Automated Fact Extraction and Verification for Detecting False Information

by

Shyam Subramanian

A Thesis

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Master of Science

in

Data Science

by

August 2020

APPROVED:

Professor Kyumin Lee, Thesis Advisor

Professor Mohamed Y. Eltabakh, Reader

Abstract

Automated fact extraction and verification is a challenging task that involves *finding relevant evidence sentences* from a reliable corpus to *verify the truthfulness of a claim*. Existing models either (i) concatenate all the evidence sentences, leading to the inclusion of unnecessary sentences containing redundant, distracting, noisy or irrelevant information; or (ii) process each claim-evidence sentence pair separately and aggregate all of them later, missing the early combination of related sentences for more accurate claim verification. Unlike the prior works, in this thesis, we propose Hierarchical Evidence Set Modeling (HESM), a framework to extract evidence sets (each of which may contain multiple evidence sentences) and verify a claim to be SUPPORTED, REFUTED, or NOT ENOUGH INFO, by encoding and attending the claim-evidence set pairs at different levels of hierarchy. Each evidence set combines only the related sentences while limiting unnecessary sentences. Thus, our HESM framework overcomes the limitations of existing models that concatenates evidence sentences or aggregates individual claim-evidence sentence pairs. HESM consists of document retriever, multi-hop evidence retriever, and claim verification components. In the framework, we extract multiple evidence sets, and process and evaluate a claim based on each evidence set. Then, we aggregate all the evidence sets using word-level and evidence set-level attention for final verification of the claim. Our experimental results show that HESM outperforms 7 state-of-the-art methods for fact extraction and claim verification.

Acknowledgements

I would like to express my sincere gratitude to my advisor, Professor Kyumin Lee, for his valuable guidance and contributions throughout my thesis work. I would like to thank Professor Mohamed Y. Eltabakh for being a part of the thesis committee. I would also like to acknowledge the members of InfoLab at Worcester Polytechnic Institute, who constantly shared their knowledge. Lastly, I would like to thank my parents and friends for their motivation and support.

Contents

1 Introduction **1**

1.1 Motivation and Challenges 3

1.1.1 Concatenation 4

1.1.2 Evidence sentence level processing 7

1.1.3 Proposed evidence set level processing 8

1.2 Contribution 9

1.3 Problem Definition 10

1.4 Remainder of the Document 11

2 Background **12**

2.1 Similar Tasks 12

2.1.1 Natural Language Inference 13

2.1.2 Stance Detection 13

2.1.3 Fake News Detection 14

2.1.4 Question Answering 14

2.2 Existing Fact Verification Models 15

2.2.1 Document retrieval 15

2.2.2 Evidence sentence retrieval 16

2.2.3 Claim verification 17

2.3 Modeling Natural Language 18
2.3.1 Word Embeddings 18
2.3.2 Attention 20

3 Methodology **21**

3.1 Hierarchical Evidence Set Modeling 21
3.2 Document Retriever 22
3.3 Multi-hop Evidence Retriever 23
3.4 Claim Verification 26
3.4.1 Evidence Set Modeling Block 27
3.4.2 Hierarchical Aggregation Modeling 30
3.4.3 Training Loss and Inference 32

4 Experimentation and Results **33**

4.1 Experiment Setting 33
4.1.1 Dataset 33
4.1.2 Baselines 34
4.1.3 Evaluation Metrics 35
4.1.4 Implementation and Training details 36
4.2 Experimental Results and Analysis 37
4.2.1 Multi-hop evidence retrieval 38
4.2.2 Claim verification 39
4.2.3 Aggregation Analysis 39
4.2.4 Ablation Study 40
4.2.5 Contextual and Non-contextual Aggregations 40
4.2.6 Impact of noisy evidence sentences on Concat and HESM models. 44

5 Future Work **46**

5.1 Joint Learning 46

5.2 Alternative Approaches 47

5.3 Fake News Detection 48

5.4 Robustness 49

6 Conclusion **50**

A Resources **51**

A.1 Links to resources 51

List of Figures

1.1 An example of claim, evidence and verdict. 2

1.2 Types of evidence combinations 4

1.3 Ideal verification system 8

3.1 Our HESM framework. 22

3.2 Multi Hop Evidence Retrieval. 23

3.3 Evidence Set Modeling Block. 26

3.4 Attention Sum Block. 29

3.5 Hierarchical Aggregation. 30

4.1 Performance of contextual and non-contextual aggregations given different claim labels. 41

4.2 Performance of contextual and non-contextual aggregations given claims requiring different number of evidence sentences. 43

4.3 Evidence-set level attention in Contexual and Non-contextual aggregation 44

4.4 Performance comparison of Concat and HESM with different noise evidence sentences retrieved 45

List of Tables

1.1 Different kinds of information found in evidence sentences. 6

1.2 Claim requiring composition of multiple evidence sentences. 7

4.1 Statistics of FEVER Dataset. 34

4.2 Evidence retrieval performance of the baselines and our model in development set. 38

4.3 Performance of the baselines and our model in test set. 38

4.4 Claim verification with different aggregation methods in development set. 39

4.5 Ablation analysis in development set. 40

4.6 Contextual and Non-contextual aggregation attention metrics 43

A.1 Links to resources 51

Chapter 1

Introduction

The large amount of user-generated content being produced and consumed on the web has facilitated the growth and spread of false and inaccurate information. With the web being the go-to place for every query, this diffusion of false information poses a serious threat to its audience. Unfortunately, the rise of social media has further accelerated the communication and propagation of unverified information. A study has revealed that 60% people on social media share news articles after reading just the title, skipping to read the content of the news [1]. False information spreads faster, deeper and broader than the truth in twitter, a micro-blogging social network service [2]. Moreover, even though the majority of people mistrust information in social media, they continue to share the information [3]. The presence of enormous volume of information and the alarming rate at which false information spreads, renders manual verification impossible and makes automation of fact verification inevitable. Towards this end, our work focuses on automated fact extraction and verification task, which requires automatically retrieving evidences related to a claim from a large corpus, and then verifying the claim based on the evidences retrieved. The complete evidence to support or refute a claim might be present in a single

Claim: Janelle Monáe is signed to Warner Music Group.

Evidence set:

[**wiki/Janelle Monáe**]
Janelle Monáe Robinson (born December 1, 1985) is an American singer, songwriter, actress, and model signed to her own imprint, Wondaland Arts Society, and Atlantic Records.

[**wiki/Atlantic Records**]
In 1967, Atlantic Records became a wholly owned subsidiary of Warner Bros.-Seven Arts, now the Warner Music Group, and expanded into rock and pop music with releases by bands such as Led Zeppelin and Yes.

Verdict: Supported

Figure 1.1: An example of claim, evidence and verdict.

sentence or across multiple sentences. These sentences, in turn, might be present in a single or multiple document(s). In Figure 1.1, we show an example of a claim, evidence and its verdict. For the claim *"Janelle Monáe is signed to Warner Music Group"*, the system has to retrieve the relevant evidence sentences from two documents namely *Janelle Monáe* and *Atlantic Records*. Then, the system has to arrive at a verdict as *Supported*, based on the context of both the sentences. Specifically, the system has to understand that *Janelle Monáe is signed to Atlantic Records* (from the first sentence), which in turn is a *subsidiary of the Warner Music Group* (from the second sentence) and infer that *Janelle Monáe* is in fact signed to *Warner Music Group*. In this work, we refer to the set of sentences that can verify the truthfulness of a claim as an evidence set. In general, it is possible to find multiple evidence sets for a claim in the corpus, where each evidence set can either support or refute a claim.

1.1 Motivation and Challenges

Automated fact extraction and verification is important in different contexts of information including journalism [4], community forums [5], information extraction or question answering systems [6] such as search engines[1] and personal assistants[2], scientific publications [7], crowd-sourced information, product reviews and several others. Fact verification in journalism, also known as Fake News Detection [8], has gained the most attention after the potential influence of fake news in the 2016 US presidential election [9]. Many works have been published for automating fake news detection [10, 11, 12, 13]. Fake news detection is a complicated task by itself which includes finding fact-check worthy claims, assessing the credibility of the news source, finding evidences and verifying the claims. The difficulty in automation of fake news detection is much more complicated due to the lack of large scale fact-checked fake news, fine grained annotations for evidences and reliable factual source at a single place. The fact verification task studied in this thesis is comparatively simpler where the evidence knowledge base is a single reliable source such as Wikipedia or Freebase, and the claims can be verified by a small number of evidence sentences. Nevertheless, this work can be an efficient step towards fully automated fake news detection. Also, the learned models are transferable to fact verification in different contexts of information, using transfer learning strategies.

The task is challenging since it requires semantic understanding and reasoning to learn the subtleties that differ between evidences that support and evidences that refute a claim. The difficulty of the task is further amplified for claims that do not have complete evidence that can verify their truthfulness (i.e. claims with NOT ENOUGH INFO label) and for claims that require aggregating information

[1] https://www.blog.google/outreach-initiatives/google-news-initiative/how-we-highlight-fact-checks-search-and-google-news/?

[2] https://reporterslab.org/fact-checking-comes-amazon-echo/

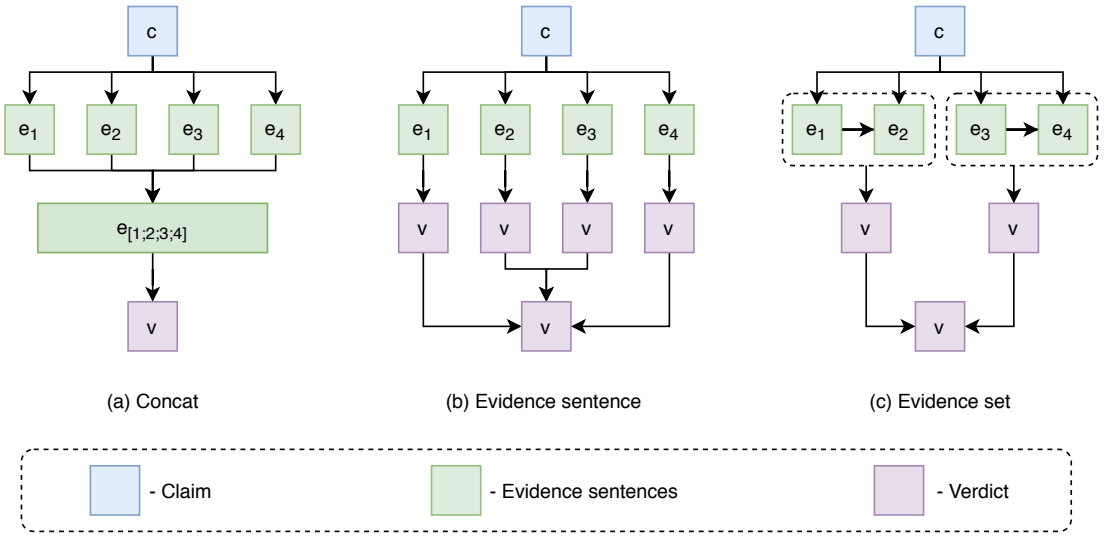

Figure 1.2: Types of evidence combinations.

from multiple evidence sentences in different documents. Previous works in fact verification use different strategies for combining the evidence sentences retrieved for a claim. They either operate by concatenating all the evidence sentences together [14] or they operate at each evidence sentence level and aggregate them later [15, 16]. In Figure 1.2, we show the different strategies previously used along with proposed strategy. We will discuss these strategies, their issues and how our proposed strategy overcomes these challenges, briefly below.

1.1.1 Concatenation

Figure 1.2 (a) shows the concatenation of the evidence sentences retrieved. The evidence sentences e_1 to e_4 are concatenated to form a single evidence set $e_{[1;2;3;4]}$ and a claim c is verified using the evidence set to produce a verdict v (SUPPORTS/REFUTES/NOT ENOUGH INFO). Concatenating all the sentences together may lead to redundant, noisy, distracting and irrelevant information being combined with the relevant information. In Table 1.1 we show these different kinds of information present in evidence sentences.

Redundant information. In the claim (1), *"Kuching is in the state of Johor"*, the relevant evidence sentence (1) has information that states *"Kuching is the capital of Sarawak state"*, while the redundant evidence sentence (2) also has the same information *"Kuching is the situated in the state of Sarawak"*. Note that, the vice-versa, when sentence (2) is considered relevant, the sentence (1) becomes redundant. While concatenating these sentences to predict a single verification label for the claim might not affect the verification accuracy, predicting a verification label for the claim based on each sentence separately can boost the confidence of the verification since each sentence can correctly verify the claim individually.

