sentences as input. It uses a modified hinge loss function with negative sampling to calculate the loss. The positive ranking score (sp) is obtained by feeding the network a claim and the concatenated sentences from its ground truth evidence set. The negative ranking score (sn) is obtained by randomly sampling five sentences from Wikipedia articles that are not part of the ground truth evidence sets for the claim. Both sp and sn are computed using the same ESIM, and the goal is to maximize the margin between positive and negative samples.

During testing, an ensemble of ten models with different random seeds is deployed. The score between a claim and each sentence in the retrieved documents is calculated using the ensemble. The mean score of a claim-sentence pair across all ten models is computed, and the pairs are ranked based on these scores. The top five pairs, along with their corresponding sentences, are considered as the output of the model. With the emergence of transformer-based models, there has been a growing trend in utilizing the considerable efficacy of pre-trained knowledge within these models to ascertain sentence relevance scores and subsequently filter sentences of highest ranking based on a provided claim. [6] applied BERT based models with two different approach of

pointwise and pairwise loss function while fine-tuning. Each input consists of a set of sentences and also the claim. The pointwise approach requires each input to be classified as either evidence or non evidence. And the exact loss function that is used in this approach is as follows:

$$Loss_{point} = \sum_{i=1}^{N} y_i log(p_i) \tag{2.3}$$

On the other hand, in pairwise approach, a positive and a negative sample are passed through BERT embedding layers separately and then compared against each other by a ranknet loss function, where the ouput from the positive sample and negative sample are used to train the following loss function:

$$Loss_{Pair}^{Hinge} = \sum_{i=1}^{N} max(0, 1 + o_{neg} - o_{pos}) \tag{2.4}$$

For both pointwise and pairwise experiments, a pre-trained BERT model was used. For further training, a batch size of 32 and a learning rate of $2e-5$ were adopted for one epoch. The mechanism of these two approach is depicted in Figure 2.8.

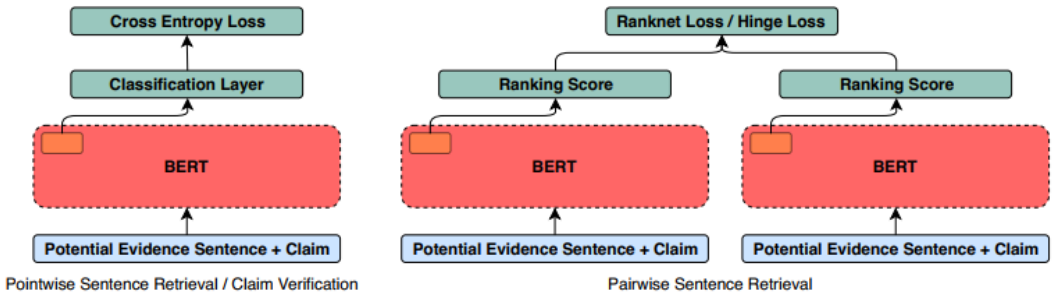

Figure 2.8: Pointwise sentence retrieval and claim verification (left), Pairwise sentence retrieval (right). (Courtesy of [6])

2.2.3 Multihop Iterative Sentence Selection:

By considering the methods mentioned so far, it becomes apparent that they all share a common flaw. These approaches simultaneously filter out entire documents and sentences, potentially resulting in inaccuracies. This is because there are instances where an evidence sentence relies on the retrieval of a previous evidence sentence. Motivated by this observation, a novel late interactive method has been proposed by [27]. This research paper introduces Baleen, a cutting-edge multi-hop reasoning system. Baleen enhances its efficacy through an iterative retrieval and compression procedure. This

iterative procedure is repeated for T predetermined time increments. In the case of HotpotQA, T is set to 2, whereas T is set to 4 in the case of HoVer.

At each time step, Baleen retrieves the top K passages using the query from the previous phase. It then creates a new query for the current time step by extracting and condensing relevant facts (sentences) from the passages retrieved. At each hop, Baleen also summarizes each passage to reduce the search space, which makes it even more efficient. This meticulous extraction and condensing mechanism plays an essential role in refining the data required for reasoning. Once the retrieval process is complete (i.e., $t = T$), the final query is passed into the reader model to produce the final predictions. The reader model makes accurate predictions using the refined data obtained through the iterative retrieval and condensing stages. For the purpose of determining Baleen's efficacy, exhaustive experiments were conducted on prominent question-answering and claim verification benchmarks, namely HotpotQA and HoVer. The experimental results on HotpotQA demonstrate Baleen's competitive passage retrieval performance. In addition, the HoVer results demonstrate its state-of-the-art performance, which substantially exceeds the performance of baseline methodologies.

Overall, Baleen presents a novel strategy that employs iterative retrieval and condensing to enhance reasoning abilities in multiple-hop question-answering tasks. The experimental findings confirm its efficacy and demonstrate its superiority over existing techniques.

2.2.4 Claim Verification:

The final sub-task within the module pertains to claim verification, wherein a language model is presented with a set of evidence sentences and a claim. The primary objective is to predict the ultimate label, which can either be "Supported" or "Not Supported." In the earlier approach, a neural semantic model is utilized. This model incorporates GloVe and ELMo embeddings, supplemented by additional separate vectors that are generated specifically for this purpose. [26] The additional vector is generated from a Wordnet library meant to describe certain emotional features of a sentence. The process is described in the following figure:

With the introduction of the encoder-decoder language model, subsequent methodologies naturally employed BERT-like models for the purpose of categorizing claims into specific labels. In the publication referenced as [6], the authors selected the most relevant five sentences from the sentence retrieval module. These sentences were subsequently subjected to individual assessment using a BERT classification head, and their outcomes were compared independently. Ultimately, the results were consolidated to determine a final class label, which may be designated as either "Supported"

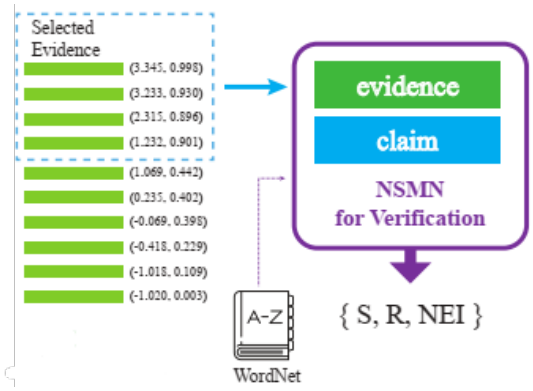

Figure 2.9: Claim Verification via NSMN model. (Courtesy of [26])

or "Not Supported." The whole pipeline became significantly easier with BERT, as demonstrated in the Figure 2.10.

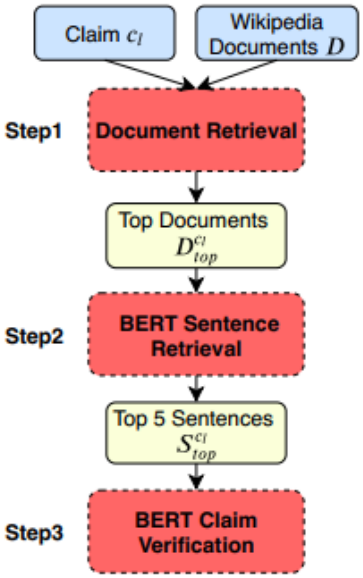

Figure 2.10: Three-step pipeline evidence extraction and claim verification. (Courtesy of [6])

One significant drawback observed in previous approaches is their inability to accurately consider the interdependency among evidence sentences, particularly when a claim relies on a multi-hop reasoning chain encompassing several pieces of information. To address this issue, graph-based approaches have shown some success by establishing an evidence reasoning chain among multiple evidence sentences. The GEAR framework, proposed by Zhou et al. in [28], offers a solution in claim verification by employing a series of steps.

Firstly, a sentence encoder based on BERT is utilized to obtain representations for both the claim and the retrieved evidence. The final hidden state of the [CLS] token in

BERT serves as the representation for each input sentence. To guide the message passing process in the reasoning graph, the evidence and claim are concatenated to extract the evidence representation, taking into account the claim's informative guidance.

Next, an evidence reasoning network (ERNet) is constructed to facilitate the propagation of information among the evidence nodes. A fully-connected evidence graph is established, where each node represents a piece of evidence. Self-loops are added to enable each node to receive information from itself during the message propagation. Attention coefficients between each node and its neighbors are computed using a multi-layer perceptron (MLP). These coefficients are then normalized using the softmax function. Through the iterative application of the ERNet across multiple layers, information is effectively propagated among the evidence nodes. The final hidden states of the evidence nodes are then fed into an evidence aggregator to make the final inference.

The evidence aggregator collects information from diverse evidence nodes to obtain the ultimate hidden state. Within the framework, three types of aggregators are suggested: attention aggregator, max aggregator, and mean aggregator. The attention aggregator utilizes the claim representation to attend to the hidden states of evidence and derive the aggregated state. The max aggregator conducts an element-wise maximum operation on the hidden states, while the mean aggregator performs an element-wise mean operation. Once the final state is obtained, a one-layer multi-layer perceptron (MLP) is employed to generate the final prediction.

