A Review on Fact Extraction and Verification

GIANNIS BEKOULIS, ETRO Department, Vrije Universiteit Brussel, Belgium and imec, Belgium

CHRISTINA PAPAGIANNOPOULOU and NIKOS DELIGIANNIS, ETRO Department, Vrije Universiteit Brussel, Belgium and imec, Belgium

We study the fact checking problem, which aims to identify the veracity of a given claim. Specifically, we focus on the task of Fact Extraction and VERification (FEVER) and its accompanied dataset. The task consists of the subtasks of retrieving the relevant documents (and sentences) from Wikipedia and validating whether the information in the documents supports or refutes a given claim. This task is essential and can be the building block of applications such as fake news detection and medical claim verification. In this paper, we aim at a better understanding of the challenges of the task by presenting the literature in a structured and comprehensive way. Moreover, we describe the proposed methods by analyzing the technical perspectives of the different approaches and discussing the performance results on the FEVER dataset, which is the most well-studied and formally structured dataset on the fact extraction and verification task. We also conduct the largest experimental study to date on identifying beneficial loss functions for the sentence retrieval component. Our analysis indicates that sampling negative sentences is important for improving the performance and decreasing the computational complexity. Finally, we describe open issues and future challenges, and we motivate future research in the task.

CCS Concepts: • **Computing methodologies** → **Information extraction**.

Additional Key Words and Phrases: Fact Extraction, Claim Verification, Fake News, Sentence Retrieval

1 INTRODUCTION

Nowadays we are confronted with a large amount of information of questionable origin or validity. This is not a new problem as it has appeared since the very first years of the printing press. However, it attracted a growing interest with the wide use of social media streams as online news sources. On a daily basis, a large audience accesses various media outlets such as news blogs, etc., rapidly consuming a vast amount of information with possibly inaccurate or even misleading context. The proliferation of the misleading context happens really quickly due to the fast dissemination of news across various media streams.

Recently, a lot of research in the natural language processing (NLP) community has been focused on detecting whether the information coming from news sources is fake or not. Specifically, automated fact checking is the NLP task that aims at determining the veracity of a given claim [106]. Since the main units of a document are sentences, the goal of the fact checking system is to

Authors' addresses: Giannis Bekoulis, gbekouli@etrovub.be, ETRO Department, Vrije Universiteit Brussel, Pleinlaan 2, Brussels, 1050, Belgium, imec, Kapeldreef 75, Leuven, 3001, Belgium; Christina Papagiannopoulou, cppapagi@gmail.com; Nikos Deligiannis, ndeligia@etrovub.be, ETRO Department, Vrije Universiteit Brussel, Pleinlaan 2, Brussels, 1050, Belgium, imec, Kapeldreef 75, Leuven, 3001, Belgium.

automatically identify the nature of the relationship between any two sentences (e.g., if they contradict or support one another). To this end, such systems require a certain level of understanding the language and the coherence of textual units. Thus, the methods that have been proposed to solve the fact verification problem essentially belong to the broader family of Natural Language Inference (NLI) systems [12]. However, we should note that an NLI system (i.e., used to validate the veracity of a claim) requires all the necessary information (i.e., the claim and the potential evidence sentences which contradict or support the claim) to be available upfront. For this reason, retrieval systems are also important in order to identify the relevant and trustworthy evidence sentences coming from various sources (e.g., Wikipedia).

In this paper, we study the fact checking problem [106]. In particular, we focus on the recently introduced task of FEVER and its accompanied dataset [99]. Given an artificially constructed claim and Wikipedia documents, the goal of the task is to validate the veracity of the claim. For that, a dataset comprising 145,449 training claims has been introduced in the work of Thorne et al. [99] and two competitions (i.e., shared tasks) have been organised [101, 102], where several models have been proposed. Since then, the task has received a lot of attention from the NLP community and several complex architectures – relying, for example, on Graph Neural Networks (GNNs) [44] or language models [25] – have been presented in top-tier NLP venues to resolve the task which is far from being solved (i.e., score of 74.27 percentage points in terms of FEVER score).