Noise information. In the claim (2), *"Tenacious D was released before 2006"*, the relevant evidence sentence (1) has information that states *"Tenacious D released in 2001"*, while the noisy evidence sentence (2) has the information *"Tenacious D was certified platinum in 2005"*. We term it as noisy evidence sentence since it can contain confounding information which can make claim verification complicated even though it does not have redundant or irrelevant information. Note that the evidence sentence (2) can still verify the claim individually. Similar to redundant information, while concatenating relevant with noisy sentences might not affect the verification accuracy, predicting a verification label for the claim based on each sentence separately can boost the confidence of the verification.

Distracting information. In the claim (3), *"Gal Gadot was ranked behind Esti Ginzburgh"*, the relevant evidence sentence (1) has information that states *"Gal Gadot was ranked ahead of Esti Ginzburg"*, while the distracting evidence sentence (2) has the information *"Gal Gadot was ranked behind Bar Refaeli"*. It is distracting since it can confuse the model in understanding *who is ahead and who is behind in the rankings*. Note that the distracting sentence does not contradict the information

Claim	**Evidence type**	**Evidence sentence**
(1) Kuching is in the state of Johor	Relevant	(1) Kuching, officially the City of Kuching, is the capital of Sarawak state in Malaysia
	Redundant	(2) Kuching city is situated on the Sarawak River at the southwest tip of the state of Sarawak
(2) Tenacious D was released before 2006	Relevant	(1) Tenacious D album released on September 25, 2001
	Noise	(2) Tenacious D album was certified platinum by the Recording Industry Association of America by the end of 2005
(3) Gal Gadot was ranked behind Esti Ginzburgh for highest earning actress/models in Israel	Relevant	(1) Gal Gadot was ranked ahead of Esti Ginzburg and Shlomit Malka, in highest earning actress/models in Israel
	Distracting	(2) Gal Gadot was ranked as the second highest earning actress/models in Israel, behind Bar Refaeli
	Irrelevant	(3) According to Forbes Israel, she was among the top ten highest paid models in Israel

Table 1.1: Different kinds of information found in evidence sentences.

in relevant evidence sentence. Concatenating these evidence sentences, makes claim verification more complicated in terms of identifying and learning the context of only the relevant sentences. This was observed in the work [17], which shows that the state-of-the-art models in a similar semantic understand task (question answering) are not robust and the performance of the models drop significantly when distracting sentences are added. While making the model more robust despite the presence of distracting sentences is the expected approach, it is much more challenging and is a study of its own. Therefore, in this work, we only focus on omitting distracting sentences from combining with relevant sentences as much as possible and leave

Claim	Evidence type	Evidence sentence
(1) Wentworth is an Australian television series	Relevant	(1) Wentworth was first broadcast on SoHo on 1 May 2013
	Relevant	(2) SoHo was an Australian cable and satellite channel

Table 1.2: Claim requiring composition of multiple evidence sentences.

improving the robustness of the model as a future work.

Irrelevant information. Finally, in the claim (3), *"Gal Gadot was ranked behind Esti Ginzburgh"*, the irrelevant evidence sentence (3) has information that states *"Gal Gadot was among top ten highest paid models"*. This sentence does not contain any information that can support or refute the claim and thus it is irrelevant to the claim. Similar to distracting information, this can make claim verification more complicated in terms of identifying and learning the context of only the relevant sentences.

1.1.2 Evidence sentence level processing

Figure 1.2 (b) shows the processing of each claim-evidence sentence pair separately. The claim c is verified based on each evidence sentence e_1 to e_4 individually and then produces a final verdict v (SUPPORTS/REFUTES/NOT ENOUGH INFO) by aggregating the verdicts from all the evidence sentences. In Table 1.2, we show an example of a claim requiring composition of multiple evidence sentences. In the claim (1), *"Wentworth is an Australian television series"*, the relevant evidence sentence (1) has information that states *"Wentworth was first broadcast on SoHo"*, while the relevant evidence sentence (2) has the information *"SoHo was an Australian cable and satellite channel"*. Here, neither of the sentences can verify the

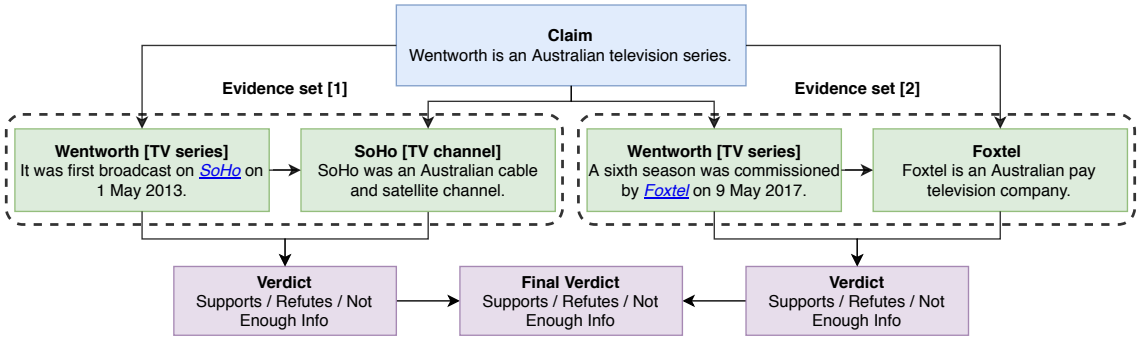

Figure 1.3: Ideal verification system. The arrows represent the hierarchy of the fact extraction and verification process.

claim individually, but can verify the claim when they are combined. Therefore, the system has to arrive at a verdict based on the context of both the sentences. Specifically, the system has to understand that *"Wentworth was broadcast on SoHo which is an Australian channel"*. While individually processing each evidence sentence overcomes many of the issues in concatenation strategy, it delays the combination of these relevant sentences which belong to the same evidence set. This makes claim verification harder since it summarizes information without complete context and then aggregates the summarized information of all the sentences.

1.1.3 Proposed evidence set level processing

Figure 1.3 depicts an example of an ideal verification system, which extracts evidence sets, processes them individually, and then aggregates them later. For better understanding of the comparison between the proposed strategy and the previous strategies, we also show a corresponding minimal version of the Figure 1.3 in Figure 1.2 (c). In the example, four evidence sentences are retrieved. Sentences which are relevant and hyperlinked, are combined to form evidence sets (called Evidence Set [1] and Evidence Set [2] in the figure). Each evidence set verifies the claim individually, and then they are aggregated for the final verification. This strategy

of combining evidence sentences into evidence sets can potentially get the best of both previous strategies (concatenation and evidence sentence level processing). We achieve this by combining only the sentences that belong to the same evidence set, which is identified by hyperlinks between the sentences. Then, we formulate two different aggregation strategies for combining multiple evidence sets to form a final verification label, which we explain in detail in the following sections. These two different aggregation strategies account for improved performance of our model by their division of labour in obtaining best of both the concatenation and evidence sentence level processing strategies. We also follow a training procedure that naturally allows for a division of labour between the two aggregation strategies, in verifying claims with NOT ENOUGH INFO label and SUPPORTS/REFUTES label.

1.2 Contribution

Like Figure 1.3, our proposed framework also retrieves and combines evidence sentences into evidence sets. Then, it processes each evidence set individually to form a representation of the evidence set using word-level attention. Then, it combines information from all the evidence set representations using contextual and non-contextual aggregation methods, which use evidence set-level attention. The word-level attention along with evidence set-level attention forms a hierarchical attention mechanism. Finally, our framework learns to verify the claim at different levels of hierarchy (i.e., at each evidence set-level and at the aggregated evidence level).

Our main contributions are as follows:

1. We propose Hierarchical Evidence Set Modeling which consists of document retriever, multi-hop evidence retriever and claim verification.

2. Our multi-hop evidence retriever retrieves evidence sentences and combines

them as evidence sets. Our claim verification component conducts the hierarchical verification based on each evidence set individually and then based on all the evidence sets combined.

3. Our analysis of the contextual and non-contextual aggregation methods shows that, each method has a different role in claim verification in terms of the verification labels and number of evidence sentences required for verification.

4. Our experimental results show that our model outperforms 7 state-of-the-art baselines in both evidence retrieval and claim verification.

1.3 Problem Definition

Given a set of $\boldsymbol{m}$ textual documents and a claim c_i, the problem is to find a set of evidence sentences $\hat{E}_i = \{s_1, s_2, ..., s_{|\hat{E}_i|}\}$ and classify the claim c_i as $\hat{y}_i \in \{S, R, NEI\}$ (i.e., SUPPORTED, REFUTED or NOT ENOUGH INFO). For a successful verification of the claim c_i, there are two conditions: (1) $\hat{E}_i$ should match at least one evidence set E_i in the list of ground truth evidence sets and (2) $\hat{y}_i$ should match the ground truth entailment label y_i.

1.4 Remainder of the Document

The rest of the sections are structured as follows. Background and related work is provided in the section 2. Methodology is provided in chapter 3, where we discuss our strategy for Document retrieval in section 3.2, Multi-hop evidence retrieval in section 3.3 and Claim verification in section 3.4. We provide the experiment setting and results in chapter 4 where we discuss the dataset used, implementation and training process and results of the experiments and analysis. In chapter 5, we describe our future work and conclude in chapter 6. Lastly, we provide additional information of our implementation, code and other resources relevant to this work in the appendix.

Chapter 2

Background

In this chapter, we review the tasks similar to fact verification. Then, we briefly discuss the previous works in fact verification.

2.1 Similar Tasks

Several works in fact verification exist based on different forms of claim and evidence. Numerical claims are verified using subject-predicate-object triples from knowledge graph as evidence in [18, 19]. Claims in subject-predicate-object triple format are verified in [20, 21]. Textual claims are verified using evidences in a tabular format in [22]. In this work, we focus on fact verification using FEVER dataset [23], which consists of textual claims and evidences and requires both retrieval of correct evidences and claim verification based on the retrieved evidences. A thorough survey of the existing datasets and task formulations for fact checking and verification is provided in [24].

Fact verification has been studied in different natural language settings namely Recognizing Textual Entailment or Natural Language Inference [25, 26] and Stance Detection [27]. A differently motivated but closely related problem is fact checking

in journalism also known as fake news detection [11]. Question answering [28, 29] is another task that requires similar semantic understanding of natural language. All the tasks are related to each other in terms of understanding and reasoning about natural language and we benefit from using these works to refer best practices. In the following subsections, we explain each task briefly.

2.1.1 Natural Language Inference

Recognizing Textual Entailment (RTE) is a task that is defined as recognizing whether the meaning of one text can be inferred (entailed) from another. Natural Language Inference (NLI) task extends RTE to predict if a premise sentence entails, contradicts or is neutral to a given hypothesis sentence. There are a number of large annotated datasets such as SNLI [30] and MNLI [31]. These datasets are constructed with the intention of improving fundamental understanding and reasoning of natural language and not just facts. Different works exist to successfully improve textual entailment [32, 33, 34]. Fact verification task can be considered as an instance of NLI task for facts, which additionally requires evidence retrieval.

2.1.2 Stance Detection

Another closely related task is stance classification, which requires a model to predict the stance of an evidence sentence as favorable or against a claim sentence. Stance detection task is different from fact verification in that the same claim might have multiple stances from different entities, for example, stance detection is used to predict stances in debates [35], tweets [27], and different news media reports [36]. A more recent work [37] adds retrieval of evidence candidates as an additional task similar to evidence sentence retrieval in fact extraction and verification.

2.1.3 Fake News Detection

Fake news detection task requires prediction of the veracity (true or fake) of a piece of news. It can be thought of as stance detection task in the context of news and journalism, followed by veracity prediction based on the stance of different sources of evidence. Fake news detection has become an important area after the potential influence of fake news to the 2016 US presidential election [9]. Many fact-checking websites have emerged and aimed to educate the public regarding such fake news (ex.: Snopes, PolitiFact). They have also provided information that are manually verified by trained journalists with evidences obtained from trusted and unbiased sources. It has fueled the research in automation of fake news detection since the journalists created vast amount of ground truth data. Several datasets [11, 13, 38] and several works [39, 40, 41] exist that has accelerated the automation of fake news detection.