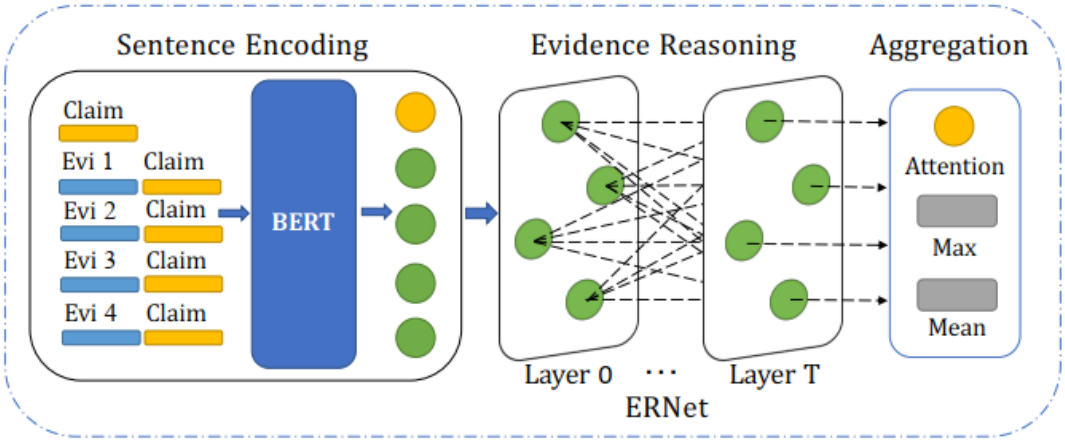

Figure 2.11: Claim Verification using GEAR framework [28])

To summarize, the GEAR framework addresses the issue of interdependency among evidence sentences in claim verification. It accomplishes this by employing a sentence encoder, constructing an evidence reasoning network for information propagation, and

employing an evidence aggregator to obtain the final prediction. The claim verification process using the GEAR framework is depicted in the Figure 2.11.

Chapter 3

Proposed Methodology

In this chapter, we discuss the proposed pipeline for joint position-based gait recognition system. First, we provide an overview of the overall architecture. Then we describe each of its individual components and the reasoning behind our choices.

3.1 Overview

In order to elicit the full potential from language models when it comes to the claim verification task, it is crucial to generate language prompts that appear organic, i.e., close to what a genuine English passage may seem as the models are pre-trained on organic English texts sources such as Wikipedia. We reformulate the claim verification task into a language generation task using prompt-tuned language models [29]. As for the input to the model, we generated natural language prompts containing the evidence sentences followed by the original claim and a manually selected question template that stays the same for all the inputs. An overview of the proposed approach is given in Figure 3.1, and detailed discussions of different prompt variations are included in the experiment section. We evaluate the effectiveness of prompt-based methods for claim verification in two scenarios:

3.2 Prompt Tuning for Claim Verification:

The use of "prompts" to guide language models to carry out a variety of tasks has been on the rise [29]. Using prompts to keep the fine-tuning objective similar to pre-training is found to be very helpful to effectively utilize linguistic knowledge from pre-trained language models while alleviating the discrepancy between the pre-training and fine-tuning objectives [30]. A prompt refers to a textual fragment that is inserted within input examples, enabling the formulation of the original task as a (masked) language modeling problem. For instance, if our objective is to classify the sentiment of the

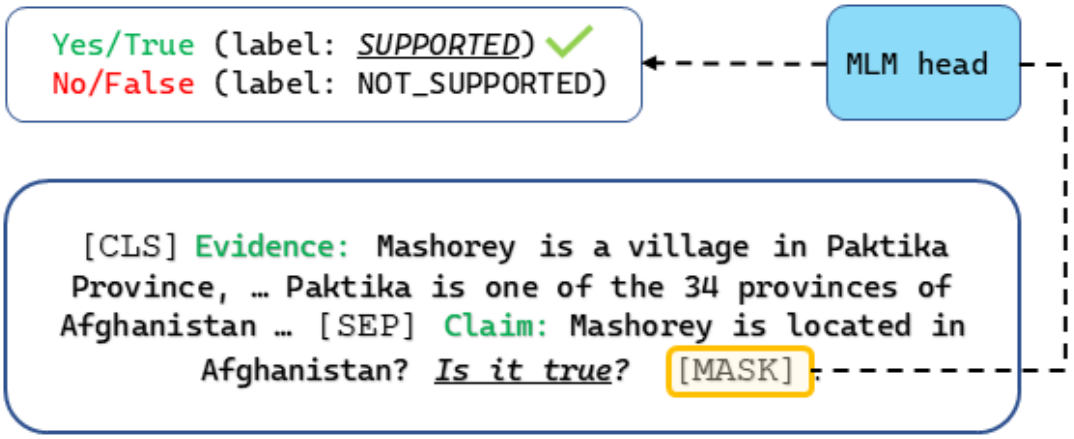

Figure 3.1: An overview of our proposed prompt-based fine-tuning technique for the claim verification task. In our formulation, the evidence passage is appended with the claim posed as a question, and followed by another question type prompt: 'Is it True? [MASK]'. Consequently, the most probable word, in this case, "Yes" is predicted to replace the masked token to support the claim.

movie review 'There is no need to watch this movie', we can simply append an additional piece of text with a blank to it and let a language model predict a suitable word in the blank. After adding a prompt, the text may look like: 'There is no need to watch this movie, it was _____.' A good generative language model would predict words such as 'bad','terrible' in place of the blank, and finally by mapping the predicted word, we can classify the original sentiment of the review. [31] Recently, the utilization of prompts for various tasks, such as extracting factual or commonsense reasoning from LMs has been investigated [32–35]. Nonetheless, the use of prompts for the claim verification task is yet to be investigated. In this paper, we use cloze questions as a means of designing prompts following the work of [36]. Moreover, we use the masked tokens as the final outcome for a new task for prompt learning – the claim verification task. Additionally, many recent models like GPT-3 [7] possess a remarkable ability to retain vast amounts of information in their parameters. This enables them to perform well even in previously unseen scenarios, making them the preferred choice for few-shot and zero-shot scenarios. Hence, in this paper, our research also focuses on evaluating the performance of the highly capable GPT-3 model *text-davinci-003* in a few-shot and zero-shot claim verification task using the HoVer dev set and comparing its results with other models.

3.2.1 Closing the Gap between Pretraing and Finetuning?:

In the process of fine-tuning, a substantial language corpus is initially transformed into an appropriate training task, with the prevalent approach being the utilization of a MASKED word prediction task. In this task, a subset of words in the original corpus is

masked by employing a distinctive token ([MASK] token), following which the model undergoes training to anticipate a suitable word to occupy each masked position. With the large language corpus, after sufficient epoch of training, the model is expected to learn the latent linguistic characteristics well enough to perform different types of task.

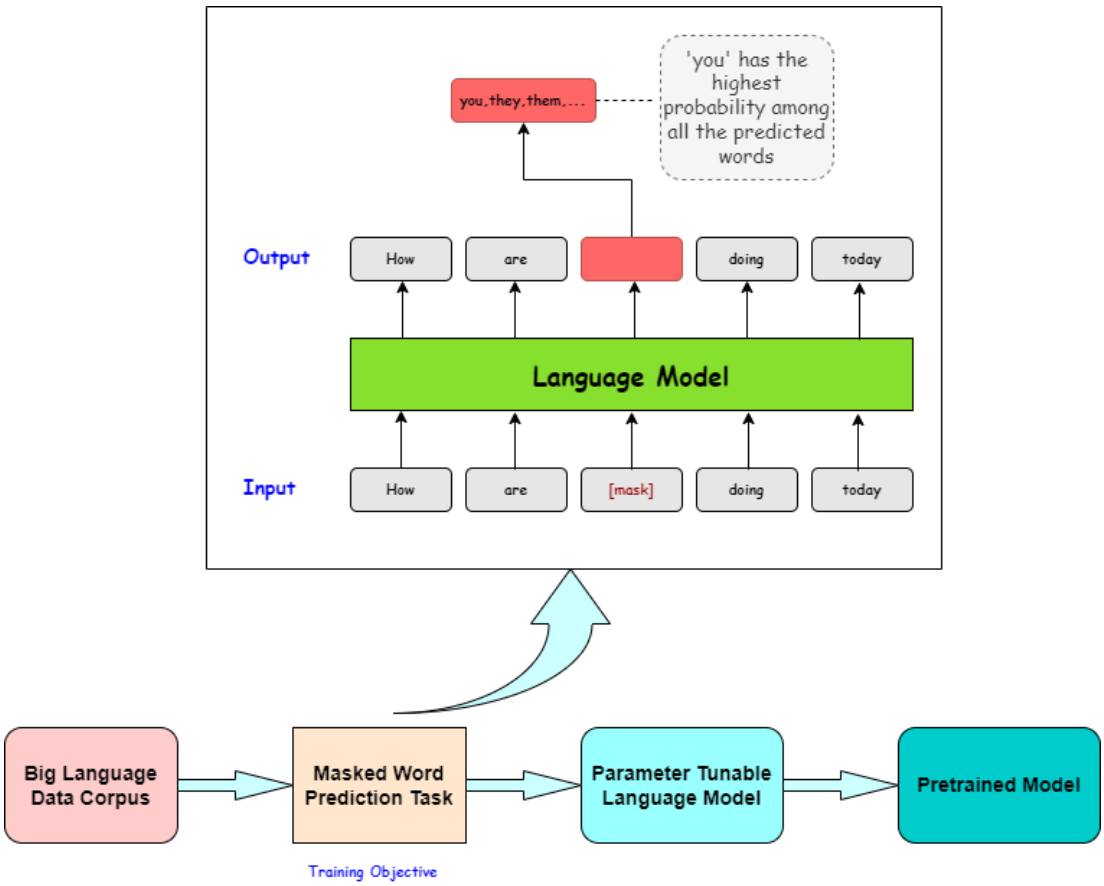

Figure 3.2: Traditional Finetuning Process of a Language Model

The procedure is visually depicted in Figure 3.2.