This work aims at summarizing the methods introduced to resolve the FEVER task, which has the specific characteristic that it comprises multiple subtasks. Our goal is to present a comprehensive study and analysis of the methods that are used in the context of claim extraction and verification. We focus on the FEVER dataset due to that it is the most popular, well-studied and formally structured dataset about the task of claim extraction and verification. Unlike the work of Thorne and Vlachos [96] that aims at providing a high level overview of fact checking in terms of terminology and methods, we aim at thoroughly examining the methods that have been developed for the FEVER task from a more technical perspective. Specifically, we describe the various technologies used in the methods, highlight the pros and cons of each architecture, discuss the contribution of each specific component in each proposed system and present the results for each individual subtask of the FEVER task. That way, we are able to compare each and every method by quantifying the benefit of each and every component of these methods. By comparing several methods on the same dataset, we are able to perform a fair and direct comparison of the various architectures. Moreover, as mentioned above, the task consists of three subtasks, making it challenging enough to act as a common ground to evaluate and benchmark various approaches. In the various works that try to solve the FEVER task, researchers use different evaluation metrics, data splits, assumptions and thus, it is not straightforward for one to compare them and identify similar and different aspects among them. This is exactly the gap that our work covers; namely, to present in a structured and comprehensive way the various methods introduced for the FEVER task. Thus, our study serves as an extensive review for the FEVER task and has as a goal to help future researchers to improve the current state-of-the-art by 1) providing an easy way to compare systems and experiments, and 2) investigating the contributions of each individual model component in the overall performance. On top of that – to the best of our knowledge – we conduct the largest experimental study on the sentence retrieval subtask. The results of our study point researchers to certain directions such as (i) that sentence sampling can be beneficial (for example, in avoiding dataset imbalance and reducing computational complexity), or (ii) that working on the sentence retrieval step can lead to same performance improvement as in the case that one would work on the claim verification step.

> **Claim:** Claire Danes is wedded to an actor from England.
> **[wiki/Claire_Danes]** She is married to actor Hugh Dancy, with whom she has one child.
> **[wiki/Hugh_Dancy]** Hugh Michael Horace Dancy (born 19 June 1975) is an English actor and
>   model.
> **Verdict:** SUPPORTED

> **Claim:** Rogue appears in Canadian comic books.
> **[wiki/Rogue_(comics)]** Rogue is a fictional superhero appearing in American comic books
>   published by Marvel Comics, commonly in association with the X Men .
> **Verdict:** REFUTED

Fig. 1. Two examples from the FEVER dataset where evidence sentences should be selected from Wikipedia. In the example illustrated on top, the claim is SUPPORTED and the relevant information identified in Wikipedia is indicated in blue color. In the example illustrated in the bottom, the claim is REFUTED by the evidence sentence.

2 BACKGROUND

In this section, we (i) define the FEVER task and the problem it solves (see Section 2.1), (ii) describe the way that the dataset is constructed (see Section 2.2), and (iii) present the subtasks of the FEVER task (see Section 2.3).

2.1 Problem Definition

The FEVER shared task provides a set of claims where each claim is a sentence of which the veracity should be identified. The veracity of a claim should be based on (sentence-level) evidence provided by Wikipedia. For that, a set of pre-processed (from the year 2017) Wikipedia documents has been shared with the participants of the competition. A claim can be either SUPPORTED or REFUTED, assuming that correct evidence has been identified. In the case that there is not enough information in Wikipedia, the veracity of the claim should be assessed as NOTENOUGHINFO (NEI). The goal of the task is to return for each claim either the SUPPORTED/REFUTED label along with the corresponding evidence or the NEI label without evidence. In 16.82% of the claims, more than one evidence sentences are needed to conclude about the veracity of the claim (the evidence is from different pages/documents in 12.15% of the claims). Two examples of the FEVER dataset are illustrated in Fig. 1.

2.2 Dataset Construction

The FEVER dataset includes 185,445 claims and the exact numbers of the examples per label (i.e., SUPPORTED, REFUTED, NEI) for the training, development and test sets are presented in Table 1. The FEVER dataset been constructed in two phases: (i) the claim generation and (ii) the claim labeling phase.

In total, 50 annotators have contributed in the process. In phase (i), the annotators created claims from randomly chosen Wikipedia sentences. The claims should be sentences that include a single piece of information. The goal of the claim generation phase is to create claims that are not trivially verifiable (i.e., too similar to the source) nor too complex. For that, hyperlinks have been included in the sentences in order for the annotators to incorporate external knowledge in a controlled way. Except for the original claims, the annotators created variations of the claims by, for example, paraphrasing, adding negation. For the claims that were the negated versions of the original claims, the authors have observed that only trivial negations were generated (i.e., by adding only the word