2.1.4 Question Answering

Question answering is a reading comprehension task that requires reading text and answering questions about it. Several types of question answering task exists such as multiple choice, Yes or No, Cloze style answering, and text span extraction amongst others. Open Domain Question Answering (ODQA) [42] requires retrieval of relevant documents and paragraphs of text that contains answers to the question and then answering the question itself. There are many large open domain question answering datasets such as SearchQA [43], MS-MARCO [44], and QUASAR-T [45]. Similar to unverifiable (NOT ENOUGH INFO) claims in fact verification task, the SQUAD 2.0 dataset [29] has unanswerable questions. Question answering is a well explored task. Numerous works exist to improve the performance, efficiency, robustness and speed

of the ODQA systems. Therefore, there is an abundance of literature available to understand what works and what aspect has not been studied yet in natural language understanding.

2.2 Existing Fact Verification Models

Previous works on the fact extraction and claim verification task follow a three stage pipeline that includes *document retrieval, evidence sentence retrieval and claim verification*. In the following sections, we briefly explain the strategies followed in previous works in each component of the pipeline.

2.2.1 Document retrieval

In the Document retrieval component, the potential documents relevant to the claim are retrieved. Since the textual corpus consists of a large number of documents and since most of them will be irrelevant to a claim, it is necessary to filter them out in an efficient manner. It is computationally infeasible to apply semantic matching approaches to filter the irrelevant documents due to the vast amount of documents present. Therefore, most previous works adopt traditional Information retrieval approaches such as Keyword matching, TF-IDF and Entity linking amongst others. Since the title of the documents usually contain the name of the entity about whose information is contained in the document, the claim is compared to the documents' titles to measure their relevance. The top performing systems [46, 15, 14] in the FEVER Shared Task 1.0 challenge [47] follow different approaches for Document Retrieval that set strong baselines for Document retrieval and other previous works followed suit. [46] follows an entity linking approach to match entities in the claim to the entities in titles of documents using the MediaWiki API. In [15], first the

documents whose title is present in the claim are retrieved. Then, it uses a feature-based logistic regression approach where statistical features (such as position and capitalization within the claims, presence of stop words, token match counts between first sentence of the documents and claim) are used from the claim and first sentence of the documents to predict their relevance to the claim. Finally, [14] uses a keyword matching technique to retrieve the documents whose title is present in the claim. Then, it uses a neural semantic matching network for resolving ambiguities in a title by predicting the relevance of the first sentence of the retrieved documents with the claim.

2.2.2 Evidence sentence retrieval

In Evidence sentence retrieval component, the most relevant sentences in the retrieved documents from the upstream document retrieval component are retrieved. To predict the relevance of the sentences, [15] uses a feature-based logistic regression model with features such as the position of the sentence within the document, its length, whether the document name is present in the sentence, token matching between the sentence and the claim, and the document retrieval score. Both [46] and [14] use supervised training with Enhanced LSTM (RNN based) [32] model proposed for Natural Language Inference task. Few recent works [48], [49], [50] and [51] use transformer [52] based pre-trained models. Evidence sentence retrieval component in most previous works retrieves all the evidences through a single iteration. Alternatively, [49] uses a multi-hop retrieval strategy through two iterations to retrieve evidence sentences that are conditioned on the retrieval of other evidence sentences. Then, all the top-most relevant evidence sentences with highest relevance scores are selected and combined into a single evidence set. Our work follows a similar strategy, but differs from the prior work by combining only evidence sentences that

belong to the same evidence set.

2.2.3 Claim verification

In claim verification component, for a given claim, the sentences retrieved from the upstream evidence sentence retrieval component are aggregated and classified as whether they support, refute or do not provide enough information about the claim. Thed [46] processes each claim-evidence sentenc top performing models [14, 15, 46] from the FEVER Shared Task 1.0, use a modified Enhanced LSTM [32] model proposed for Natural Language Inference task. [14] aggregates evidence sentences by concatenating them into a string, while [15] ane pair separately and aggregates their features or labels. [15] uses a Multi-Layer Perceptron model containing 2 hidden layers with 100 hidden units each for combining the classification probabilities for the three classes (SUPPORTS, REFUTES, NOT ENOUGH INFO) from all the sentences to form a final aggregated classification label. Similarly, [46] uses an attention based model, where each sentence is attended based on the aggregated representation of the claim from all the sentences, to obtain the relative importance of each sentence with the claim. Then, the sentences are aggregated based on their respective attention weights to form a final aggregated classification label. Recent works [48, 49, 53] use BERT based model [54] for claim verification. While [49] and [48] aggregates evidences by concatenation, [53] experiments with both concatenation and claim-evidence sentence pair aggregation. Few other works [53, 50] use graph based models for fine-grained semantic reasoning. Different from the previous works, our model operates with claim-evidence set pairs instead of claim-evidence sentence pairs. Our model benefits from encoding, attending and evaluating at different levels of hierarchy, as well as from both contextual and non-contextual aggregations of the evidence sets.

2.3 Modeling Natural Language

In recent years, Deep Learning has shown incredible success in several fields including Natural Language Processing, Computer Vision and Audio Analysis. Deep Learning models such as Convolutional Neural Networks (CNN) [55] and different variants of Recurrent Neural Networks (RNN) such as Long Short-Term Memory (LSTM) [56] and Gated Recurrent Unit (GRU) [57] have been predominantly used to model Natural Language. A more recent model called the Transformer [52] has shown promising results in different NLP tasks and has attracted the vast majority of NLP research community because of its potential for increased parallelism, allowing training of deeper models on several orders of magnitude larger datasets. The progress of Natural Language Understanding and Reasoning has been further accelerated by the tasks involving learning general representation of words, called *Word Embeddings*. An important architectural component that is a crucial part of several recent models is *Attention*. The following subsections will provide a brief explanation of different Word Embeddings and Attention mechanisms.

2.3.1 Word Embeddings

Traditional representation of words involved one hot encoding, bag of words, count vectorization, TF-IDF vectorization amongst others. These representations are independent of other words in the corpus and do not capture the semantics of the words. Rather, they capture only the statistical features such as the occurence or frequency of words across a large training corpora. Distributed representations, on the other hand, capture the semantics of words based on their surrounding context or co-occurrence with other words. Word Embeddings are distributed representation of words that maps words to real-valued distributed feature vectors. The word

embeddings are learned using self-supervised learning on large corpora of long documents such that the word vectors for relevant words have similar representation in the vector space. Word2Vec [58] and Glove [59] are well known examples of Word Embeddings and have been proven to be useful in representation of words. More recent advancements to Word Embeddings come from learning contextual representation of the words. Contextualized word embedding is required since words can have different meaning depending on their context. For example, *"bat"* refers to an object in the sentence *"A baseball bat"* and refers to an animal in the sentence *"Bats are awake at night"*. Embeddings from Language Models (ELMo) [60] is one of the early works in contextual word embedding, which is based on a modified version of RNN-LSTM called Highway Networks [61]. Since RNN based models only encode one word at a time, the models does not allow much parallelization. In order to train deeper models on longer and larger number of documents, recent contextualized word embedding models have adopted a more recently proposed Transformer model which is highly parallelizable. Several works have been published including Bidirectional Encoder Representations from Transformer (BERT) [54], Generative Pre-Training (GPT) [62], Transformer-XL [63], XL-Net [64] and several successors of BERT such as A Robustly Optimized BERT pretraining Approach (RoBERTa) [65], DistilBERT [66], Tiny BERT [67] and A Lite BERT (ALBERT) [68]. Tasks such as Masked Language Modeling, Next Sentence Prediction and Sentence order prediction amongst others are used for self supervised language representation training. Since these models are trained on large number of long documents such as documents from Wikipedia, they form a more generalizable way of representing language. As a result, the important advantage of these contextualized word embedding models are that they can be fine-tuned for downstream tasks providing better accuracy as well as faster training.

2.3.2 Attention

Attention is the mechanism of focusing on or attending to a particular segment of the input that is relevant and ignoring other segments of input that are not relevant. The two major types of attention are *Hard attention* and *Soft attention*. While *Hard attention* attends only to the important input segment, *Soft attention* attends to all the segments but with higher attention weights on the important input segment. Most architectures use soft attention since it can be trained using back propagation whereas hard attention cannot be trained using back propagation since it replaces a deterministic method with a stochastic sampling of the input segments. Attention mechanism was introduced in Natural Language Processing for Machine Translation task [69] to capture the alignment between the source and the target language (i.e.) to focus on the words important in the source language to the word being translated in the target language. Several ways exist to compute the attention weight such as Cosine-similarity attention [70], Additive attention [69], Multiplicative attention [71] and Dot product attention [71] amongst others. A difference between global and local attention was also introduced in [71]. Different variants of attention mechanism exist based on the source and the target for attention such as Two-way attention [72], Co-attention [73] and Self Attention [74]. Several multi-level attention mechanisms exists such as Attention over Attention [75], Iterative Attention [76], Hierarchical Attention [77] and a similarly motivated multi-hop memory model called the Memory Networks [78]. Attention mechanisms based on syntactic trees (constituency or dependency trees) have also been employed [79, 80, 81]. Other uses of attention include using attention as pointers [82], Attention flow [83] and Attention Propogation [84]. Finally, attention can also be used as an explanation under certain constraints [85]. Transformer model [52] is based on the Self Attention mechanism and it uses a scaled version of the Dot product attention.

Chapter 3

Methodology

In this chapter, we explain our methodology for fact extraction and verification using Hierarchical Evidence Set Modeling Framework.

3.1 Hierarchical Evidence Set Modeling

Our Hierarchical Evidence Set Modeling (HESM) framework consists of three components namely Document Retriever, Multi-hop Evidence Retriever and Claim Verification. Figure 3.1 shows an overview of our framework. The document retriever component retrieves the top K_1 documents that are relevant to the claim. The multi-hop retriever component retrieves the relevant top K_2 evidence sets from the K_1 retrieved documents via an iterative fashion. The claim verification component classifies the claim as SUPPORTS, REFUTES or NOT ENOUGH INFO based on the retrieved K_2 evidence sets. Following prior works, in our framework, we reuse the document retriever component from [14] which works well in terms of relevant document retrieval. We mainly focus on and propose novel multi-hop evidence retriever and claim verification components.

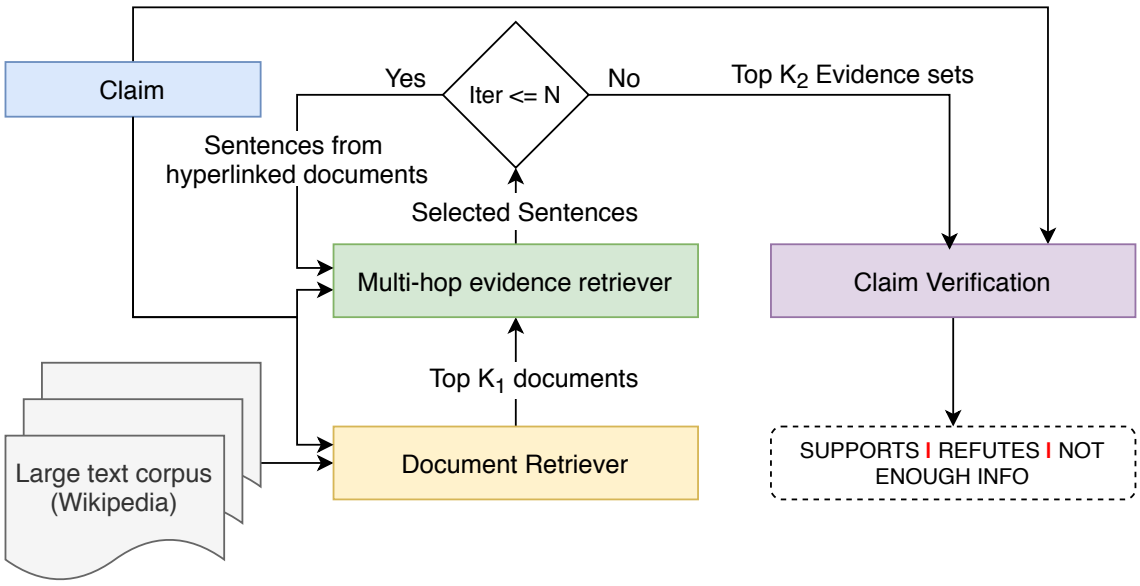

Figure 3.1: Our HESM framework.