But Merely relying on pretraining is insufficient for achieving satisfactory performance across various downstream tasks. This is where the concept of finetuning becomes crucial. Theoretically, we could finetune the entire pretrained model using task-specific data from a new corpus. However, such an approach can be excessively resource-intensive. A more elegant approach involves freezing the initial layers of the language model, which have been empirically demonstrated to capture general linguistic features, and focusing the training solely on the final few layers of the model. These latter layers are predominantly responsible for acquiring task-specific features related to the downstream task at hand. In our case, the task-specific dataset at at hand is claim verification dataset. The conventional finetunig process using our dataset is depicted in the figure 3.3.

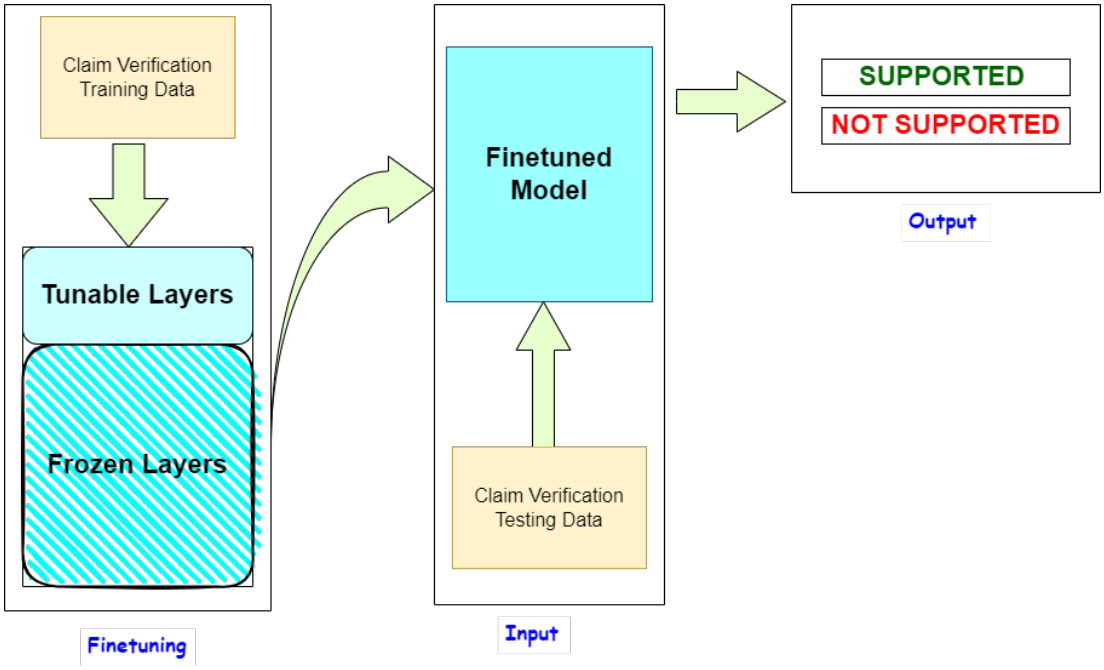

Figure 3.3: Finetuning Process of Language Models using Task Specific Data

The conventional approach of "pre-training and fine-tuning" exhibits a significant disparity between the initial pre-training phase and the subsequent downstream task. This disparity arises due to differing objectives, often necessitating the inclusion of new parameters. For instance, an additional set of 1,024 x 2 parameters is required to utilize a BERT-large model for binary classification. Prompts can align the downstream tasks with the pre-training objectives without introducing new parameters. In the context of a classification task, let us consider the implementation of the "<text> It is <mask>" template, where "<text>" represents the original text. The system consists of the mappings: "positive": "great," "negative": "terrible". To demonstrate this, let us take the sentence, "Albert Einstein was one of the greatest intellects of his time." Initially, the sentence is enveloped within the predefined template, resulting in "Albert Einstein was one of the greatest intellects of his time. It is <mask>". Subsequently, the wrapped sentence undergoes tokenization and is fed into a pre-trained language model (PLM) to predict the probability distribution across the vocabulary, specifically for the "<mask>" token position. Ideally, the word "great" should exhibit a higher probability than "terrible." By narrowing the gap between these two stages, the deployment of pre-trained models for specific tasks becomes considerably simpler, particularly in scenarios with limited training examples (few-shot). Even in the absence of training using task-specific data, the testing set can exhibit improved accuracy, as the task is transformed in a manner resembling the original relevant objective. This scenario is commonly referred to as a "zero-shot" scenario, wherein the pretrained model pos-

sesses no prior knowledge of any training sample. The process can be visualized in Figure 3.4.

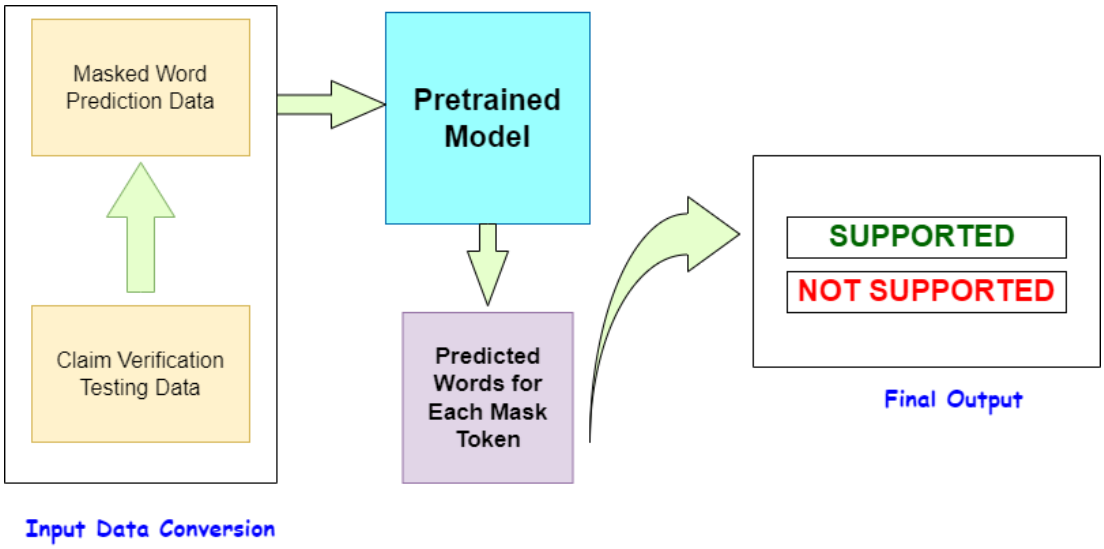

Figure 3.4: Prompt Tuning in Zero Shot Scenario

The prompt tuning paradigm proves to be particularly advantageous in scenarios where the available training samples are limited in number. Traditionally, employing the standard finetuning approach with such a restricted dataset often falls short of achieving a satisfactory level of accuracy. However, by introducing appropriate prompts into the original training samples and adapting the finetuning process to a prompt tuning setup, we can effectively enhance the accuracy, even when dealing with a smaller set of training samples. The key to this approach lies in the strategic utilization of prompts, which are carefully crafted instructions or queries designed to guide the model's response generation. By incorporating prompts that encapsulate the desired behavior or context of the task, we can effectively leverage the pretrained model's language understanding capabilities to yield more accurate outputs. With prompt tuning, the prompts act as a form of guidance that facilitates the model's learning process and enables it to generalize more effectively from the limited training samples. This paradigm offers a practical and efficient solution, as it allows us to make the most of the available data, even when it is insufficient for conventional finetuning methods. In our specific case, tackling the task of multihop claim verification presents a formidable challenge. Additionally, an even greater obstacle arises when attempting to construct an appropriate dataset that encompasses the complex nature of multi-hop reasoning. In light of these difficulties and the scarcity of suitable data, prompt tuning emerges as an ideal alternative to address these limitations effectively.By utilizing prompt tuning, we can circumvent this limitation and maximize the potential of the existing data. Through the strategic inclusion of prompts that simulate the multi-hop reasoning pro-

cess, we can effectively train the model to handle the complexities of the task, even with limited training examples. The process of finetuning with fewer training samples is described in the figure 3.5.