"not"). To alleviate this issue, the annotation interface has been re-designed to highlight the "not" trivial negations. In phase (ii) of the dataset construction process, the annotators were asked to label the claims as SUPPORTED, REFUTED or NEI. For the SUPPORTED and REFUTED labels, the annotators also provided the sentences that have been used as evidences for supporting or refuting the veracity of the claim. For the NEI label, only the label itself was provided since the annotator could not conclude whether the claim was supported or refuted based on the available Wikipedia sentences. Finally, to improve the quality of the provided dataset, (i) super-annotators checked randomly 1% of the data, (ii) the Fleiss κ score [29] (which measures the inter-annotator agreement among a fixed number of annotators when assigning categorical labels to a number of instances) for 4% of randomly selected claims has been calculated among five annotators, and (iii) the authors have manually re-validated the quality of the constructed dataset (for 227 examples).

2.3 FEVER Subtasks

In the literature, the FEVER task has been mostly treated as a series of three subtasks, namely document retrieval, sentence retrieval and claim verification [63, 99, 119].

Split	SUPPORTED	REFUTED	NEI
Train	80,035	29,775	35,639
Dev	6,666	6,666	6,666
Test	6,666	6,666	6,666

Table 1. The statistics of the FEVER dataset as presented in Thorne et al. [101].

2.3.1 Document Retrieval. Document retrieval is the task that aims at matching a query against a collection of unstructured documents and return the most relevant articles [15] (see Section 3.1 for more details).

2.3.2 Sentence Retrieval & Claim Verification. The FEVER task also consists of the subtasks of sentence retrieval and claim verification. The sentence retrieval is the task that has as goal to retrieve the relevant sentences out of a given document, or a set of documents, for a given query (i.e., claim in the context of FEVER). The claim verification task aims at verifying the veracity of a given claim. In the context of FEVER, the veracity of a claim is assessed by taking the retrieved evidences from the sentence selection subtask into account. For a detailed description of the FEVER subtasks see Section 3.1.

2.4 Three-Step Baseline Model

Along with the FEVER dataset, Thorne et al. [99] provided a three-step pipeline model to solve the FEVER task. The three subtasks are (i) the *document retrieval* step, where the goal is to retrieve relevant documents from Wikipedia for a given claim, (ii) the *sentence retrieval* step, which is responsible for retrieving relevant evidence sentences from the documents retrieved from step (i), and (iii) the *claim verification* step, which aims at predicting the correct label (i.e., SUPPORTED, REFUTED or NEI as defined in Section 2.1). A graphical illustration of the three-step pipeline model is provided in Fig. 2. Most of the existing works so far (see Zhao et al. [120], Zhong et al. [121]) are also following this three-step pipeline approach; however, more complex architectures have been proposed to solve the FEVER task in an end-to-end fashion [117].
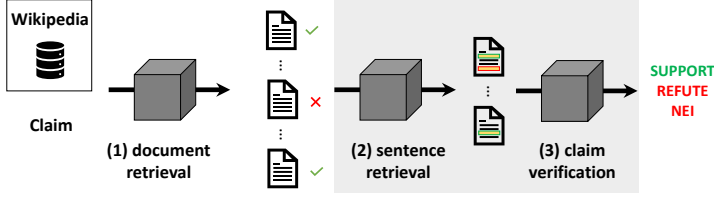
Fig. 2. A three-step pipeline model for the FEVER task. It consists of three components, namely document retrieval, sentence retrieval and claim verification. The input to the first component is Wikipedia and a given claim sentence. The output of the first component is a set of Wikipedia documents related to the claim. The retrieved documents are fed as input to the sentence retrieval component and the output of that module is a set of sentences related to the claim from the input documents. Finally, the input to the claim verification component is the retrieved sentences from step (2) and the output is a label which indicates the veracity of the claim. Note that the claim is provided as input to every component of the pipeline system. The shaded box illustrates the fact that in several systems, steps (2) and (3) are performed in a joint setting.

2.5 Evaluation

In this subsection, we describe the evaluation metrics that are used for evaluating the performance in the different FEVER subtasks. Note that since the three subtasks are stacked the one on top of the other, higher performance in the downstream components (e.g., document retrieval) leads to better performance on the upstream components (e.g., sentence retrieval and claim verification). The organizers of the FEVER shared task [101] have released a Github repository with the code of their evaluation module[1].

2.5.1 Document Retrieval. The evaluation of the results for the subtask of document retrieval is based on the work of Thorne et al. [99], which defined that the task relies on two metrics, namely, *oracle accuracy* and *fully supported*. The *fully supported* metric indicates the number of claims for which the correct documents (i.e., along with the corresponding evidences) have been fully retrieved by the document retrieval component. This metric takes only into account the claims that are supported/refuted by evidences (i.e., does not consider the NEI class). The *oracle accuracy* is the upper bound of the accuracy over all the three classes (i.e., considers the claims of the NEI class as correct).