3.2 Document Retriever

Document retrieval is the task of selecting documents related to a given claim. First, documents are selected by an exact match between their titles and a span of text of the claim. In particular, CoreNLP toolkit [86] is used for retrieving text spans from the claim. To obtain more relevant documents, the same procedure is applied again after eliminating articles such as 'a', 'an' or 'the' from the claim, and once again after singularizing each word in the claim. For documents, whose titles are ambiguous (e.g., *"Savages (band)"* and *Savages (2012 film)*), a semantic understanding strategy based on *Neural Semantic Matching Network* (NSMN) [14] is used to calculate the relevance of each of the documents by comparing the first line of each document with the claim. NSMN consists of four key components namely encoding, alignment, matching and output. The claim and the evidence are encoded using a recurrent neural network, after which the tokens in the claim and the evidence are aligned with each other using attention. Finally, the aligned representations are matched

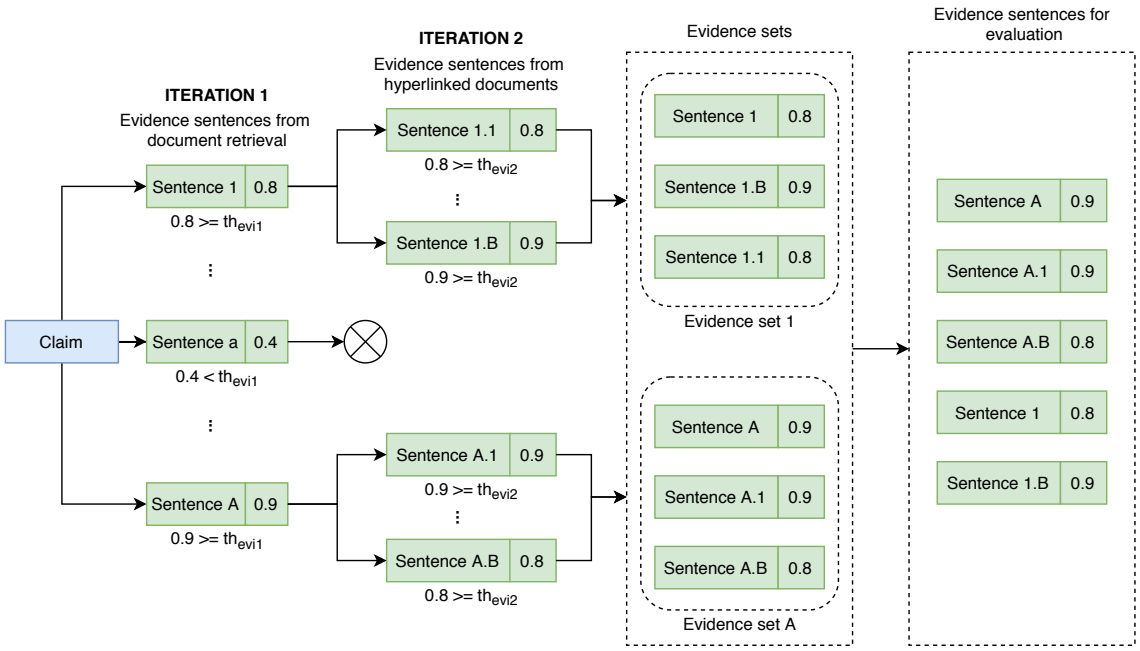

Figure 3.2: Multi Hop Evidence Retrieval.

using another recurrent neural network and the relevance score for document d is obtained using a linear classifier. Only the top K_1 ranked documents are selected.

3.3 Multi-hop Evidence Retriever

According to statistics of the FEVER dataset [23], 16.82% claims require multiple evidence sentences to verify their truthfulness, and 12.5% claims' evidence sentences are located across multiple documents. Based on this, we propose a multi-hop evidence retriever which is an iterative retrieval mechanism with N number of iterations or hops. From analysing the FEVER dataset, almost all the evidence sentences are at most two hops away from a claim, and thus can be retrieved in two iterations. Hence, for this work we set N as 2. We retrieve a maximum of K_2 evidence sets for each claim. Each evidence set contains a maximum of M_s evidence sentences. With the recent success of Transformer [52] based pre-trained models in NLP, we

incorporate the ALBERT model [68] as a part of our multi-hop evidence retriever. ALBERT is a lightweight BERT based model that is pre-trained on large-scale English language corpus for learning language representation.

In the first iteration, given a claim c_i, each sentence j in the selected documents from the document retriever is concatenated with the claim c_i as $[[CLS];c_i;[SEP];j]$ and passed through the ALBERT model. $[CLS]$ and $[SEP]$ are classification and separator tokens required by ALBERT model. From the ALBERT model representation of each input token, the representation of the $[CLS]$ token is pooled and fed to a linear layer classifier to produce the two scores m^+ and m^- for selecting and discarding the sentence, respectively. In Transformer based models, $[CLS]$ token is considered as representation of the whole input. Then, a selection probability $p(x = 1|c_i, j)$ is calculated as a softmax normalization between the two scores. Only the top K_2 sentences with highest m^+ scores and probability score greater than or equal to a threshold th_{evi1} are selected.

In the second iteration, each of the K_2 evidence sentences from the first iteration is considered as an evidence set. In FEVER dataset, for claims requiring multiple sentences for verification, most of the sentences missed in the first iteration of retrieval are found in hyperlinked documents of the sentences retrieved in first iteration. Therefore, in second iteration, the claim c_i, each of the K_2 evidence sentences j, and each sentence k from the hyperlinked documents in sentence j are concatenated as $[[CLS]; c_i; [SEP]; j; k]$ and fed as input to the ALBERT model. Similar to the first iteration, two scores m^+ and m^-, and a selection probability $p(x = 1|c_i, j, k)$ are obtained. Finally, for each evidence sentence j, a maximum of $(M_s - 1)$ sentences with highest m^+ scores and probability score greater than or equal to a threshold th_{evi2} are selected and added to the corresponding evidence set.

Different from previous works, we combine a sentence j retrieved in first itera-

tion, only with the sentences retrieved from hyperlinked documents in j, in second iteration. Thus, we form multiple evidence sets for a given claim, similar to Figure 1.3. Figure 3.2 summarizes the multi-hop retrieval. The number present along with each sentence is its selection probability $p(x = 1|c_i, j, k)$. For simplicity, we assume the m^+ score and $p(x = 1|c_i, j, k)$ are same. Let th_{evi1} be 0.5 and th_{evi2} be 0.8. For a claim, K_1 documents are retrieved by Document retrieval. Let A be the total number of sentences present in the K_1 documents. In iteration 1, sentences with probability greater than or equal to th_{evi1} are selected for iteration 2. Here, sentence a has a probability 0.4 which is lesser than th_{evi1} and thus not selected for iteration 2. Let us assume that only the sentences 1 and A are selected for second iteration. In iteration 2, the selection probability is obtained for all the sentences 1.1 to $1.B$ found in hyperlinked documents in sentence 1, where B is the total number of sentences found in the hyperlinked documents. Similarly, the selection probability is obtained for all the sentences in the hyperlinked documents of sentence A. Let us assume that for sentence 1, only the sentences $1.1, 1.B$ have probability greater than or equal to th_{evi2} and for sentence A, only $A.1, A.B$ have probability greater than or equal to th_{evi2}. Then, sentences $1, 1.B, 1.1$ form an evidence set and sentences $A, A.1, A.B$ form another evidence set. Notice the order in the evidence set with sentence 1. The sentences retrieved in second iteration are sorted according to their m^+ scores. Then, the two evidence sets are sorted according to the m^+ score of sentence retrieved in first iteration. Here, evidence set with sentence A has precedence over evidence set with sentence 1, since score for sentence A (0.9) is greater than score for sentence 1 (0.8). Assuming that, for evaluation of evidence retrieval, at most 5 sentences are considered, sentence 1.1 is omitted. Therefore, the final evidence sentences selected for evaluation are $A, A.1, A.B, 1, 1.B$.

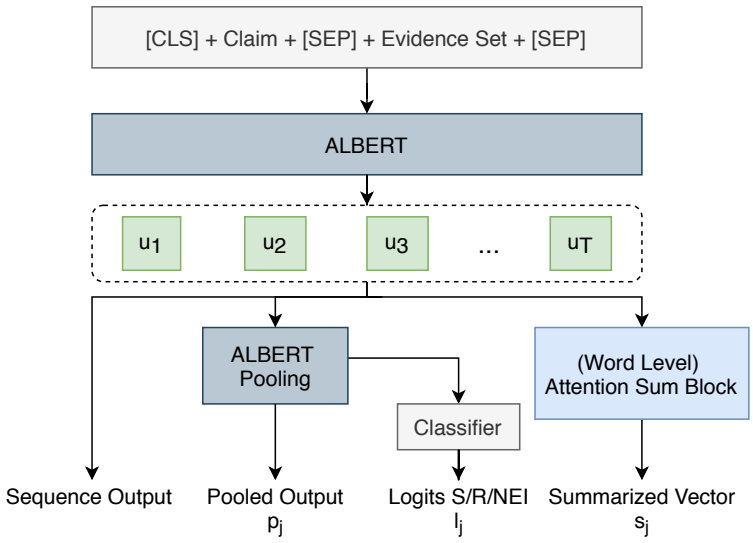

Figure 3.3: Evidence Set Modeling Block.

3.4 Claim Verification

Claim verification is a three-way classification task to label the claim as SUPPORTED, REFUTED or NOT ENOUGH INFO, based on the extracted evidences. Inspired by Hierarchical Attention Network [77], we propose a neural network that combines evidence sets hierarchically. While [77] uses word-level and sentence-level attention to hierarchically combine words into sentences and sentences into a document, in this task we use word-level and evidence set-level attention to hierarchically combine words and sentences into evidence sets, and evidence sets into an aggregated evidence. Different from [77], we propose two ways of aggregating evidence sets. Also, we train each evidence set to be able to verify the claim individually. The model consists of two parts: (1) Evidence Set Modeling Block that contains word-level encoder and attention layers to model each evidence set based on its words and sentences; and (2) Hierarchical Aggregator that contains evidence set-level encoder and attention layers to combine multiple evidence sets.

3.4.1 Evidence Set Modeling Block

The Evidence Set Modeling Block in Figure 3.3 takes a claim c_i and each evidence set e_j as input, and returns: (1) a sequence output $u_1, u_2, ..., u_T$, that is the representation of each token in the sequence; (2) a pooled output p_j, that can be considered as a joint representation of the claim and the evidence set (3) a summarized vector s_j, that is also a joint representation of the claim and the evidence set obtained using word level attention; and (4) the logits l_j from classification of the claim as SUPPORTS, REFUTES or NOT ENOUGH INFO, based on the evidence set e_j.

Word Encoder. We use the ALBERT model for word level encoder. Let J be the number of evidence sets retrieved for the claim c_i. First, all the sentences in an evidence set j are concatenated to form the evidence set sequence e_j, where $j \in [1, J]$. Then, the claim c_i and the evidence set sequence e_j are concatenated as $[[CLS]; c_i; [SEP]; e_j; [SEP]]$ to form the input sequence x_j. The word embeddings, $X_j \in \mathbb{R}^{T \times d}$, of the input sequence x_j is obtained from the ALBERT embedding layer, where T denotes the number of tokens in the input sequence x_j and d is the size of the word embedding. Then, the ALBERT model processes the input X_j and produces a sequence output $u_1, u_2, ..., u_T$ denoted by $U_j \in \mathbb{R}^{T \times d}$, which consists of the representation of each token t in x_j. The ALBERT model also consists of a pooling layer that returns the vector representation p_j of the $[CLS]$ token which is considered to be representation of the whole sequence in Transformer based models.

$$U_j = \text{ALBERT}(X_j) \in \mathbb{R}^{T \times d} \tag{3.1}$$

$$p_j = \text{ALBERT\_POOLER}(U_j) \in \mathbb{R}^{d} \tag{3.2}$$

Attention Sum Block. The Attention Sum block in Figure 3.4 returns a weighted sum of all the value token vectors v_1 to v_R, where the weights are calculated using attention between input token vectors q_1 to q_R and a trainable weight vector u_q that is randomly initialized. Each vector q_r is passed through a linear layer to get hidden representation f_r for each token $r \in [1, R]$. The hidden representation f_r is then subjected to a dot product with the vector u_q to form a scalar c_t which is the attention score for each q_r. Then, softmax is computed over all the attention scores c_1 to c_R to get an attention weight a_r for each token r. Finally, the value token vectors v_r are subjected to a weighted sum with attention probabilities from the softmax operation as weights and returns the summarized vector s. The attention weights denote the importance of each token in the value vectors sequence. The Attention Sum block is used in the following *Word Attention* and *Hierarchical Aggregation* components.