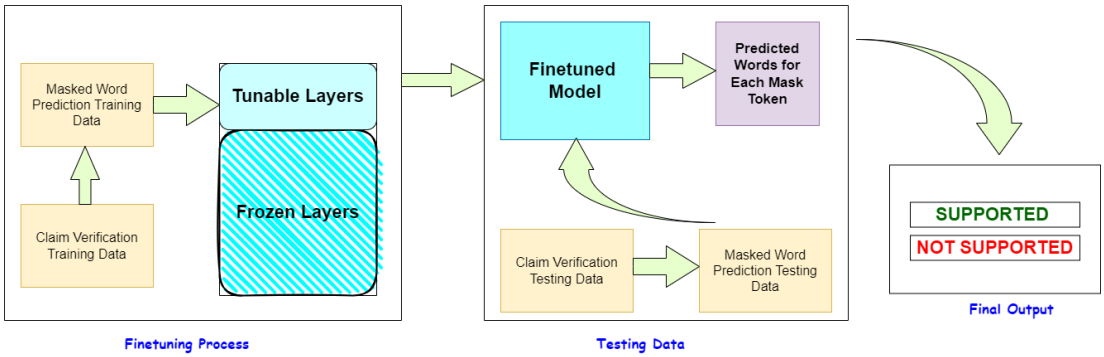

Figure 3.5: Prompt Tuning in Few Shot Scenario

Fine-tuning the pre-trained models and task-specific parameters becomes challenging under such circumstances. However, the implementation of prompting significantly streamlines this process. Recent research conducted by [37] suggests that a prompt can be as valuable as 100 conventional data points, emphasizing the substantial potential of prompts to enhance sample efficiency. Prompt research has given rise to two distinct paradigms, each presenting unique perspectives. Prompt-based fine-tuning, inspired by the PET papers [36], is considered a viable method for improving few-shot learning in smaller language models, typically encompassing millions of parameters, as opposed to the billions found in models like BERT or RoBERTa. In contrast, due to its difficulty and cost, fine-tuning poses challenges for super-large models such as the 175B GPT-3 and 11B T5 [20]. In such cases, it is more practical to maintain fixed model parameters and utilize different prompts for different tasks.

3.2.2 Openprompt Framework:

OpenPrompt is a comprehensive open-source library and framework designed to facilitate the seamless implementation of prompting techniques in natural language processing (NLP) tasks. It serves as a unified platform, catering to the needs of both researchers and practitioners, enabling them to experiment with and effectively apply prompts across various NLP models and applications.

The core of OpenPrompt revolves around a flexible and modular architecture that empowers users to define and customize prompts according to their specific requirements. The library supports different prompt formats, including text templates, cloze-style prompts, and masked language modeling prompts. This versatility allows users to adapt prompts to various NLP tasks, such as text classification, named entity recog-

nition, question answering, and machine translation.

The critical components of OpenPrompt include:

1. Templates: The template module is central to prompt learning. It wraps the original text with textual or soft-encoding templates. Templates typically consist of contextual and masked tokens (textual or soft). In OpenPrompt, all templates inherit from a standard base class with universal attributes and abstract methods. To enhance practical usability and minimize the learning cost, OpenPrompt employs a template language inspired by the dictionary grammar of Python. This design ensures flexibility and clarity, enabling users to construct various prompts easily.

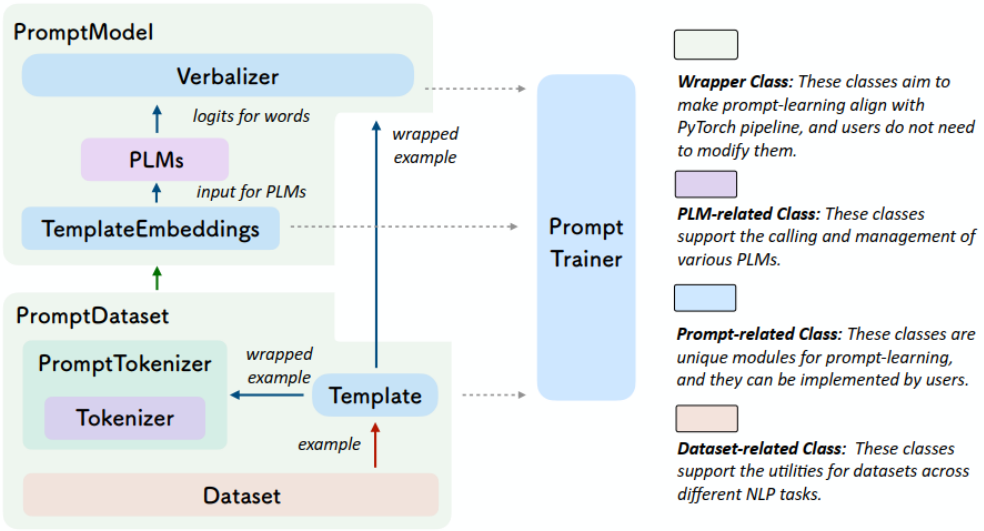

Figure 3.6: The overall architecture of OpenPrompt Framework

2. Verbalizers: In prompt-based classification, a verbalizer class is constructed to map original labels to label words in the vocabulary. When a pre-trained language model (PLM) predicts a probability distribution over the vocabulary for a masked position, a verbalizer extracts the logits of label words and integrates them into the corresponding class, thereby facilitating loss calculation. OpenPrompt provides a simple way to define binary sentiment classification verbalizers, as depicted in Figure **??**. Like templates, all verbalizer classes inherit from a standard base class with necessary attributes and abstract methods. In addition to manually-defined verbalizers, OpenPrompt implements automatic verbalizers such as AutomaticVerbalizer and KnowledgeableVerbalizer [38]. Furthermore, essential operations like calibrations [39] are also realized within OpenPrompt.

3. PromptModel: OpenPrompt utilizes a PromptModel object to handle training and inference tasks. This object comprises a PLM, a Template object, and an optional Verbalizer object. Users can flexibly combine these modules and define advanced in-

teractions among them. The base class implements a model-agnostic forward method, allowing users to "predict words for positions that need to be predicted" without specific head implementations for different PLMs. This approach ensures a unified API for prompt prediction across various pre-training objectives.

4. Training: From a parameter training perspective, prompt learning in OpenPrompt can be divided into two strategies. The first strategy simultaneously fine-tunes both the prompts and the PLM, which has been proven effective in low-data regimes. OpenPrompt also provides a FewshotSampler to support few-shot learning scenarios. The second strategy involves training only the prompt parameters while keeping the PLM frozen. This approach is considered a parameter-efficient tuning method and holds promise for optimizing super-large PLMs. OpenPrompt's trainer modules implement the training process, incorporating prompt-oriented training techniques, such as an ensemble of templates. Moreover, OpenPrompt supports experimentation through configuration, allowing for large-scale empirical studies. The library provides comprehensive tutorials that cover the usage of basic and advanced attributes in OpenPrompt.

In summary, OpenPrompt offers researchers and practitioners a powerful toolkit for prompt-based NLP tasks. Its flexible architecture, support for various prompt formats, and integration with pre-trained language models enable efficient, prompt implementation and experimentation. Through OpenPrompt, users can explore and harness the potential of prompts to enhance model performance, efficiency, and transfer learning in diverse NLP applications.

3.2.3 Supervised Fine-Tuning Scenarios:

For this experiment, we utilized the T5 model, which is a pre-trained prompt-based language model that leverages in-domain fine-tuning on the training set of claim verification datasets. We employ the OpenPrompt[1] [40] framework in conjunction with the prompt-based PLM: *T5* [20]. The OpenPrompt framework utilizes hand-crafted templates and label words to provide a probability distribution over the vocabulary for each masked position. Next, the probability logits only for the label words are mapped back to their original classes via the OpenPrompt verbalizer in an automated fashion. In this paper, we use several variants of the T5 [20] model (T5-base and T5-large) to conduct our experiments. T5 is a transformer-based model that identifies each task as a sequence-to-sequence problem, as opposed to the traditional BERT like language models [19, 41, 42] that assign a class label to the input text. With different training objectives, the model is pre-trained on a large corpus and fine-tuned on task-specific inputs in order to produce the required outputs. In our case study, the training of T5

[1] https://github.com/thunlp/OpenPrompt

and its variants required us to adapt the OpenPrompt [40] framework with a variety of different label words and templates.

3.2.4 Zero-Shot Learning Scenarios:

We used one of OpenAI's most advanced generative model, the `text-davinci-003`, to evaluate the effectiveness of prompt-based methods in zero-shot scenarios. We structured an input prompt by combining evidence sentences, a claim, and a question asking whether the claim is supported by the context or not. Since the `text-davinci-003` model is a generative model that tends to provide detailed responses, we manually evaluated its responses and categorized them as either "SUPPORTED" or "NOT SUPPORTED". For example, upon providing an input prompt as: *'Context: Universities and university colleges normally use the ECTS grading scale. The ECTS grading scale is a grading system defined in the European Credit Transfer and Accumulation System (ECTS) framework by the European Commission. Claim: North America countries utilize the ECTS grading scale that Norway adopted. Is the claim supported or not supported according to the context?'*, the model `text-davinci-003` generated the following response: *'Not supported. The context does not mention any North American countries utilizing the ECTS grading scale that Norway adopted.'* From this response, it is evident that the claim should belong to "NOT SUPPORTED" class.