2.5.2 Sentence Retrieval. The evaluation of this subtask is performed by using *precision*, *recall* and F_1 scores. Specifically, the organizers of the shared task suggested the precision to count the number of the correct evidences retrieved by the sentence retrieval component with respect to the number of the predicted evidences for the supported/refuted claims. The recall has also been exploited for the supported/refuted claims. A claim is considered correct in the case that at least a complete evidence group is identified. Finally, the F_1 score is calculated based on the aforementioned metrics.

2.5.3 Claim Verification. The evaluation of the claim verification subtask is based on the *label accuracy* and the *FEVER score* metrics. The label accuracy measures the accuracy of the label predictions (i.e., SUPPORTED, REFUTED and NEI) without taking the retrieved evidences into account. On the other hand, the FEVER score counts a claim as correct if a complete evidence group has been correctly identified (for the supported/refuted claims) as well as the corresponding label. Thus, the FEVER score is considered as a strict evaluation metric and it was the primary metric for ranking the systems on the leaderboard of the shared task.

---

[1]https://github.com/sheffieldnlp/fever-scorer

Model	Document Retrieval					Sentence Retrieval				Claim Verification							
	Term-based			DRQA	Features	TF-IDF	ESIM	LM	Other	Neural			Simple	LM		Joint	
	Mention	Keyword	Exact Match							ESIM	LSTM/CNN	DA		Graph	Seq2seq	Supervised	Generated
2018																	
Hanselowski et al. [32]	✓					✓	✓			✓							
Thorne et al. [99]	✓					✓				✓		✓					
Yoneda et al. [119]	✓				✓	✓				✓						✓	
Yin and Schütze [118]	✓			✓		✓					✓						
Hidey and Diab [36]	✓			✓	✓	✓	✓				✓						
Chakrabarty et al. [14]	✓			✓		✓					✓						
Malon [56]		✓							✓		✓						
Luken et al. [54]	✓		✓						✓		✓						
Yin and Roth [117]	✓					✓						✓				✓	
Taniguchi et al. [95]			✓						✓								
2019																	
Nie et al. [63]	✓	✓					✓			✓							
Nie et al. [64]	✓	✓						✓					✓				
Ma et al. [55]	✓	✓				✓	✓		✓		✓		✓				
Zhou et al. [122]	✓							✓					✓	✓			
Chernyavskiy and Ilvovsky [19]	✓												✓				
Stammbach and Neumann [91]	✓												✓				✓
Jobanputra [42]			✓														
Tokala et al. [103]									✓		✓						
2020																	
Zhao et al. [120]	✓							✓						✓			
Liu et al. [53]	✓							✓						✓			
Soleimani et al. [89]	✓						✓	✓					✓	✓			
Zhong et al. [121]		✓				✓								✓			
Portelli et al. [77]		✓											✓				
Lee et al. [46]													✓				
Lewis et al. [47]		✓					✓								✓	✓	
Nie et al. [62]	✓							✓	✓	✓						✓	
Chen et al. [16]	✓							✓	✓				✓				✓
Ye et al. [116]	✓							✓	✓				✓				✓
Subramanian and Lee [92]	✓																

Table 2. Timeline with the works that have been developed so far for the FEVER task, grouped based (i) on the year and (ii) in a similar way to the one presented in Section 3. LM stands for language model based approaches and the ✓ symbol indicates whether a model uses a particular method. Note that most of the works developed in 2019–2020 rely on pre-existing document retrieval components and the main focus is on the sentence retrieval and the claim verification components.

3   METHODS

In this section, we describe the various methods that have been developed so far for solving the FEVER task. Most of the existing studies in the literature [32, 121] divide the task into a series of three subtasks (i.e., document retrieval, sentence selection and claim verification, see Section 3.1 for a detailed description) similar to the baseline model as described in Section 2.4. However, there are some studies that merge the two subtasks of sentence selection and claim verification into one (see Fig. 2) mostly by exploiting multi-task learning architectures [62, 117]. For a detailed description of these joint architectures, we refer to Section 3.2. Table 2 presents a timeline that summarizes the architectures developed so far for the FEVER task.