$$f_r = W_q q_r + b_q, r \in [1, R] \tag{3.3}$$

$$c_r = f_r^T u_q \tag{3.4}$$

$$a_r = softmax(c_r) \tag{3.5}$$

$$s = \sum_r v_r a_r \tag{3.6}$$

Word Attention. In the word-level attention component, the sequence output u_t, where $t \in [1, T]$, of the evidence set j obtained from *Word Encoder* is passed (as both the input q_r and value v_r vectors) through the *Attention Sum block* to obtain a summarized vector representation s_j (denoted as s in Attention Sum block), based on the importance of each word. s_j is used in the *Hierarchical Aggregation* component

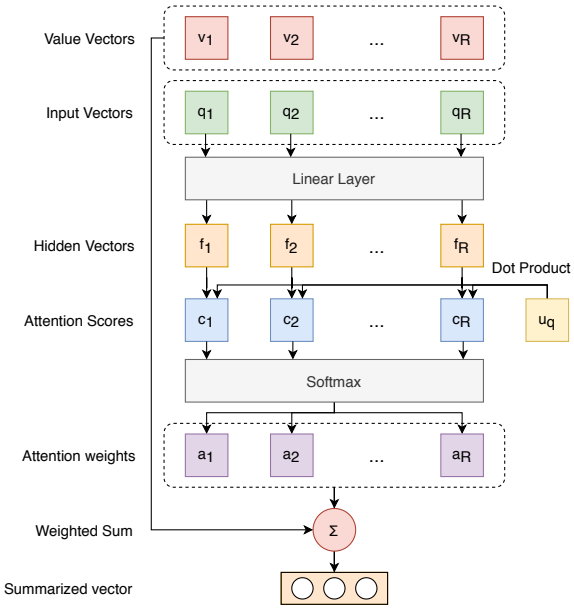

Figure 3.4: Attention Sum Block.

in Section 3.4.2.

$$s_j = \text{ATTN\_SUM}(u_1, u_2, ..., u_T) \in \mathbb{R}^d \tag{3.7}$$

Classifier. The pooled output vector p_j containing representation of $[CLS]$ token from the Word Encoder is passed through a linear layer to obtain a three way classification score l_j (SUPPORTS, REFUTES and NOT ENOUGH INFO classes) of the claim c_i based on the evidence set e_j. This classifier verifies the claim based on the evidence set.

$$l_j = W_w p_j + b_w \tag{3.8}$$

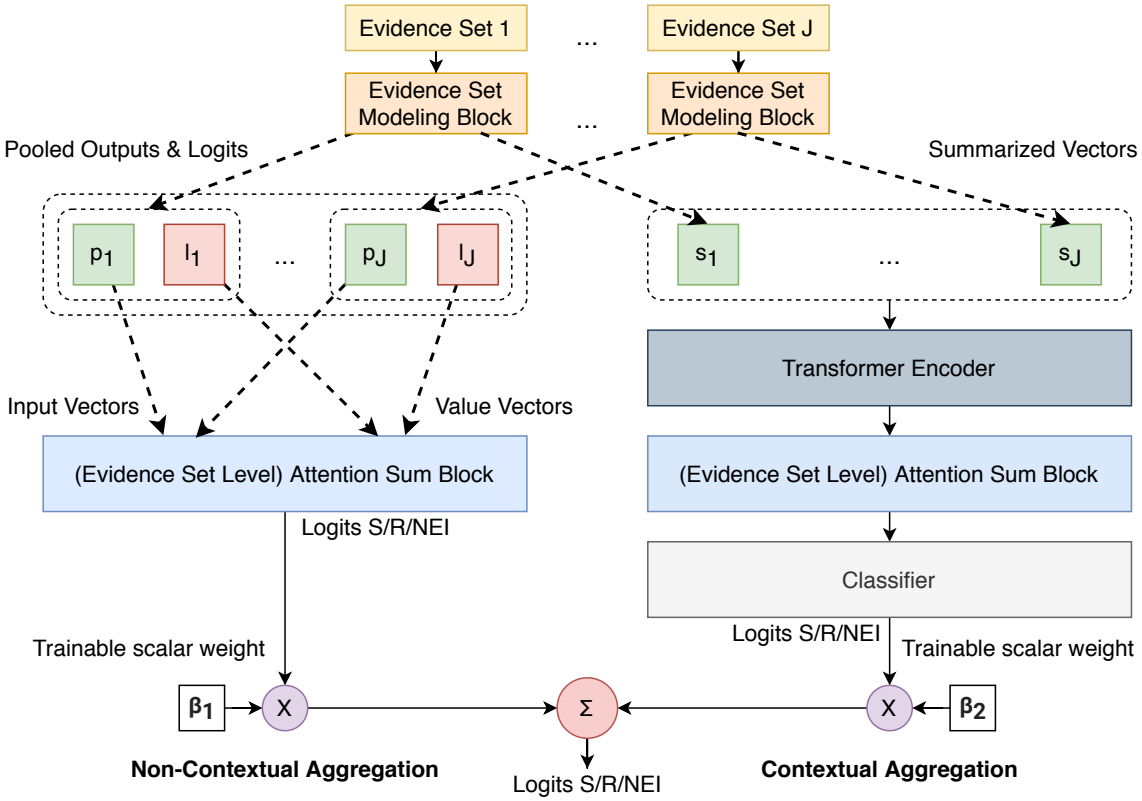

Figure 3.5: Hierarchical Aggregation.

3.4.2 Hierarchical Aggregation Modeling

The hierarchical aggregation component in Figure 3.5 takes the output of the Evidence Set Modeling block of all J evidence sets as input, and produces the three-way classification score for the claim based on all the evidence sets. It consists of two types of aggregations namely contextual and non-contextual aggregations. Both components compute an evidence set level attention to combine all the evidence sets, forming a hierarchy.

Non-contextual Evidence Set Aggregation. Non-contextual aggregation combines the logits $l_1, ..., l_J$ of all the evidence sets to produce the aggregated verification logits l_{nc}. Here, we do not contextually combine the evidence sets since the majority of claims only needs a single evidence sentence/evidence set for verification. The

pooled output $p_1, ..., p_J$ and the classification logits $l_1, ..., l_J$ of all the evidence sets, from the Evidence set modeling block, are passed through the Attention Sum block to compute the aggregated representation of all the evidence sets. Here, the sequence of vectors $p_1, p_2, ...p_J$ forms the input vectors of Attention Sum block and the logits $l_1, l_2, ..., l_J$ forms the value vectors of the Attention Sum block. Thus, it aggregates the logits of all evidence sets based on the importance of each evidence set.

$$l_{nc} = \text{ATTN\_SUM}(p_1, ..., p_J; l_1, ..., l_J) \tag{3.9}$$

Contextual Evidence Set Aggregation. Contextual aggregation combines the representation s_j of each evidence set j with one another to produce the claim verification logits l_c. Even though we combine evidence sentences into evidence sets through the multi-hop retriever, our extracted evidence sets might not be completely accurate for some claims (i.e., some evidence sentences that belong to the same ground truth evidence set might be distributed across our extracted evidence sets). Therefore, we combine the evidence sets contextually to overcome the limitation. Let $S \in \mathbb{R}^{J \times d}$ denotes the summarized representations $s_1, s_2, ..., s_J$ of all the evidence sets $[1, J]$. S is passed through a Transformer encoder, in order to obtain contextual representations $m_1, m_2, ..., m_J$ denoted by $M \in \mathbb{R}^{J \times d}$. Here, the Transformer encoder layer ensures that the context from one evidence set is combined with other evidence sets. Then, the evidence set representations m_j, where $j \in [1, J]$, from the encoder are passed (as both the input q_r and value v_r vectors) through the Attention Sum block to obtain an aggregated vector representation k of all the evidence sets. Finally, the vector representation k is fed into a linear layer

classifier to obtain the three way classification logits l_c of the claim.

$$M = \text{Transformer\_Encoder}(S) \in \mathbb{R}^{J \times d} \tag{3.10}$$

$$k = \text{ATTN\_SUM}(m_1, m_2, ..., m_J) \tag{3.11}$$

$$l_c = W_s k + b_s \tag{3.12}$$

Aggregated Logits. The aggregated logits are computed based on a weighted combination of the scores from contextual and non-contextual aggregations. The weights β_1 and β_2 are trainable weights that denote importance of each aggregation.

$$l = \beta_1 l_c + \beta_2 l_{nc} \tag{3.13}$$

3.4.3 Training Loss and Inference

The three-way classification logits l_j from the Evidence set Modeling block for each evidence set j are subjected to a cross entropy loss. All the losses from each evidence set j are averaged to get an aggregated loss L_{esm}. The aggregated classification logits l from the Hierarchical Aggregation Modeling block are subjected to a cross entropy loss L_{ham}. The final loss is the sum of L_{esm} and L_{ham}.

During the inference, the aggregated logits l from the Hierarchical Aggregation Modeling is used as the final three-way classification score of the claim verification. The label with the maximum score is selected as the final classification label.

Chapter 4

Experimentation and Results

In this chapter, we describe the dataset used, evaluation metrics, baselines, and implementation details, results in our experiments and further analysis.

4.1 Experiment Setting

4.1.1 Dataset

We evaluate our framework HESM in the FEVER dataset, a large scale fact verification dataset [23]. The dataset consists of 185,445 claims with human-annotated evidence sentences from 5,416,537 documents. Each claim is labeled as SUPPORTS, REFUTES or NOT ENOUGH INFO. The dataset consists of training, development and test sets as shown in Table 4.1. The training and development sets along with their ground truth evidences and labels are available publicly. But, the ground truth evidences and labels of the test set are not publicly available. Instead, once extracted evidence sets/sentences and predicted labels of the test set by a model are submitted to the online evaluation system[1], its performance is measured and

[1]https://competitions.codalab.org/competitions/18814

Split	SUPPORTED	REFUTED	NOT ENOUGH INFO
Train.	80,035	29,775	35,639
Dev.	6,666	6,666	6,666
Test	6,666	6,666	6,666

Table 4.1: Statistics of FEVER Dataset.

displayed at the system. In this work, we train and tune our hyper-parameters on training and development sets, respectively.

4.1.2 Baselines

We compare our model with 7 state-of-the-art baselines including the top performed models from FEVER Shared task 1.0 [47], BERT based models, and a graph based model. Although we compare ours against all of them, the BERT based models are our major baselines since we use ALBERT which is a light weight BERT based model. The detailed description of the baselines is presented below.

The top performed models from FEVER shared task 1.0 include UNC NLP [14], UKP Athene [16] and UCL MRG [15]. All three models use a modified version of Enhanced Sequential Inference Model [32] for claim verification. UNC NLP model concatenates all retrieved evidence sentences together to verify the claim whereas UCL MRG and UKP Athene models process each evidence sentence separately and aggregate them at a later stage. UCL MRG reports the best results with linear layer aggregation. UKP Athene uses an attention based aggregation.

The BERT based models include [48, 49, 53]. [48] uses BERT-base and BERT-large for evidence retrieval and claim verification, respectively. They also experiment with both pairwise and point-wise ranking for evidence retrieval. [49] uses two iterations of evidence retrieval similar to our work, but different from our work they concatenate all the sentences retrieved. [53] reports performance for both BERT-concat that concatenates all the sentences, and BERT-pair model that processes

each claim-evidence sentence pair separately. GEAR [53] uses BERT Base as backbone and aggregates claim-evidence sentence pair using fully-connected graph based evidence reasoning network. A graph based model KGAT [50] uses a modified version of Graph Attention Network [87] to model a graph constructed from claim and evidences. KGAT experiments with both BERT Base and BERT Large models as its backbone.

4.1.3 Evaluation Metrics

The official evaluation metrics of the FEVER dataset are Label Accuracy (LA) and FEVER score. Label Accuracy is the three-way classification accuracy for the labels SUPPORTS, REFUTES and NOT ENOUGH INFO, regardless of the retrieved evidence. FEVER score considers a claim to be correctly classified only if the retrieved evidence set matches at least one of the ground truth evidence sets along with the correct label. Between the two metrics, FEVER score is considered as the most important evaluation metric because it considers both correct evidence retrieval and correct label prediction.

For evidence retrieval performance evaluation, recall and OFEVER are reported since these two scores matter for the claim verification process. Note that OFEVER is the oracle fever score calculated assuming that the claim verification component has 100% accuracy. As formulated by [23], a maximum of 5 evidence sentences are extracted to calculate evidence retrieval performance. For our model's evaluation purpose, we assign the score of evidence sentences retrieved in first iteration to their corresponding evidence sets. Then, we sort the evidence sets based on their assigned scores and select at most 5 sentences from the evidence sets in the same sorted order.