3.2.5 Multitasking Ability of Prompt Tuned Models:

The traditional finetuning strategy has a significant drawback when it comes to multitasking capabilities. In this approach, pretrained language models need to be individually finetuned for specific tasks, and the knowledge gained from one task does not transfer effectively to other downstream tasks. However, prompt tuning introduces a paradigm shift by allowing us to overcome this limitation. In prompt tuning, the concept of converting any task into a common task similar to the original training objective becomes pivotal. By leveraging this approach, we can finetune a pretrained language model using the training dataset of one task and effectively apply the knowledge gained to another task. This newfound flexibility and convenience unlock the potential for streamlining the use of a single-core model across various tasks. With prompt tuning, the traditional barriers between tasks are significantly reduced. Previously, each task required its own specialized finetuning process, which was time-consuming and computationally intensive. However, in prompt tuning, the training dataset of one task can serve as a foundation for finetuning the model, enabling it to capture the essential patterns and features necessary for that particular task. This knowledge can then be

carried over and utilized in subsequent tasks, eliminating the need for separate finetuning steps for each specific task.

In summary, prompt tuning revolutionizes the finetuning process by enabling the conversion of any task into a common task similar to the original training objective. This breakthrough facilitates the transfer of knowledge across tasks, allowing us to use a single core model for multiple tasks efficiently. With prompt tuning, the barriers to multitasking are significantly reduced, leading to enhanced productivity and knowledge sharing in the field of natural language processing and beyond.

We conducted experiments in the field of claim verification, specifically focusing on two benchmark datasets as mentioned earlier. One dataset consisted of single hop factual claims, while the other included multihop factual claims. By exploring various combinations, we aimed to empirically demonstrate that models finetuned using language prompts from one dataset performed better on a combined test set, which included both single and multihop samples, compared to models finetuned without language prompts. The process is described in the figure 3.7 and the detailed result is described in the chapter 4.

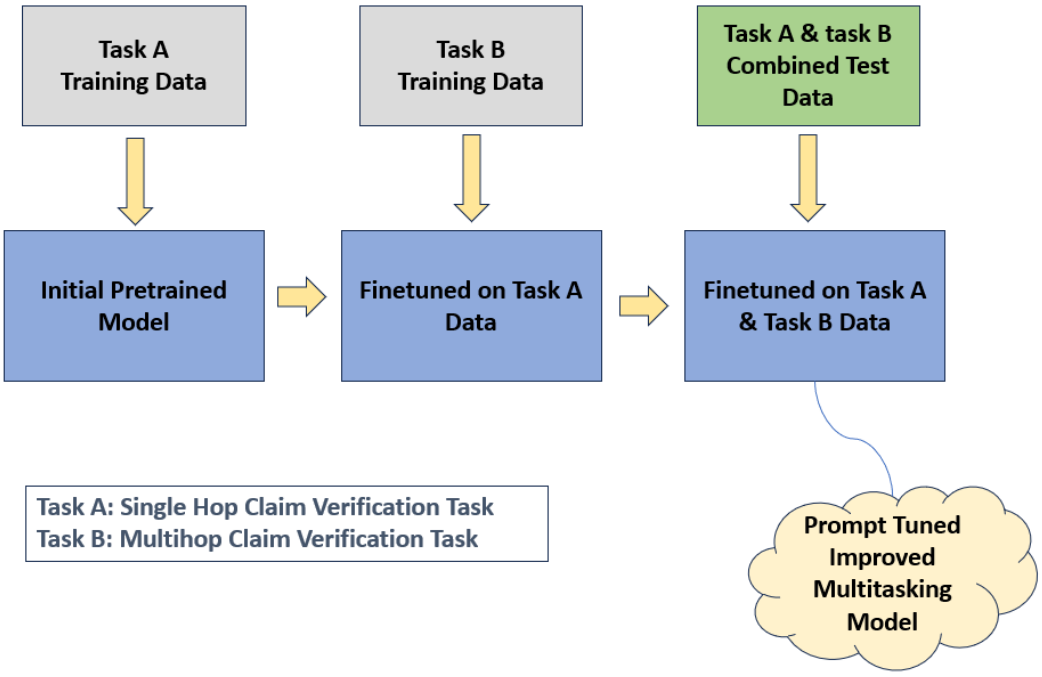

Figure 3.7: Prompt Tuned Model is a better Multitasking Model

Chapter 4

Results and Discussion

4.1 Datasets

In this paper, we particularly leverage the only two benchmark claim verification datasets that are well renowned, namely FEVER and HoVer. Since these in FEVER, there are 185445 claims, the vast majority of which just include a single sentence providing evidence. It is an opensourced dataset first introduced in FEVER shared task for fact extraction and claim verification from wikipedia articles [2].[1] Table 4.1 displays the full statistics of this dataset.

Split	SUPPORTED	REFUTED	NOT ENOUGH INFO
Train	80035	29775	35639
Test	3333	3333	3333
Dev	3333	3333	3333
Reserved	6666	6666	6666

Table 4.1: FEVER dataset split for SUPPORTED, REFUTED, NOT ENOUGH INFO classes

On the other hand, HoVer is made up of substantially fewer samples than FEVER, about nine times fewer, but each of the claims has multiple relevant sentences sourced from different documents that serve as evidence. It was first introduced by [4] for the multihop claim verification task. The detailed breakdown of this dataset is provided in Table 4.2.

Split	SUPPORTED	NOT-SUPPORTED
Train	11023	7148
Dev	2000	2000
Test	2000	2000

Table 4.2: HoVer dataset split for SUPPORTED and NOT SUPPORTED classes

[1]The reserved set mentioned in the Table 4.1 is used as a blind test set for the original FEVER shared task.

4.1.1 Combined Train and Dev Set Creation

To measure the performance of our approaches in a cross-domain setup, we utilized a combined training dataset and tested the performance of the prompt-tuned and non-prompt tuned models on a combined dev set. Both of these combined versions contain data samples from two separate datasets, i.e., HoVer [4] and FEVER [2]. But the FEVER dataset contains three different labels: 'SUPPORTED', 'REFUTED', and 'NOT ENOUGH INFO', while the HoVer dataset contains only two labels: 'SUPPORTED' and 'NOT SUPPORTED'. For our experiments, from the FEVER dataset, we discarded all the data samples labeled 'NOT ENOUGH INFO' since they do not provide any gold evidence sentence. As for the remaining two types of samples, we considered the 'REFUTED' label equivalent to the 'NOT SUPPORTED' label from HoVer, and data samples labeled as 'SUPPORTED' in both datasets have been considered equivalent. The combined dev set contains a total of 15001 samples, whereas the combined training set has 120549 data samples.

The development sets of Hover and FEVER were combined with the intention of testing the robustness of the fine-tuned models on a situation that was more representative of the real world, in which claims often require both single and multi-hop reasoning for verification.

4.2 Experimental Setup

In our supervised learning experiments, we have used 0.00001 as the learning rate. Experiments were run on Google Colab with an Nvidia A100 GPU and locally using an Nvidia RTX 3090 GPU. We did not run more than 5 epochs in any of our experiments.

For the Zero-Shot evaluation, since the `text-davinci-003` model gives generative output, we manually compare its generated response with the gold label to measure the accuracy.

Throughout our evaluation, we have utilized accuracy as the primary metric. The balanced nature of our dataset, where both classes (SUPPORTED and NOT SUPPORTED) hold equal importance, makes accuracy an appropriate and reliable measure. Additionally, considering that other baselines have also used accuracy as their metric, using the same evaluation metric enables a meaningful comparison between our work and theirs.

Model	Accuracy
BERT + Oracle [4]	81.20
Prompt Learning with T5-large	**83.89**

Table 4.3: Comparison of Performance on the HoVer Dev Set

4.3 Discussions

Since our focus is the more challenging claim verification task, the multi-hop claim verification, we primarily use the HoVer development set to evaluate the performance of our proposed prompt learning models. Below, we first describe the performance based on our prompt-based models that leverage supervised fine-tuning, followed by the zero-shot prompt-based models. Finally, we conduct some error analyses to further evaluate the performance of prompt learning for claim verification.

4.3.1 Performance based on Supervised Fine-Tuning

For supervised fine-tuning scenarios: (i) we evaluate the performance of our proposed approach on the HoVer development set by comparing it with other baselines (BERT, DistilBERT) that did not leverage prompt learning, (ii) we conduct experiments to evaluate the cross-domain generalization performance of our proposed approach via combining the development set of HoVer and FEVER, (iii) we train our model for few-shot learning with randomly re-sampled training sets of the following sizes {64, 128, 256, 512, 1024}, similar to the settings of Seoh et al. [10], (iv) finally, we evaluate the performance based on prompt variations.

(i) Prompt-tuning outperforms traditional fine-tuning. Here, we first discuss our findings in the original claim verification task. Our experimental results are shown in Table 4.3. According to the table, it is evident that our prompt-based method performs considerably better than the no-prompt-based baselines. In comparison to the baseline BERT model that achieved 81.20% accuracy [4], we are able to obtain 83.89% accuracy when evaluated on the HoVer dev set[2] using the T5-large model fine-tuned on the HoVer training set using our prompt learning setup, outperforming the baseline by 2.69%.