3.1   Pipeline Models

3.1.1   Document Retrieval. In this subsection, we describe the main methods that have been proposed for the document retrieval task. The input to the document retrieval step is Wikipedia and a given claim sentence. The output of this module is a set of relevant to the claim Wikipedia documents.

Preliminaries
DrQA: Several approaches, which have been exploited to partly solve the FEVER task, rely on the DrQA component [15] for retrieving relevant information from Wikipedia. The goal of DrQA is to answer questions on open-domain datasets such as Wikipedia. DrQA consists of two components (i) the *document retriever*, which is responsible for identifying relevant articles, and (ii) the *document reader*, which is responsible for pointing to the start and end positions of the answers inside the document or a set of documents. However, most of the existing literature on the FEVER task uses only component (i) (i.e., the document retriever) to collect relevant documents from Wikipedia (see Section 2.1). Specifically, the document retriever does not rely on machine learning methods. It calculates an inverted index lookup, computes the TF-IDF bag-of-words representations (bigrams) and scores the articles with the questions based on the aforementioned word vector representations.

Baseline
For this subtask, Thorne et al. [99] exploited the DrQA module, and used cosine similarity to obtain the k most similar documents to the claim based on the TF-IDF word representation.

Related work
Mention-based methods: Hanselowski et al. [32] proposed a mention-based approach to retrieve the relevant documents from Wikipedia for a given claim. This method consists of three components, namely, (i) mention extraction, (ii) candidate article search, and (iii) candidate filtering. Component (i) relies on a constituency parser, as developed in the work of Gardner et al. [30]. Based on the parser, every noun phrase in a claim is considered as a potential entity. In addition, they employ heuristics based on which potential entity mentions are considered all words before the main verb of the claim and the whole claim itself. Component (ii), presented in the work of Hanselowski et al. [32], uses an external search API[2] in order to match the potential entity mentions identified by component (i) in the titles of Wikipedia articles. Component (ii) also returns some Wikipedia titles that are longer than the entity mentions. To deal with this case, component (iii) is responsible for stemming the Wikipedia title as well as the claim and discard all titles that are not part of the claim. The methodology of the work presented in Hanselowski et al. [32] is also followed by Chernyavskiy and Ilvovsky [19], Liu et al. [53], Soleimani et al. [89], Stammbach and Neumann [91], Zhao et al.

---

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

[120], Zhou et al. [122], sometimes with minor modifications. Another piece of work that leverages the value of named entities is that of Chakrabarty et al. [14], Hidey and Diab [36], Malon [56], Yin and Roth [117]. The work of Chakrabarty et al. [14], except for named-entity recognition uses the Google custom search API and dependency parsing as another task that improves the coverage of the retrieved documents. The works of Chakrabarty et al. [14] and Malon [56] also exploit disambiguation information, e.g., whether the Wikipedia title refers to a "film" (e.g., Titanic might refer to either the ship or the movie).

Keyword-based methods: The work of Nie et al. [63] ranked at the first position at the FEVER competition. They presented a three-stage model that relies on Neural Semantic Matching Network (NSNM), i.e., a variation of ESIM [17] (see Section 3.1.2). For the document retrieval, they exploit a keyword-matching approach that relies on exact matching (between the Wikipedia title and the spans of the claim), article elimination (i.e., remove the first article, e.g., "a") and singularization (every token is considered as a span when no documents are returned). Afterwards, all documents that do not contain disambiguative information (e.g., "band", "movie") are added in the retrieved document list. The rest of the documents (i.e., those with disambiguative information) are ranked and filtered out using NSNM and a threshold value. Several works [55, 64, 77, 121] exploit the document retrieval module developed by Nie et al. [63]. The work of Luken et al. [54] is also designed to extract part-of-speech tags, dependencies, etc. by using the CoreNLP parser [57] for keyphrase identification.

Other methods: However, there are some methods that do not fall into any of the aforementioned categories. The work of Yin and Schütze [118] relies on the work of the baseline model [99]. Similar to the baseline model, the work of Hidey and Diab [36] exploit DrQA along with hand-crafted features and neural methods for the document retrieval task. Yoneda et al. [119] design hand-crafted features such as position, capitalization in the claim and train a logistic regression classifier. Unlike most of the works on the document retrieval FEVER subtask that aim for high recall, the work of Taniguchi et al. [95] aims for high precision using exact matching techniques. Similar to this work, the work of Tokala et al. [103] relies on exact matching methods in order to reduce the number of available documents, but it also relies on a Bidirectional Attention Flow for Machine Comprehension (BIDAF) model [84] in order to rank the remaining documents. In addition, as we observe in Table 2, many of the works that have been developed for the competition shared task (2018) focus on hand-crafted features (see categories "Exact Match", "DrQA", and "Features"). However, this is not the case for more recent works (2019-2020) that focus mostly on the sentence retrieval and claim verification components and use mention- and keyword-based approaches. These works are classified in separate categories in the timeline (Table 2), see columns "Exact Match", "DrQA", "Features", so that the reader can identify each component easily.