4.1.4 Implementation and Training details

In the Document retrieval stage we follow the retrieval mechanism used by [14] which involves keyword matching and training a neural network to resolve the documents containing ambiguous title. For training the neural network, Adam optimizer [88] is used with a batch size of 128. Cross Entropy Loss is used to train the network. The maximum number of documents retrieved, given by K_1 is set to 10.

In the Multi-hop evidence retrieval stage we mainly use ALBERT model. The number of iterations N is set to 2. For both iterations, ALBERT-Base model for sequence classification is used and is trained using a batch size of 64 along with AdamW optimizer [89] and a learning rate of 5e-5. In the first iteration, we set the threshold probability th_{evi1} for selection to 0.5 and maximum number of sentences per claim K_2 to 3. We also use the annealed sampling strategy followed by [14] to decrease the number of negative examples after each epoch so that model learns to be more tolerant about selecting sentences while being discriminative enough to filter out apparent negative sentences.

In the second iteration we use the ALBERT-Base model to retrieve relevant sentences in hyperlinked documents of evidence sentences retrieved in first iteration. Similar to first iteration, we use annealed sampling here as well. We set the maximum number of sentences in an Evidence set, M_s to be 3. Finally, we choose either K_2 evidence sets or lesser if lesser number of evidence sets leads up to 5 or more evidence sentences, since only 5 evidence sentences are considered for calculating FEVER score. We set the threshold probability th_{evi2} to 0.8 since we find that the model is able to retrieve correct evidence sentences with a high probability and lowering this probability to 0.5 adds a lot of irrelevant evidence sentences thus reducing the precision score. Both the iterations are trained for 4 epochs. Cross entropy loss is used in both iterations.

Finally, in claim verification stage, we use hierarchical evidence set modeling which uses ALBERT model as its backbone. We use AdamW optimizer with a batch size of 32 and a learning rate of 2e-5 in our final model. It also uses a 2 layer transformer encoder for evidence-set level encoding. We set the maximum token count for the model to be 300. The claim verification is trained for 4 epochs.

Hyper-parameter Tuning. We use PyTorch framework to optimize both Multi-hop evidence retriever and claim verification components. We use grid-search on development set to search over a batch size from {32, 64}, a learning rate from {2e-5, 5e-5}, and number of epochs from {2, 4, 6}. The maximum number of evidence sets K_2 is selected from {2, 3, 4} and maximum number of sentences per evidence set M_s is selected from {2, 3, 4}. In claim verification, the number of transformer encoder layers in contextual aggregation is selected from {1, 2, 3}.

4.2 Experimental Results and Analysis

Experiments are conducted to evaluate performance of evidence retrieval, claim verification and aggregation approaches. In addition, we conduct ablation study. Only claim verification experiment is conducted in the test set since each baseline's officially evaluated results are reported in the FEVER leaderboard. In other experiments and analysis, we use the development set since the test set does not contain the ground truth of evidence sets/sentences and claim classification labels, The leaderboard does not provide all the necessary evaluation results for these experiments on test set as well.

Model	# of Iterations	Recall	OFEVER (%)
UNC NLP [14]	1	0.868	91.19
BERT-Base [49]	2	0.898	93.20
our HESM (ALBERT-Base)	**2**	**0.905**	**93.70**

Table 4.2: Evidence retrieval performance of the baselines and our model in development set.

Model	LA(%)	FEVER(%)
UKP Athene [46]	65.46	61.58
UCL MRG [15]	67.62	62.52
UNC NLP [14]	68.21	64.21
BERT Pair [53]	69.75	65.18
BERT Concat [53]	71.01	65.64
BERT (Base) [48]	70.67	68.50
GEAR (BERT Base) [53]	71.60	67.10
KGAT (BERT Base) [50]	72.81	69.40
our HESM (ALBERT Base)	**73.25**	**70.06**
BERT (Large) [48]	71.86	69.66
BERT (Large) [49]	72.71	69.99
KGAT (BERT Large) [50]	73.61	70.24
KGAT (RoBERTa Large) [50]	74.07	70.38
our HESM (ALBERT Large)	**74.64**	**71.48**

Table 4.3: Performance of the baselines and our model in test set.

4.2.1 Multi-hop evidence retrieval

As shown in Table 4.2, we compare the performance of our model with two baselines, UNC NLP [14] and BERT based model [49]. UNC NLP uses ESIM [32] based model, and [49] uses a BERT based model. Since most other previous works either use ESIM based model or BERT based model for evidence retrieval, we compare with these two representative baselines (i.e., the results of the other 5 baselines in evidence retrieval would be similar to one of them). Our HESM with ALBERT Base outperforms the baselines, achieving 0.905 recall and 93.70% OFEVER score. We can also notice that multiple-hop evidence retrieval approaches (ours and [49]) performed better than UNC NLP, which conducts a single iteration.

Aggregation	LA(%)	FEVER(%)
Logical	68.92	66.32
Top-1	69.92	67.77
MLP	74.25	72.03
Concat	74.87	72.13
Attention-based	74.96	72.74
HESM	**75.77**	**73.44**

Table 4.4: Claim verification with different aggregation methods in development set.

4.2.2 Claim verification

Table 4.3 shows claim verification results of our HESM model and baselines. Our model with ALBERT Large outperforms all the baselines, achieving 74.64% label accuracy (LA) and 71.48% FEVER score. In particular, our model performed much better than the top performed models from FEVER Shared task 1.0 (i.e., UKP Athene, UCL MRG and UNC NLP). Compared with baselines using BERT Base, our HESM with ALBERT Base performed better than them. Likewise, compared with baselines using large language models, our model with ALBERT large still performed better than them. This experimental result confirms that our model with ALBERT large improved 1.1% FEVER score compared with the best baseline, KGAT with RoBERTa Large, indicating our model's capability of producing more correct label prediction and evidence extraction.

4.2.3 Aggregation Analysis

We compare our hierarchical aggregation with different baseline aggregation methods. Table 4.4 shows the results of aggregation analysis in the development set. Top-1 aggregation is using just the top-1 relevant evidence set to verify the claim. Logical aggregation involves classifying the claim as SUPPORTS or REFUTES if at least one of the evidence sets has the label SUPPORTS or REFUTES respectively.

Model	LA(%)	FEVER(%)
HESM	75.77	73.44
- w/o Evidence set level Loss	75.35	72.74
- w/o Non-Contextual Aggregation	75.33	72.74
- w/o Contextual Aggregation	73.70	71.96

Table 4.5: Ablation analysis in development set.

In case both labels appear in the evidence sets, then the label from top scoring evidence set is used to break the tie. If both labels do not appear in any of the evidence sets, we predict the claim as NOT ENOUGH INFO. MLP aggregation is to use a MLP layer to aggregate the class label probability of all the evidence sets to get a final verification label. Concat aggregation concatenates all the sentences in all evidence sets into a string to verify the claim. Attention-based aggregation is similar to the aggregation technique used in [46] using attention between claim and each evidence set to get the importance of each evidence set and then combine them using Max and Mean pooling. Finally, our HESM model aggregates evidence sets using hierarchical aggregation. From the results, we can see that our HESM model outperforms all other aggregation methods.

4.2.4 Ablation Study

Table 4.5 shows the label accuracy and FEVER score of our model after removing different components including evidence set level loss L_{esm}, and contextual and non-contextual aggregations. All of the proposed components positively contributed to boost performance of our framework.

4.2.5 Contextual and Non-contextual Aggregations

In this section, we study the performance of contextual and non-contextual aggregations in different aspects in the development set.

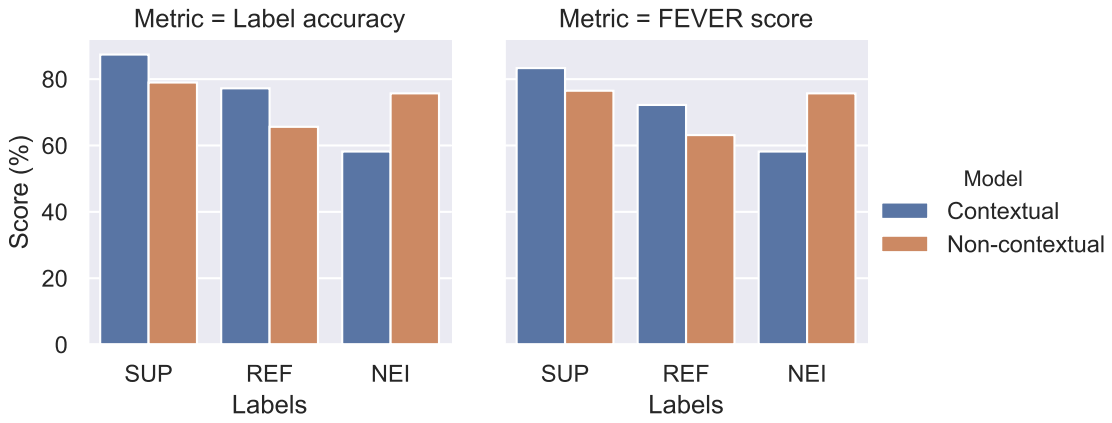

Figure 4.1: Performance of contextual and non-contextual aggregations given different claim labels.

Label-wise performance. Figure 4.1 shows performance of contextual and non-contextual aggregations with respect to the class labels. We use the logits l_c and l_{nc} to calculate performance of contextual and non-contextual aggregations. In both label accuracy and FEVER score, contextual aggregation performs better for correctly verifying a claim when the relevant evidence either supports or refutes the claim, whereas non-contextual aggregation performs better in identifying correct evidences that do not have enough information to support or refute the claim (i.e., claims with the label NOT ENOUGH INFO). Thus, each aggregation complements the other in claim verification. The reason behind the division of labour is the outcome of the strategic architectural choices made for contextual and non-contextual aggregation, combined with the training strategy. Since the evidence set modeling block is trained to verify the claim based on each evidence set (retrieved from the multi-hop evidence set retrieval component) individually, the classification output from evidence set modeling block is more accurate for evidence sets that cannot completely verify the claim (i.e.) NOT ENOUGH INFO evidence sets. The reason is that, for each claim, the majority of the evidence sets retrieved from the multi-hop retrieval component will not have enough information to verify the claim and thus the ev-

idence set modeling block can identify these evidence sets more accurately. Since non-contextual aggregation aggregates the classification labels using word-level attention, rather than aggregating the contextual information from all the evidence sets, it is still capable of more accurately identifying the NOT ENOUGH INFO evidence sets (and in general NOT ENOUGH INFO claims). On the other hand, since contextual aggregation aggregates the contextual information from all the evidence sets, it is able to learn from the context of all sentences (including the sentences that belong to the same ground truth evidence set, that are mistakenly grouped into different evidence sets by multi-hop evidence retrieval component), leading to more accurate classification of claims with SUPPORTS/REFUTES label.

Performance on claims requiring different number of evidences. Figure 4.2 shows performance of contextual and non-contextual aggregations with respect to claims requiring different number of evidence sentences for verification. *Overall* refers to all the claims, *Single* refers to claims requiring only single evidence sentence for verification, *Any* refers to claims that can be verified with one or more evidence sentences and *Multi* refers to claims that can be verified only with multiple sentences. Non-contextual aggregation performs better than contextual aggregation in claims requiring only *Single* evidence sentence, whereas contextual aggregation performs better than non-contextual aggregation in claims requiring *Any* and *Multi* evidence sentences. The results make sense because contextual aggregation combines the context of multiple evidence sets while non-contextual aggregation usually selects one of the evidence sets based on the attention mechanism.

Attention analysis. In Table 4.6 we show the weights β_1 and β_2 of the final model and also the evidence-set level attention accuracy. The weights can be seen as the importance of each aggregation. The evidence set-level attention accuracy

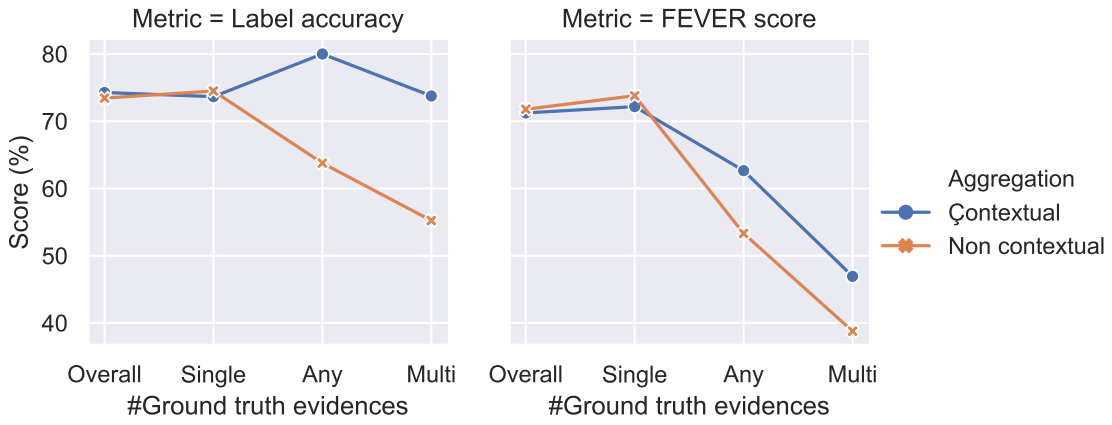

Figure 4.2: Performance of contextual and non-contextual aggregations given claims requiring different number of evidence sentences.