(ii) Prompt-tuning can achieve better cross domain generalization. Here, we present our findings based on cross-domain generalization performance of our proposed method. For that purpose, we combine the development sets of HoVer and FEVER together and compare the performance of the prompt-based T5-large model

[2]We evaluated our models performance on the dev set because it contains the original gold evidence from the dataset whereas test set does not provide any gold evidence sentence.

Training Dataset	Models		
	BERT	DistilBERT	T5
HoVer	61.09	73.02	**86.76**
FEVER	70.95	74.56	**77.36**
HoVer + FEVER	73.13	78.24	**82.30**

Table 4.4: Comparison of Performance in Cross-Domain Generalization. To evaluate the models' performance, we combined the development set of HoVer and FEVER.

with BERT and DistilBERT. We train these models in different training dataset settings, as shown in Table 4.4. We observe based on our experiments that prompt-based language model demonstrates better cross-domain generalizability in the combined development set[3] of both single-hop and multi-hop evidence reasoning.

In all three cases, the prompt-based model showed superior performance over traditional transformer encoder-based models, i.e., BERT and DistilBERT. One interesting pattern that can be observed from Table 4.4 is that, in the case of encoder-based models (BERT/DistilBERT), the performance improves with more training samples, hence achieving the highest accuracy when fine-tuned with the combined train set (HoVer + FEVER). Both these models achieve the second highest accuracy when finetuned with only FEVER dataset, and the lowest accuracy with HoVer train set as it contains the fewest training samples. However, in case of encoder-decoder-based model T5, the pattern is not the same. For the prompt-based T5 model, HoVer is proved to be a superior training resource, as it includes multiple contexts per claim. This large pool of context allows the T5 model to perform well in the combined dev set, whereas the performance of the model deteriorates when fine-tuned with only the FEVER dataset due to its limited number of evidence sentences to learn from. Meanwhile, one interesting finding is that combining the training data with both HoVer and Fever makes the model perform worse in the combined development set in comparison to the scenario when the model was only trained on HoVer dataset. This gives a strong indication that the prompt-based models suffers when datasets with fewer context to train from are used, whereas datasets with multiple contexts helps the prompt-based models (T5 in our case) perform better in the claim verification task.

(iii) Prompt-tuning is better for few-shot learning. Here, we present our experimental findings on the following few-shot learning scenarios: 64-shot, 128-shot, 256-shot, 512-shot, and 1024-shot scenarios. As evident from Table 4.5, there is a noticeable gap in the learning capacities of non-prompt-based baseline models in few-shot conditions when compared to prompt-based models. We observe that prompt-based methods (T5-base and T5-large) acquire an average of 3% accuracy gain using

[3]The detailed discussion of the combined dev set is provided in Dataset

Model	Number of Training Samples				
	64	128	256	512	1024
	Acc	*Acc*	*Acc*	*Acc*	*Acc*
BERT	49.79	50.24	49.64	50.65	51.77
DistilBERT	49.96	50.56	50.56	51.98	54.76
T5-base	52.01	**53.75**	**58.39**	58.93	69.81
T5-large	**52.38**	52.97	53.29	**62.07**	**70.70**

Table 4.5: Comparisons of Performance in terms of Few-Shot Learning. *Acc* refers to accuracy. We used three random seeds for each of the experiments and averaged their scores.

Template	Accuracy
`Context: {Evidence} Claim: {Claim}? Does the context provide enough evidence to support the claim fully? [MASK].`	81.60
`Context: {Evidence} Claim: {Claim} Is the claim supported according to the context? [MASK].`	83.33
`Evidence: {Evidence} Question: {Claim}? Is it true?[MASK].`	83.50
`{Evidence} Question: {Claim}? Is it true?[MASK].`	83.80
`Question:{Claim}? Is it true according to the context? Context: {Evidence} Question type: Yes-No. [MASK].`	**83.89**

Table 4.6: Performance based on different Language Prompts used in T5-large model

fewer training examples (although increasing the number of training samples also results in an even greater accuracy gain compared to traditional fine-tuning procedure). Moreover, while taking into account the whole dataset, we only used 5.6% (1024-shot) of the training set at most and yet we were able to achieve a competitive accuracy with the prompt-based fine-tuning strategy.

(iv) Different prompts yield different results. We experimented with several manually curated language prompts for the claim verification task where we concatenate the evidence passage with the claim. Some of our manually curated prompt templates and the accuracy gained by our best performing T5-large model when fine-tuned and tested using those prompts are listed in Table 4.6.

Note that in Table 4.6, we denote `{Claim}` to be the substitute for the claim statement and `{Evidence}` is set to be the placeholder for all of the evidence sentences concatenated together as a passage. Moreover, `[Mask]` refers to the label word that the PLM will predict, and it is essential that a set of label words mapped to individual

Model	Accuracy
BERT-base	49.80
BERT-large	50.20
DistilBERT-base	50.20
T5-base	49.90
T5-large	49.90
text-davinci-003	**60.48**

Table 4.7: Comparison of Zero-Shot Performance on the HoVer Dev Set

classes be specified beforehand. We experimented with a different set of label words (e.g., labels) for each class. For example, words like `'yes'`, `'correct'`, `'supported'`, `'poss` can be selected as label words for the SUPPORTED class and `'no'`, `'incorrect'`, `'not supported'`, `'impossible'`, etc., for the NOT SUPPORTED class. It is important to note that even minor alterations to the template affect the accuracy too, indicating that language prompts can be tailored to specific scenarios. Our experiments revealed that the model achieved the highest accuracy when the template was structured as a question (i.e., the last example in Table 4.6).

4.3.2 Performance based on Zero-Shot Learning

We conducted zero shot evaluation on the HoVer dev set using `text-davinci-003`. From our experiments (see Table 4.7), it is evident that `text-davinci-003` performs the best in the zero-shot setting. Though `text-davinci-003` obtains much lower accuracy than the supervised fine-tuned models with prompts (as we observed in Table 4.3), it outperforms the similar models (e.g., BERT, T5) by a large margin in zero-shot scenarios (see Table 4.7).

4.3.3 Label Word Probing

One important question occur when we think about why prompt tuning on T5 works better than BERT and what exactly we are doing different. Also, how is it fundamentatly different than classification tasks? We followed the method described in [43] We

	Positive Class	Negative Class	Accuracy
Baseline	supported	not supported	83.89
Reverse	not supported	supported	77.4
Antonyms	Hot	Cold	78.9
Related Words	Apple	Orange	79.74
Unrelated Words	Hot	Orange	79.14

Table 4.8: Results on the Dev Set with Varying Label Words

believe that these two issues are intricately related. Specifically, when answering the second query, both neural models play a crucial role in acquiring latent representations that are relevant to the given task, veracity classification. This method entails beginning with a model that has been pretrained and then mapping these latent representations into task-specific decisions. Therefore, the overall performance of the end-to-end task is dependent on a combination of the knowledge imparted during pretraining, which is already present, and the knowledge acquired through fine-tuning on task-specific data. In the classification-based approach employing BERT, the end-to-end model relies on a single fully-connected layer to facilitate the mapping of the latent representation (derived from the [CLS] token) into the required binary decision. While this method can utilize pretrained knowledge during the process of fine-tuning the latent representations, the final mapping, i.e. the fully-connected layer, must be learned fresh because it is initialized arbitrarily.

T5, on the other hand, can utilize both pretrained and fine-tuned knowledge in order to learn the appropriate task-specific latent representations and the mapping to relevance decisions. T5 can exploit the portion of the network responsible for generating output, unlike the fully-connected layer in the classification-based approach. This neural architecture contains dormant knowledge of semantics, linguistic relations, and other characteristics required to generate coherent text. In other words, T5 has access to an additional source of information that BERT does not. In order to verify this theory in the claim verification task, we have experimented with varying label words and monitored the overall accuracy gained on the dev set. It can be observed that, varying label words indeed affects the accuracy and we gain the highest percentage when finetuned with a more suitable words. And if we change the label words to irrelevant order, the accuracy drops. The accuracy drops and gains are detailed in the table 4.8. We conducted experiments using the following variants for comparison:

- **"Reverse"**: In this variant, "not supported" indicates a supported claim and "supported" indicates a claim that is not supported. By doing so, we intended to determine whether or not the model relied on latent knowledge regarding linguistic relationships. If the model did indeed exploit such knowledge, we hypothesized that compelling it to make opposite associations on the same polarity scale would reduce its efficacy relative to the baseline.

- **"Antonyms"**: Here, we mapped a supported claim to the term "hot" and an unsupported claim to the term "cold." This mapping maintained the use of adjectives at opposite extremes of a polarity scale; however, the polarity scale itself had no relevance. We anticipated the model's efficacy to be lower than the baseline if it were genuinely utilizing latent knowledge.

- **"Related Words"**:In this variant, we designated the term "apple" to a supported claim and the term "orange" to a claim that is not supported. Despite the fact that these words were semantically related, they lacked the polarity contrast of the previous variants. As a result, we anticipated that the efficacy would be inferior to the baseline.

- **"Unrelated Words"**: For this variant, we associated the term "hot" with a supported claim and a wholly unrelated term, "orange," with a claim that is not supported. Thus, we forced the model to generate an arbitrary semantic mapping. We anticipated that the performance would be inferior to both the baseline and the use of related terms.