3.1.2   Sentence Retrieval. In this subsection, we describe the main methods that have been proposed for the sentence retrieval component. The input to the sentence retrieval step is the Wikipedia documents retrieved from the previous component and the given claim sentence. Each Wikipedia document consists of sentences and the relevant to the claim sentences, the so-called evidences, are the output of the second component.

Preliminaries
The tasks of sentence retrieval and claim verification are commonly framed as NLI problems (i.e., are treated with methods originally developed to solve NLI tasks). This does not mean that the aforementioned methods are pre-trained models on commonly used NLI datasets, but rather that

the input of each subtask (i.e., sentence retrieval and claim verification) is framed in such a way that traditional NLI methods can be used to resolve each of these two subtasks.

Assuming that we have two sentences, the *hypothesis* and the *premise* sentence, the goal of the NLI task is to determine whether the *premise* sentence *entails*, *contradicts* or is *neutral* to the *hypothesis*. The most well-known datasets for NLI are the Stanford Natural Language Inference (SNLI) Corpus [12], the Multi-Genre Natural Language Inference (MultiNLI) Corpus [111], and the cross-lingual NLI (XNLI) Corpus [21]. Several approaches have been proposed to solve the NLI tasks; however, the neural models that have been mostly explored in the context of the FEVER task are the Decomposable Attention (DA) [71], Enhanced Long Short-Term Memory (LSTM) for Natural Language Inference (ESIM) [17], and Bidirectional Encoder Representations from Transformers (BERT)-based NLI [25].

DA: The model has been proposed in the work of Parikh et al. [71] and unlike the trend of using LSTMs, DA solely relies on word embeddings in order not to increase the complexity by $O(d^2)$, where d is the size of the hidden dimension. Specifically, DA consists of three components (i) the *attention* step, which computes soft-alignment scores between the two sentences (i.e., the premise and the hypothesis) similar to the method of Bahdanau et al. [8], (ii) the *comparison* step, which applies a feed-forward neural network with a non-linearity between the aligned representations and (iii) the *aggregation* step, which combines the information from previous steps via a summation operation to predict the final label.

ESIM: Chen et al. [17] rely on LSTM models to perform the NLI task. In particular, the model exploits the use of bidirectional LSTMs [39] (i.e., on top of the word embeddings) to form representations of the premise and the hypothesis sentences. A soft-alignment layer that calculates attention weights similar to the DA model [71] is being used. In addition to the original representations, operations such as the difference and the element-wise product between the LSTM and the attended representations are calculated to model complex interactions. In the next layer, LSTMs are also exploited to construct the representation for the prediction layer. Finally, unlike DA, in the prediction layer, average and max pooling operations are used for the prediction of the final label.

BERT: Pre-trained Language Models (LM) have revolutionized the way that NLP tasks are solved. Examples include ELMo (Embeddings from Language Models) [73], OpenAI GPT [79] and BERT [25]. In the context of the FEVER task, several models [53, 89] rely on the pre-trained BERT model. The BERT relies on WordPiece embeddings [113] and on Transformer networks [104]. The input to the BERT model is either a single sentence or a pair of sentences encoded in a single sequence. The first token is always the special token [CLS], which is used in classification tasks, and the sentences are separated by the special [SEP] symbol. Two approaches have been proposed to pre-train the BERT model; specifically, (i) the Masked LM, where a percentage of random Word-Piece tokens are masked and the goal is to predict the masked tokens, and (ii) the Next Sentence Prediction task, where the goal is to validate (0 or 1) whether the second sentence is the sequel of the first one. The pre-training task (ii) has been shown to be extremely useful for downstream tasks such as Question Answering (QA) and NLI. For finetuning BERT for NLI, the sentence pair is separated by the [SEP] symbol and the classification label is predicted on top of the [CLS] symbol.

Baseline

For sentence selection, in the proposed three-step model, Thorne et al. [99] obtained the most similar sentences from the retrieved documents (see previous subtask) by using either DrQA or

unigram TF-IDF vectors. Moreover, they used a cut-off threshold tuned on the development set.