Aggregation	Weights	Attention acc.
Contextual	0.48	84.88
Non-contextual	0.52	85.98

Table 4.6: Contextual and Non-contextual aggregation attention metrics

evaluates whether the evidence sets that match one of the ground truth evidences has highest attention of all the evidence sets. The evidence set-level block in both non-contextual and contextual aggregation calculates the weight of each evidence set using attention-sum block. The evidence set-level attention accuracy for each aggregation is calculated by taking the evidence set with maximum attention and checking if this evidence set matches with one of the ground truth evidence sets. The overall attention accuracy for contextual and non-contextual are 84.88% and 85.98% respectively, which shows that our evidence-set level attention is capable of weighting the correct evidence sets more.

Finally, in Figure 4.3, we show the the concentration of maximum attention on evidence sets for all 19998 instances in dev set sorted according to the attention value. We can see that the attention is mostly focused on single evidence for both contextual and non-contextual aggregation. For non-contextual aggregation the at-

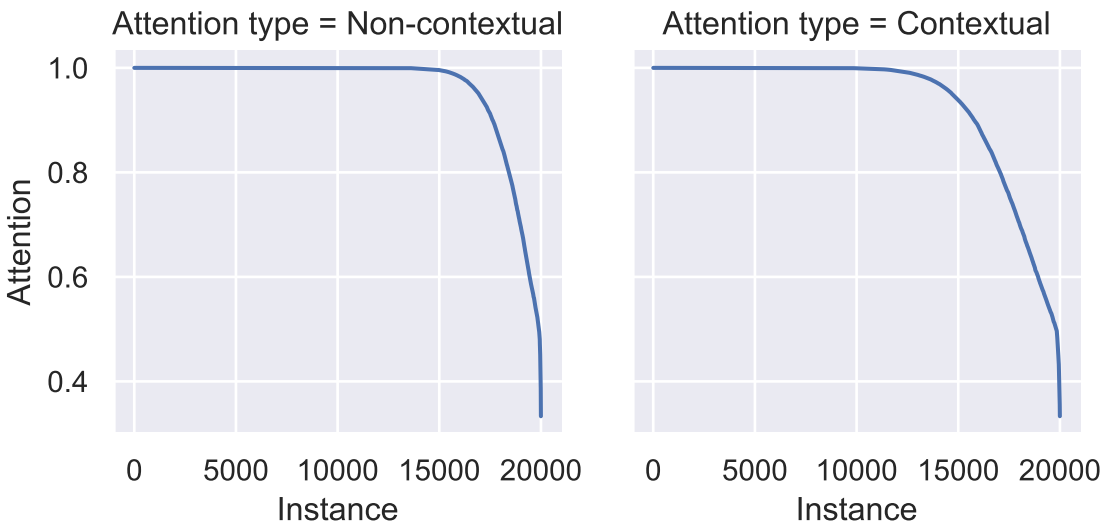

Figure 4.3: Evidence-set level attention in Contexual and Non-contextual aggregation

tention becomes less focused around 16000^{th} instance and for contextual aggregation the attention becomes less concentrated around 16000^{th} instance which is agreement with the better performance of contextual aggregation on claims requiring textit-Multi evidence sentences for verification seen in Figure 4.2.

4.2.6 Impact of noisy evidence sentences on Concat and HESM models.

Figure 4.4 shows the performance of Concat model with ALBERT-Base backbone which concatenates all the evidence sentences and HESM with contextual aggregation which contextually aggregates the evidence sets. The number of noisy evidences denote the number of non-ground truth evidence sentences selected from multi-hop retriever. We consider only the claims for which the retrieved evidence sentences match at least one ground truth evidence set. Here, we choose the result from contextual aggregation l_c, since in section 4.2.5 we show that contextual aggregation is

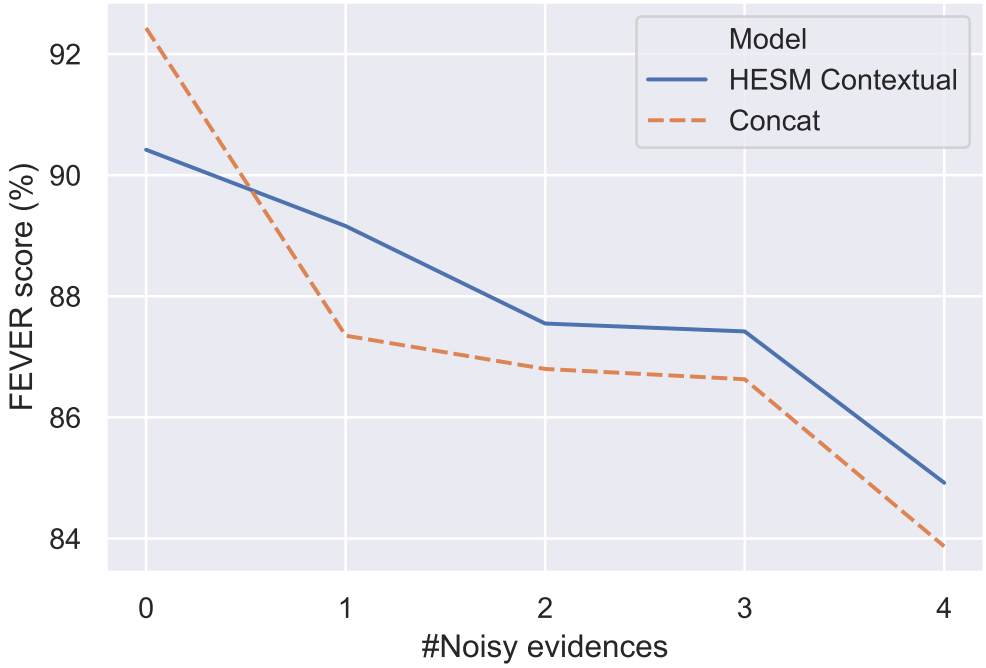

Figure 4.4: Performance comparison of Concat and HESM with different noise evidence sentences retrieved

responsible for higher scores in claims with SUPPORTS and REFUTES label. Here, we can see that concatenation works best when there are no noisy evidence sentences and HESM with contextual aggregation performs better than Concat model when number of noisy evidence sentences increase. This supports our initial claim that concatenation of sentences suffer from learning the context of only the relevant sentences when noisy or distracting sentences are present. Our HESM model's performance shows that it can predict better results when noisy evidence sentences are present which mostly is the case in any open domain information extraction task.

Chapter 5

Future Work

5.1 Joint Learning

Since we are training multi-hop retriever and claim verification separately, the claim verification component relies heavily on the performance of multi-hop retriever. When the multi-hop retriever fails to retrieve the correct evidences, recovering from the failure is not possible in the claim verification component. Therefore, if we jointly train the two components the errors from the claim verification will be propagated to the multi-hop retriever providing a way to recover from its failure. This has been observed by research works in open domain question answering [90, 91] and this applies to fact verification task as well. Also, in this work we end up with picking several hyper-parameters such as K_2, th_{evi1} and th_{evi2}. These hyper-paramaters are crucial in inference and tuning them is a complex task. To overcome this limitation and to solve the discrete selection of evidences in each component, several models in question answering task train the whole architecture using Reinforcement learning [92], [93], [94]. Similar training procedure can be adopted for fact verification.

5.2 Alternative Approaches

Forming evidence sets without hyperlink information. In this work, evidence sets are constructed based on selecting sentences from hyperlinks, but an ideal system should be capable of combining evidence sentences into evidence sets without hyperlink information. Combining evidence sentences into evidence sets without hyperlink information could benefit systems not only in fact verification, but several other tasks such as open-domain question answering and fake news detection.

Incorporating Constituency and Dependency trees Many textual encoders including Transformer based models, process text as a sequential input. Text can also be seen as a hierarchical input when it is represented using constituency and dependency trees. Incorporating the tree structure has several advantages including capturing the compositional effects of language and phrase-level encoding and attention. It also imposes strong inductive bias which can be helpful when training data is minimal. Several works exist to incorporate the tree structure [95, 96, 97], which we can use for fact-verification.

Adversarial attention for learning partial associations. From analysis we find that some words in evidence sentences are partially associated with more than one label. For example, for the claim *"Statue of Liberty is in New York"* and the evidence sentence *"Statue of Liberty is in USA"*, the word USA is partially associated with SUPPORTS since New York is in USA, and partially associated with NOT ENOUGH INFO since the evidence does not mention where exactly Statue of Liberty is in USA. Exploiting these partial associations can be an interesting direction of research. [98] solves a similar problem in multi-dimensional emotion regression by using adversarial attention to learn one encoding per label for each

word in a sentence.

Incorporating world knowledge and numerical processing Some claims in the FEVER dataset require additional world knowledge and numerical processing to verify a claim. For example, for the *claim "Statue of Liberty is in USA"* and the evidence sentence *"Statue of Liberty is in New York"*, the model has to have world knowledge to know that New York is part of USA. Without this information, the model might refute the claim. Also, for the claim *"Syracuse metropolitan area has more than 600,000 residents"* and evidence sentence *"Syracuse metropolitan area has a population of 662,577"*, the model has to understand that 662,577 is more than 600,000. This could be a potential research direction to exploit.

Better handling of *Not Enough Info* claims From analysis we find that, NOT ENOUGH INFO class still seems to be the problematic label to classify because of its minor and subtle differences from claims with SUPPORTS/REFUTES label. One straight-forward approach is to collect to more data for SUPPORTS/REFUTES label. Other modeling approaches [99, 100, 101, 102] used in *NO ANSWER* classification for question answering task in SQUAD 2.0 dataset can be utilized in fact verification.

5.3 Fake News Detection

Fake news detection is a complex task requiring immediate attention since fake news is spreading at an alarming rate. Several datasets exist for fake news detection, but most of them are several orders of magnitude smaller than FEVER dataset. Therefore, it would be interesting to check whether the learned knowledge is directly transferrable to fake news domain. A recently released fake news detection dataset

that is large scale and following a similar task formulation is *Richly Annotated FC* [38]. It would also be interesting to jointly train and evaluate our HESM model on both Richly Annotated FC dataset and the FEVER dataset.

5.4 Robustness

Robustness of natural language understanding architectures is a major concern [17, 103]. To address this issue, FEVER Shared Task 2.0 [103] has hosted a *Build-It, Break-It, Fix-it* style challenge for fact-verification on FEVER dataset. To this extent, we could evaluate our model on adversarial examples provided in [103] and improve our model based on the analysis. [104] points out the lexical bias towards refuted claims in FEVER dataset. To solve this, we can consider a meaning-preserving data-augmentation technique called back-translation [105], to obtain different wordings of the same claim which might help in alleviating the bias.

Chapter 6

Conclusion

In this thesis, we have proposed HESM framework for automated fact extraction and verification. HESM operates at evidence set level initially and combines information from all the evidence sets using hierarchical aggregation to verify the claim. Our experiments confirm that our hierarchical evidence set modeling outperforms 7 state-of-the-art baselines, producing more accurate claim verification. Our aggregation and ablation study show that our hierarchical aggregation works better than many baseline aggregation methods. Our analysis of contextual and non-contextual aggregations illustrates that the aggregations perform different roles and positively contribute to different aspects of fact-verification.

Appendix A

Resources

A.1 Links to resources

In Table A.1, we provide links to important resources related to our work.