4.3.4 Error Analysis

In this section, we conduct some error analyses of our proposed prompt-based approaches (fine-tuned and zero-shot) for the claim verification task. Below, we discussed the errors produced by these models.

(i) Error analysis of the best performing fine-tuned model: The supervised T5-large model, which achieved the highest accuracy when fine-tuned on the training set, encounters difficulties and tends to generate incorrect outputs in some complex situations. Some of these failures are exemplified in Table 4.9. In Sample 1, the model failed to determine which individual died first, given their respective lifespans. A potential failure point in this instance may be the notation used to indicate the authors' lifespans in the example. As the lifespan notation may vary in various example scenarios, it is challenging for the models to determine the correct response for this type of examples. In Sample 4, the model was unable to perform a comparison between two numbers, indicating its inability to make mathematical inferences. In Sample 5, the model faced challenges in making a long reasoning connection between two distantly mentioned named entities in a passage. Similarly, in Samples 3 and 4, a complex causal reasoning chain was required to be understood between multiple named entities, which the model struggled to comprehend.

(ii) Error analysis of the best performing zero-shot model: We picked a subset of cases in which the best performing zero-shot model `text-davinci-003` model failed to classify the claims correctly and carried out an investigation into those cases. We presented the error analysis in Table 4.10 that contains five examples where the model was unable to correctly predict the accurate class. In samples 3 and 4, mathematical reasoning was needed to correctly identify the veracity of the given claim, and clearly `text-davinci-003` model failed in this case. Furthermore, the model was unable to categorize the gold labels in Samples 1 and 2, since doing so would have

necessitated the use of a sophisticated multi-hop reasoning procedure to validate the claims' authenticity. As for the final example in Table 4.10, in addition to two-hop reasoning, it was needed to correctly infer between named entities and their respective pronouns, which proved to be challenging for the model.

#	Claim and Evidence	Gold Label	Predicted Label
1	**Claim:** Vladimir Igorevich Arnold died after Georg Cantor. **Evidence:** Georg Ferdinand Ludwig Philipp Cantor (March 3 (O.S. February 19) 1845 – January 6, 1918) was a German mathematician. Vladimir Igorevich Arnold (alternative spelling Arnol'd, 12 June 1937 – 3 June 2010) was a Soviet and Russian mathematician.	S	NS
2	**Claim:** Barton Mine was halted by a natural disaster not Camlaren Mine. **Evidence:** Barton Mine, also known as Net Lake Mine, is an abandoned surface and underground mine in Northeastern Ontario, Canada. Conditions attributed to World War II halted development at Camlaren in 1939. Barton was the site of a fire in the early 1900s, after which it never had active mining again.	S	NS
3	**Claim:** The product lithium-ion tank is being built at a 107,000 acre industrial park for Tesla Motors. **Evidence:** The Gigafactory 1 is being built there to serve Tesla Motors and Panasonic. The Tahoe Reno Industrial Center (TRI Center, or TRIC) is a privately owned 107,000 acre industrial park, located at Interstate 80 next to Clark, Storey County, Nevada. The Tesla Gigafactory 1 is an operational lithium-ion battery factory under construction, primarily for Tesla Inc., at the Tahoe Reno Industrial Center (TRIC) in Storey County (near the Community of Clark, Nevada, US).	S	NS
4	**Claim:** The operas Vanessa and Le roi malgré lui contain different number of acts. **Evidence:** Le roi malgré lui ("King in Spite of Himself" or "The reluctant king") is an opéra-comique in three acts by Emmanuel Chabrier of 1887 with an original libretto by Emile de Najac and Paul Burani. Vanessa is an American opera in three (originally four) acts by Samuel Barber, opus 32, with an original English libretto by Gian-Carlo Menotti.	NS	S
5	**Claim:** William McGrath was a loyalist from Northern Ireland, but was known for being in favor of a united Ireland. **Evidence:** William McGrath was a loyalist from Northern Ireland who founded the far-right organisation Tara in the 1960s, having also been prominent in the Orange Order until his expulsion due to his paedophilia. Like unionists, loyalists are attached to the British monarchy, support the continued existence of Northern Ireland, and oppose a united Ireland.	NS	S

Table 4.9: Sample Cases of misclassification made by T5 model. Here `NS` and `S` stand for NOT SUPPORTED class and SUPPORTED class respectively.

#	Claim and Evidence	Gold Label	Predicted Label
1	**Claim:** Happily was co-written and sang by the One Direction band member who got his debut as a singer for the band, White Eskimo. **Evidence:** It was co-written by band member Harry Styles. He made his debut as a singer with his band White Eskimo, who performed locally in Holmes Chapel, Cheshire.	NS	S
2	**Claim:** William Zabka, who was born on October 21, 1965, appeared in the 2014 American drama film Where Hope Grows. **Evidence:** The film stars David DeSanctis, Danica McKellar, Kerr Smith, Brooke Burns, William Zabka, Kristoffer Polaha and McKaley Miller. William Michael Zabka (born October 21, 1965) is an American actor, screenwriter, director and producer.	NS	S
3	**Claim:** Erich Schmidt-Leichner was the defense counsel of a German citizen who underwent Catholic exorcism rites during the year before her death. **Evidence:** Erich Schmidt-Leichner (14 October 1910 – 17 March 1983) was a German lawyer who made a name as a distinguished defense counsel at the Nuremberg Trials (1945 - 1946). In 1978, he was a defense counsel in the "Klingenberg Case" (Anneliese Michel), where a married couple were accused of negligent homicide for failing to call a medical doctor during an exorcism of their daughter. Anneliese Michel (21 September 1952 – 1 July 1976) was a German woman who underwent Catholic exorcism rites during the year before her death.	NS	S
4	**Claim:** The writer of the novel "Horizon" is American. They are younger than the author of "Dubin's Lives". **Evidence:** Bernard Malamud (April 26, 1914 – March 18, 1986) was an American novelist and short story writer. Lois McMaster Bujold (; born November 2, 1949) is an American speculative fiction writer. Dubin's Lives is the seventh published novel by the American writer Bernard Malamud.Horizon is a fantasy novel by American writer Lois McMaster Bujold.	S	NS
5	**Claim:** The author of Anastasia on Her Own won the 2002 Rhode Island Children's Book Award. **Evidence:** Anastasia on Her Own (1985) is a young-adult novel by Lois Lowry. Her book "Gooney Bird Greene" won the 2002 Rhode Island Children's Book Award.	S	NS

Table 4.10: Sample Cases of misclassification made by the `text-davinci-003` model. Here `NS` and `S` stand for NOT SUPPORTED class and SUPPORTED class respectively.

Chapter 5

Conclusion

5.1 Summary

In this comprehensive dissertation, we present a groundbreaking approach that leverages prompt-based language models to tackle the challenging task of claim verification. Our aim was to explore the potential benefits of incorporating language prompts of high quality into the existing framework and to investigate how they could enhance the performance of language models in handling multi-hop claim verification problems. Through extensive experiments, we evaluated our approach and found that incorporating carefully designed language prompts as additional input information significantly improved the performance of language models, particularly in the complex domain of multi-hop claim verification. Furthermore, Our prompt-based methods were rigorously tested and validated in diverse scenarios, including few-shot learning, cross-domain generalization, and zero-shot learning, demonstrating superior accuracy compared to conventional approaches and highlighting the robustness and versatility of prompt-based language models in handling various data availability and domain adaptation challenges. We also explored the potential of prompt-based generative decoder-based language models, specifically the highly capable `text-davinci-003` model. We discovered promising opportunities for utilizing generative large language models in claim verification research by analyzing the generated responses and their alignment with the prompts. Within the scope of our investigation, we also aimed to find a list of the best language prompts. Through meticulous analysis and experimentation, we successfully filtered out the top-performing prompts. This curated list provides valuable insights into the optimal use of prompts and is a practical resource for researchers and practitioners working in claim verification. Furthermore, our exploration extended beyond prompts to encompass the impact of label words in the generated texts. We conducted empirical analyses to investigate how the choice of label words influences the model's performance when mapping them to specific class labels. Our findings

revealed that attempting to map irrelevant words to specific class labels drastically decreased the model's performance. This emphasizes the importance of careful consideration when selecting appropriate label words for accurate classification within the claim verification framework. Overall, Our dissertation highlights the effectiveness of prompt-based language models in claim verification and their potential for advancing natural language understanding and information verification.

5.2 Future Works

In future endeavors, our work can be extended to tackle the more challenging task of claim validation using extracted evidence sentences obtained from a retrieval system i.e. the system will be robust enough to automatically extract evidences from a large corpus and providing filtered evidence sentences will no longer be necessary. Additionally, we aim to explore the impact of domain adaptation [44] and transfer learning [45] from answer selection models [46–48] on overall performance, thereby enhancing the versatility of our approach. Although in this study, we successfully employed manually generated language prompts, we acknowledge that this manual process can be laborious and complex in different scenarios. Therefore, an intriguing direction for future research would involve developing an automated system capable of generating suitable language prompts that can match the effectiveness of expert-designed prompts. Additionaly, In this dissertation, we focused on a specific evaluation metric used by all the baselines. However, for future research, it would be valuable to explore additional evaluation metrics, such as the Precision, Reacall and F1 measure, to conduct a comprehensive analysis and determine if the current evaluation is sufficient or requires further investigation. Furthermore, expanding our analysis, we intend to investigate the effect of prompt length on model accuracy and scalability. By conducting in-depth research, we aim to identify an optimal trade-off that balances prompt length with performance. This analysis will provide valuable insights into the impact of prompt size on the overall efficiency and effectiveness of claim verification models. By venturing into these future research directions, we seek to advance the field of claim validation and further optimize the performance of language models. Through automated prompt generation and a thorough analysis of prompt length, we aspire to enhance the efficiency, and accuracy of claim verification systems in various real-world applications.