Related work

TF-IDF: For the sentence retrieval task, several pipeline methods in the literature rely on the sentence retrieval component of the baseline method [99]. Specifically, these methods [19, 77, 95, 118], use a TF-IDF vector representation along with a cosine similarity function. However, there are some attempts that exploit additional representations such as ELMo embeddings [14].

ESIM-Based: An important line of research [32, 63, 122] for the sentence selection subtask of the FEVER task includes the use of ESIM-based models [17]. Those works formulate the sentence selection subtask as an NLI problem, where the claim is the "premise" sentence and the potential evidence sentence is a "hypothesis" sentence. Hanselowski et al. [32] proposed a modified version of ESIM that during training receives as input the claim and the ground truth evidence sentences, as well as the claim with negative examples, randomly selected from the Wikipedia documents that the positive samples (i.e., ground truth evidences) are coming from (i.e., they sample randomly five sentences by not including the positive ones). The loss function used in this work is a hinge loss that receives as inputs the positive and the negative ranking scores (as pairs) from the ESIM model. At test time, the model computes the ranking score between the claim and each potential evidence sentence. It is also worth mentioning that the work of Zhou et al. [122] exploits the evidences retrieved by the model of Hanselowski et al. [32]. Similar to Hanselowski et al. [32], Nie et al. [63] use the same variation of ESIM called NSNM which has been exploited by the document retrieval component as well (see the keyword-based methods in Section 3.1.1). Nie et al. [63] calculate the NSMN score between the claim and the evidence sentences. Afterwards threshold-based prediction is used to retain the highest scoring sentences. Unlike Hanselowski et al. [32] that train a pairwise hinge loss, Nie et al. [63] exploit a cross-entropy loss for training their model.

Language Model Based: Similar to the ESIM-based methods, language model based methods [53, 64, 89, 92, 120, 121] transform the sentence retrieval task to an NLI problem using pre-trained language models. The pre-trained language models are finetuned for the NLI task similar to the procedure described for the BERT-based model. It is however worth mentioning that the models developed for the sentence retrieval component do not rely only on BERT but also on RoBERTa [51] and XLNet [115]. For the language model based sentence retrieval two types of losses have been exploited: (i) the pointwise loss, where a cross-entropy classifier is used to predict 0 or 1 (or a probability value), depending on whether the claim and the potential evidence sentence are related, and (ii) the pairwise loss, where the loss function takes as input a negative and a positive example. In that case, the positive example is the concatenation of the claim with an evidence sentence from the ground truth while a negative example is a concatenation between the claim and a negative example (i.e., potential evidence, not included in the evidence set). That way, the model is learning to maximize the margin between the positive and the negative examples. For the loss of type (i), several works that use the BERT pre-trained model have been proposed [64, 89]. Unlike the previous works that use the pointwise loss, the work of Zhong et al. [121] use the RoBERTa and XLNet pre-trained models. For the loss of type (ii), the proposed architectures rely only on the BERT pre-trained language model [53, 89, 120]. Due to the high number of negative examples with respect to the number of positive examples in the sentence retrieval subtask, Soleimani et al. [89] proposed to use hard negative mining similar to Schroff et al. [82] to select more difficult examples (i.e., those with the highest loss values). Note that training a pairwise loss is more computationally expensive than training a pointwise loss, since in the first case, one should consider all the combinations of positive-negative example pairs. As we observe in Table 2, most recent works (i.e., developed in

2019-2020) focus on developing language model based approaches.

Other Methods: The two-step model of Stammbach and Neumann [91] is able to combine both the ESIM-based and the language model based sentence retrieval components. This work relies on the model of Nie et al. [63] as a first component and uses a BERT-based model with two different sampling strategies to select negative examples. Except for the aforementioned studies that can be grouped into specific categories, there are some methods [54, 69, 91, 119] that make use of different strategies for sentence retrieval. Luken et al. [54] use the root, the nouns and the named entities of the claim and construct a set of rules. For instance, if the named entities and the nouns are included in the sentence then the sentence is added in the evidence set. Similar to the work of Luken et al. [54], Otto [69] also relies on nouns and named entities extracted from the claim by using the spaCy NLP library [40]. This work is able to directly retrieve evidences using the Solr indexer[3] without relying on a document retrieval component. Unlike previous studies [54, 69], Yoneda et al. [119] manually extract features such as the length of the sentences, whether the tokens of the sentence are included in the claim, etc. These features are fed into a logistic regression model. Finally, the work of Tokala et al. [103] relies on the BIDAF model [84], where the input is the claim and each candidate evidence sentence. The model output scores for each evidence sentence and, that way, is able to rank the candidate evidence sentences.