Resource	**URL**
FEVER Training set	`https://s3-eu-west-1.amazonaws.com/fever.public/train.jsonl`
FEVER Dev set	`https://s3-eu-west-1.amazonaws.com/fever.public/shared_task_dev.jsonl`
FEVER Test set	`https://s3-eu-west-1.amazonaws.com/fever.public/shared_task_test.jsonl`
FEVER Website	`https://fever.ai/resources.html`
Test set Leaderboard	`https://competitions.codalab.org/competitions/18814`

Table A.1: Links to resources

Bibliography

[1] Maksym Gabielkov, Arthi Ramachandran, Augustin Chaintreau, and Arnaud Legout. Social Clicks: What and Who Gets Read on Twitter? In *ACM SIGMETRICS / IFIP Performance 2016*, 2016.

[2] Soroush Vosoughi, Deb Roy, and Sinan Aral. The spread of true and false news online. *Science*, 2018.

[3] Nic Newman, Richard Fletcher, Antonis Kalogeropoulos, and Rasmus Nielsen. *Reuters institute digital news report 2019*. Reuters Institute for the Study of Journalism, 2019.

[4] David MJ Lazer, Matthew A Baum, Yochai Benkler, Adam J Berinsky, Kelly M Greenhill, Filippo Menczer, Miriam J Metzger, Brendan Nyhan, Gordon Pennycook, David Rothschild, et al. The science of fake news. *Science*, 2018.

[5] Tsvetomila Mihaylova, Preslav Nakov, Lluis Marquez, Alberto Barron-Cedeno, Mitra Mohtarami, Georgi Karadzhov, and James Glass. Fact checking in community forums. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[6] Anselmo Peñas, Álvaro Rodrigo, Valentín Sama, and Felisa Verdejo. Overview of the answer validation exercise 2006. In *Evaluation of Multilingual and Multimodal Information Retrieval, 7th Workshop of the Cross-Language Evaluation Forum, CLEF 2006, Alicante, Spain, September 20-22, 2006, Revised Selected Papers*, 2006.

[7] Cyril Labbé, Natalie Grima, Thierry Gautier, Bertrand Favier, and Jennifer A Byrne. Semi-automated fact-checking of nucleotide sequence reagents in biomedical research publications: The seek & blastn tool. *PloS one*, 2019.

[8] Victoria L Rubin, Yimin Chen, and Niall J Conroy. Deception detection for news: three types of fakes. *Proceedings of the Association for Information Science and Technology*, 2015.

[9] Hunt Allcott and Matthew Gentzkow. Social media and fake news in the 2016 election. *Journal of Economic Perspectives*, 2017.

[10] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. Fake news detection on social media: A data mining perspective. *SIGKDD Explor. Newsl.*, 2017.

[11] William Yang Wang. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2017.

[12] Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. Automatic detection of fake news. In *Proceedings of the 27th International Conference on Computational Linguistics*, 2018.

[13] Isabelle Augenstein, Christina Lioma, Dongsheng Wang, Lucas Chaves Lima, Casper Hansen, Christian Hansen, and Jakob Grue Simonsen. MultiFC: A real-world multi-domain dataset for evidence-based fact checking of claims. 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)*. Association for Computational Linguistics, 2019.

[14] Yixin Nie, Haonan Chen, and Mohit Bansal. Combining fact extraction and verification with neural semantic matching networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

[15] Takuma Yoneda, Jeff Mitchell, Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. UCL machine reading group: Four factor framework for fact finding (HexaF). In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, 2018.

[16] Andreas Hanselowski, Avinesh PVS, Benjamin Schiller, Felix Caspelherr, Debanjan Chaudhuri, Christian M. Meyer, and Iryna Gurevych. A retrospective analysis of the fake news challenge stance-detection task. In *Proceedings of the 27th International Conference on Computational Linguistics*, 2018.

[17] Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017.

[18] James Thorne and Andreas Vlachos. An extensible framework for verification of numerical claims. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, 2017.

[19] Andreas Vlachos and Sebastian Riedel. Identification and verification of simple claims about statistical properties. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015.

[20] Ndapandula Nakashole and Tom M. Mitchell. Language-aware truth assessment of fact candidates. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2014.

[21] Hannah Bast, Björn Buchhold, and Elmar Haussmann. Overview of the triple scoring task at the wsdm cup 2017. *ArXiv*, abs/1712.08081, 2017.

[22] Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. Tabfact: A large-scale dataset for table-based fact verification. In *International Conference on Learning Representations*, 2020.

[23] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit 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.

[24] James Thorne and Andreas Vlachos. Automated fact checking: Task formulations, methods and future directions. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, 2018.

[25] Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. In *Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment*, 2005.

[26] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2015.

[27] Leon Derczynski, Kalina Bontcheva, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Arkaitz Zubiaga. SemEval-2017 task 8: RumourEval: Determining rumour veracity and support for rumours. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Association for Computational Linguistics, 2017.

[28] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2016.

[29] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, July 2018.

[30] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015.

[31] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. 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.

[32] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana 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.

[33] Chuanqi Tan, Furu Wei, Wenhui Wang, Weifeng Lv, and Ming Zhou. Multiway attention networks for modeling sentence pairs. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, 2018.

[34] Seonhoon Kim, Inho Kang, and Nojun Kwak. Semantic sentence matching with densely-connected recurrent and co-attentive information. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

[35] Kazi Saidul Hasan and Vincent Ng. Stance classification of ideological debates: Data, models, features, and constraints. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*. Asian Federation of Natural Language Processing, 2013.

[36] William Ferreira and Andreas Vlachos. Emergent: a novel data-set for stance classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016.

[37] Sihao Chen, Daniel Khashabi, Chris Callison-Burch, and Dan Roth. PerspectroScope: A window to the world of diverse perspectives. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2019.

[38] Andreas Hanselowski, Christian Stab, Claudia Schulz, Zile Li, and Iryna Gurevych. A richly annotated corpus for different tasks in automated fact-checking. *arXiv preprint arXiv:1911.01214*, 2019.

[39] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. Fake news detection on social media. *ACM SIGKDD Explorations Newsletter*, 2017.

[40] Preslav Nakov, Alberto Barrón-Cedeño, Tamer Elsayed, Reem Suwaileh, Lluís Màrquez, Wajdi Zaghouani, Pepa Atanasova, Spas Kyuchukov, and Giovanni Da San Martino. Overview of the clef-2018 checkthat! lab on automatic identification and verification of political claims. *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, 2018.

[41] Kashyap Popat, Subhabrata Mukherjee, Andrew Yates, and Gerhard Weikum. DeClarE: Debunking fake news and false claims using evidence-aware deep learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2018.

[42] Danqi Chen, Adam Fisch, Jason Weston, and Antoine 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)*, 2017.

[43] Matthew Dunn, Levent Sagun, Mike Higgins, V Ugur Guney, Volkan Cirik, and Kyunghyun Cho. Searchqa: A new q&a dataset augmented with context from a search engine. *arXiv preprint arXiv:1704.05179*, 2017.

[44] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. Ms marco: A human-generated machine reading comprehension dataset. *30th Conference on Neural Information Processing Systems*, 2016.

[45] Bhuwan Dhingra, Kathryn Mazaitis, and William W Cohen. Quasar: Datasets for question answering by search and reading. *arXiv preprint arXiv:1707.03904*, 2017.

[46] Andreas Hanselowski, Hao Zhang, Zile Li, Daniil Sorokin, Benjamin Schiller, Claudia Schulz, and Iryna Gurevych. UKP-athene: Multi-sentence textual entailment for claim verification. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, 2018.

[47] James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. The fact extraction and VERification (FEVER) shared task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, 2018.

[48] Amir Soleimani, Christof Monz, and Marcel Worring. Bert for evidence retrieval and claim verification. In *Advances in Information Retrieval*, pages 359–366. Springer International Publishing, 2020.

[49] Dominik Stammbach and Guenter Neumann. Team DOMLIN: Exploiting evidence enhancement for the FEVER shared task. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, 2019.

[50] Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. Fine-grained fact verification with kernel graph attention network. In *ACL*, 2020.

[51] Wanjun Zhong, Jingjing Xu, Duyu Tang, Zenan Xu, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. Reasoning over semantic-level graph for fact checking. *arXiv preprint arXiv:1909.03745*, 2019.

[52] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, 2017.

[53] Jie Zhou, Xu Han, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. GEAR: Graph-based evidence aggregating and reasoning for fact verification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

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

[55] Ye Zhang and Byron Wallace. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2017.

[56] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 1997.

[57] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[58] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality.

In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, 2013.

[59] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[60] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018.

[61] Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. In *Advances in neural information processing systems*, pages 2377–2385, 2015.

[62] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[63] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[64] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, 2019.

[65] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[66] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[67] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*, 2019.

[68] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2020.

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

[70] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

[71] Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015.

[72] Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomás Kociský, and Phil Blunsom. Reasoning about entailment with neural attention. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[73] Caiming Xiong, Victor Zhong, and Richard Socher. Dynamic coattention networks for question answering. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[74] Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017.

[75] Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. Attention-over-attention neural networks for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017.

[76] Alessandro Sordoni, Philip Bachman, Adam Trischler, and Yoshua Bengio. Iterative alternating neural attention for machine reading. *arXiv preprint arXiv:1606.02245*, 2016.

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

[78] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448, 2015.

[79] Yoon Kim, Carl Denton, Luong Hoang, and Alexander M. Rush. Structured attention networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[80] Kehai Chen, Rui Wang, Masao Utiyama, Eiichiro Sumita, and Tiejun Zhao. Syntax-directed attention for neural machine translation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, 2018.

[81] Zhuosheng Zhang, Yuwei Wu, Junru Zhou, Sufeng Duan, Hai Zhao, and Rui Wang. Sg-net: Syntax-guided machine reading comprehension. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[82] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *Advances in neural information processing systems*, pages 2692–2700, 2015.

[83] Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[84] Yi Tay, Anh Tuan Luu, Siu Cheung Hui, and Jian Su. Densely connected attention propagation for reading comprehension. In *Advances in Neural Information Processing Systems*, pages 4906–4917, 2018.

[85] Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. 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.

[86] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, 2014.

[87] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[88] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations (ICLR) 2015*, 2015.

[89] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[90] Ming Yan, Jiangnan Xia, Chen Wu, Bin Bi, Zhongzhou Zhao, Ji Zhang, Luo Si, Rui Wang, Wei Wang, and Haiqing Chen. A deep cascade model for multi-document reading comprehension. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

[91] Minghao Hu, Yuxing Peng, Zhen Huang, and Dongsheng Li. Retrieve, read, rerank: Towards end-to-end multi-document reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[92] Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. Multi-step retriever-reader interaction for scalable open-domain question answering. In *International Conference on Learning Representations*, 2019.

[93] Yelong Shen, Po-Sen Huang, Jianfeng Gao, and Weizhu Chen. Reasonet: Learning to stop reading in machine comprehension. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017.

[94] Shuohang Wang, Mo Yu, Xiaoxiao Guo, Zhiguo Wang, Tim Klinger, Wei Zhang, Shiyu Chang, Gerry Tesauro, Bowen Zhou, and Jing Jiang. R 3: Reinforced ranker-reader for open-domain question answering. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[95] Kai Sheng Tai, Richard Socher, and Christopher D. Manning. Improved semantic representations from tree-structured long short-term memory networks. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2015.

[96] Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. Ordered neurons: Integrating tree structures into recurrent neural networks. In *International Conference on Learning Representations*, 2019.

[97] Yaushian Wang, Hung-Yi Lee, and Yun-Nung Chen. Tree transformer: Integrating tree structures into self-attention. 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)*. Association for Computational Linguistics, 2019.

[98] Suyang Zhu, Shoushan Li, and Guodong Zhou. Adversarial attention modeling for multi-dimensional emotion regression. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2019.

[99] Minghao Hu, Furu Wei, Yuxing Peng, Zhen Huang, Nan Yang, and Dongsheng Li. Read + verify: Machine reading comprehension with unanswerable questions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

[100] Zhuosheng Zhang, Jun jie Yang, and Hai Zhao. Retrospective reader for machine reading comprehension. *ArXiv*, 2020.

[101] Souvik Kundu and Hwee Tou Ng. A nil-aware answer extraction framework for question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2018.

[102] Fu Sun, Linyang Li, Xipeng Qiu, and Yang Liu. U-net: Machine reading comprehension with unanswerable questions. *arXiv preprint arXiv:1810.06638*, 2018.

[103] James Thorne and Andreas Vlachos. Adversarial attacks against fact extraction and verification. *arXiv preprint arXiv:1903.05543*, 2019.

[104] Tal Schuster, Darsh J Shah, Yun Jie Serene Yeo, Daniel Filizzola, Enrico Santus, and Regina Barzilay. Towards debiasing fact verification models. *arXiv preprint arXiv:1908.05267*, 2019.

[105] Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W Black. Style transfer through back-translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2018.