REFERENCES

[1] G. Bekoulis, C. Papagiannopoulou, and N. Deligiannis, "A review on fact extraction and verification," *ACM Computing Surveys (CSUR)*, vol. 55, no. 1, pp. 1–35, 2021.

[2] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal, "Fever: a large-scale dataset for fact extraction and verification," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 809–819.

[3] D. Stammbach and G. Neumann, "Team domlin: Exploiting evidence enhancement for the fever shared task," in *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, 2019, pp. 105–109.

[4] Y. Jiang, S. Bordia, Z. Zhong, C. Dognin, M. Singh, and M. Bansal, "Hover: A dataset for many-hop fact extraction and claim verification," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, pp. 3441–3460.

[5] J. Chen, R. Zhang, J. Guo, Y. Fan, and X. Cheng, "Gere: Generative evidence retrieval for fact verification," *arXiv preprint arXiv:2204.05511*, 2022.

[6] A. Soleimani, C. Monz, and M. Worring, "Bert for evidence retrieval and claim verification," in *European Conference on Information Retrieval*. Springer, 2020, pp. 359–366.

[7] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[8] T. Gao, A. Fisch, and D. Chen, "Making pre-trained language models better few-shot learners," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association

for Computational Linguistics, Aug. 2021, pp. 3816–3830. [Online]. Available: https://aclanthology.org/2021.acl-long.295

[9] X. Han, W. Zhao, N. Ding, Z. Liu, and M. Sun, "PTR: prompt tuning with rules for text classification," *CoRR*, vol. abs/2105.11259, 2021. [Online]. Available: https://arxiv.org/abs/2105.11259

[10] R. Seoh, I. Birle, M. Tak, H.-S. Chang, B. Pinette, and A. Hough, "Open aspect target sentiment classification with natural language prompts," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 6311–6322. [Online]. Available: https://aclanthology.org/2021.emnlp-main.509

[11] C. Li, F. Gao, J. Bu, L. Xu, X. Chen, Y. Gu, Z. Shao, Q. Zheng, N. Zhang, Y. Wang *et al.*, "Sentiprompt: Sentiment knowledge enhanced prompt-tuning for aspect-based sentiment analysis," *arXiv preprint arXiv:2109.08306*, 2021.

[12] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[13] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. [Online]. Available: https://aclanthology.org/D14-1162

[14] Y. Bengio, R. Ducharme, and P. Vincent, "A neural probabilistic language model," *Advances in neural information processing systems*, vol. 13, 2000.

[15] A. Sherstinsky, "Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network," *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, 2020.

[16] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[20] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, pp. 1–67, 2020.

[21] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, "Improving language understanding by generative pre-training," 2018.

[22] D. Chen, A. Fisch, J. Weston, and A. Bordes, "Reading Wikipedia to answer open-domain questions," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 1870–1879. [Online]. Available: https://aclanthology.org/P17-1171

[23] A. Hanselowski, H. Zhang, Z. Li, D. Sorokin, B. Schiller, C. Schulz, and I. Gurevych, "Ukp-athene: Multi-sentence textual entailment for claim verification," in *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, 2018, pp. 103–108.

[24] Q. Chen, X. Zhu, Z.-H. Ling, S. Wei, H. Jiang, and D. Inkpen, "Enhanced lstm for natural language inference," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 1657–1668.

[25] D. Sorokin and I. Gurevych, "Mixing context granularities for improved entity linking on question answering data across entity categories," in *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 65–75. [Online]. Available: https://aclanthology.org/S18-2007

[26] Y. Nie, H. Chen, and M. Bansal, "Combining fact extraction and verification with neural semantic matching networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 6859–6866.

[27] O. Khattab, C. Potts, and M. Zaharia, "Baleen: Robust multi-hop reasoning at scale via condensed retrieval," *Advances in Neural Information Processing Systems*, vol. 34, pp. 27 670–27 682, 2021.

[28] J. Zhou, X. Han, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Gear: Graph-based evidence aggregating and reasoning for fact verification," *arXiv preprint arXiv:1908.01843*, 2019.

[29] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *arXiv preprint arXiv:2107.13586*, 2021.

[30] M. E. Peters, S. Ruder, and N. A. Smith, "To tune or not to tune? adapting pretrained representations to diverse tasks," in *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, 2019, pp. 7–14.

[31] T. Gao, "Prompting: Better ways of using language models for nlp tasks," *The Gradient*, 2021.

[32] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, and A. Miller, "Language models as knowledge bases?" in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 2463–2473. [Online]. Available: https://aclanthology.org/D19-1250

[33] J. Davison, J. Feldman, and A. M. Rush, "Commonsense knowledge mining from pretrained models," in *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, 2019, pp. 1173–1178.

[34] Z. Jiang, F. F. Xu, J. Araki, and G. Neubig, "How can we know what language models know?" *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 423–438, 2020. [Online]. Available: https://aclanthology.org/2020.tacl-1.28

[35] A. Talmor, Y. Elazar, Y. Goldberg, and J. Berant, "oLMpics-on what language model pre-training captures," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 743–758, 2020. [Online]. Available: https://aclanthology.org/2020.tacl-1.48

[36] T. Schick and H. Schütze, "Exploiting cloze questions for few-shot text classification and natural language inference," *CoRR*, vol. abs/2001.07676, 2020. [Online]. Available: https://arxiv.org/abs/2001.07676

[37] T. L. Scao and A. M. Rush, "How many data points is a prompt worth?" *arXiv preprint arXiv:2103.08493*, 2021.

[38] S. Hu, N. Ding, H. Wang, Z. Liu, J. Wang, J. Li, W. Wu, and M. Sun, "Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification," *arXiv preprint arXiv:2108.02035*, 2021.

[39] Z. Zhao, E. Wallace, S. Feng, D. Klein, and S. Singh, "Calibrate before use: Improving few-shot performance of language models," in *International Conference on Machine Learning*. PMLR, 2021, pp. 12 697–12 706.

[40] N. Ding, S. Hu, W. Zhao, Y. Chen, Z. Liu, H. Zheng, and M. Sun, "Openprompt: An open-source framework for prompt-learning," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2022, pp. 105–113.

[41] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[42] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," in *International Conference on Learning Representations*, 2019.

[43] R. Nogueira, Z. Jiang, R. Pradeep, and J. Lin, "Document ranking with a pretrained sequence-to-sequence model," in *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020, pp. 708–718. [Online]. Available: https://aclanthology.org/2020.findings-emnlp.63

[44] M. T. R. Laskar, E. Hoque, and J. X. Huang, "Domain adaptation with pre-trained transformers for query-focused abstractive text summarization," *Computational Linguistics*, vol. 48, no. 2, pp. 279–320, Jun. 2022. [Online]. Available: https://aclanthology.org/2022.cl-2.2

[45] M. T. R. Laskar, E. Hoque, and J. Huang, "Query focused abstractive summarization via incorporating query relevance and transfer learning with transformer models," in *Advances in Artificial Intelligence: 33rd Canadian Conference on Artificial Intelligence, Canadian AI 2020, Ottawa, ON, Canada, May 13–15, 2020, Proceedings 33*. Springer, 2020, pp. 342–348.

[46] S. M. S. Ekram, A. A. Rahman, M. S. Altaf, M. S. Islam, M. M. Rahman, M. M. Rahman, M. A. Hossain, and A. R. M. Kamal, "Banglarqa: A benchmark dataset for under-resourced bangla language reading comprehension-based question answering with diverse question-answer types," in *Findings of the Association for Computational Linguistics: EMNLP 2022*, 2022, pp. 2518–2532.

[47] M. T. R. Laskar, E. Hoque, and J. X. Huang, "WSL-DS: Weakly supervised learning with distant supervision for query focused multi-document abstractive summarization," in *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 5647–5654. [Online]. Available: https://aclanthology.org/2020.coling-main.495

[48] M. T. R. Laskar, X. Huang, and E. Hoque, "Contextualized embeddings based transformer encoder for sentence similarity modeling in answer selection task," in *Proceedings of the Twelfth Language Resources and Evaluation Conference*, 2020, pp. 5505–5514.

List of Publications

Conference

1. **Md Mezbaur Rahman**, Mohammed Saidul Islam, Md Tahmid Rahman Laskar, Md Azam Hossain and Abu Raihan Mostofa Kamal, "Multihop Factual Claim Verification Using Natural Language Prompts". To appear in *Proceedings of The 36th Canadian Conference on Artificial Intelligence (CANAI)*