3.1.3 Claim Verification. In general, most of the claim verification methods use neural model components. In the baseline method, the authors have exploited either an MLP neural model or a DA approach. In this section, the work on claim verification is divided into (i) ESIM-based architectures, (ii) language model based approaches, and (iii) other neural models. It is worth mentioning that most of the literature so far is focused on improving the task of claim verification because the previous two subtasks (i.e., document retrieval and sentence selection) have already attained quite good performance in terms of the recall evaluation metric, see Section 4 and Table 2.

Preliminaries
For the claim verification subtask, researchers exploit similar techniques to the ones used for the sentence retrieval subtask. For more details about the techniques that the claim verification models are based on can be found in Section 3.1.2.

Baseline
Two methods were developed for the claim verification component. First, a multi-layer perceptron (MLP) was used by taking as input features the term frequencies of the claim and the evidence and the TF-IDF cosine similarity between them. Second, DA [71] – which was described in Section 3.1.1 – has been used as a state-of-the-art system in NLI [12] (aka Recognizing Textual Entailment (RTE)). It is worthwhile mentioning that, for this step, evidences are needed in order to train the NLI component. However, this is not feasible for the NEI labels, since there are no such evidence sentences in the training set. To circumvent this issue, two strategies have been explored in the baseline model: (i) sampling random sentences from Wikipedia, and (ii) sampling random sentences from the most similar documents as retrieved from the document retrieval component.

Related work
ESIM-based: Hanselowski et al. [32] used an ESIM model for claim verification which has been modified to take as input multiple potential evidence sentences along with the given claim. They exploit the use of attention mechanisms, pooling operations and an MLP classifier to predict the

---

[3]https://lucene.apache.org/solr/

relevant classes (e.g., SUPPORTED, REFUTED, NEI). The winning system of the FEVER task proposed by Nie et al. [63] also relies on a modified version of ESIM called NSMN combined with additional features. This work exploits additional features such as WordNet embeddings (i.e., antonyms, hyponyms), number embeddings and the scores from the previous subtasks. Yoneda et al. [119] use the ESIM model where the claim with each potential evidence (and the associated Wikipedia article title) is considered independently. To aggregate the predictions for each evidence sentence with the claim, Yoneda et al. [119] used an MLP classifier on top of the prediction score of each evidence sentence .

Language Model Based: Language models have also been successfully applied to the claim verification subtask of FEVER. Soleimani et al. [89] formulated the problem of claim verification as an NLI task where the claim (premise) and the potential evidence sentence (hypothesis) are the inputs into a BERT-based language model. The evidences are independently considered against the claim and the final decision is made based on an aggregation rule similar to Malon [56] (i.e., the default label is NEI and, if there is a SUPPORTED label, then the label of the claim is also SUPPORTED). BERT-based models have also been adopted by multiple studies e.g., Chernyavskiy and Ilvovsky [19], Nie et al. [64], Portelli et al. [77], Stammbach and Ash [90], Stammbach and Neumann [91], Subramanian and Lee [92].

GNN-based Language Models: Zhou et al. [122] proposed a BERT-based method that makes use of GNNs [44]. By using GNNs, where evidences are nodes in a graph, they are able to exchange information between the nodes, thereby performing reasoning in order to obtain the final class verification label. Similar to the work of Zhou et al. [122], Liu et al. [53] exploit the use of kernel attention [114], both at sentence and token level, to propagate information among the evidence nodes. A graph-based approach is also explored in the work of Zhong et al. [121], where, unlike previous works, instead of using evidences as nodes in the graph, they construct the graph based on semantic roles (e.g., verbs, arguments) as those extracted by an external library. Then, GNNs and graph attention mechanisms are used to combine and aggregate information among the graph nodes for the final prediction. Zhao et al. [120] rely on a modified version of Transformers which is able to perform multi-hop reasoning even on long text sequences and combine information even along different documents.

Different from the previous works that rely on GNNs, the paper of Ye et al. [116] indicates that extracting coreference information from the text is important for claim verification. Specifically, in this paper, they rely on tasks such as entity masking in order to automatically exploit coreferential relations. Finally, Chen et al. [16] extract the core parts of a claim, then they generate questions about these core parts and that way they are able to conclude about the veracity of the claim by performing reasoning on the question answering pairs. As we observe in Table 2, most recent works (i.e., developed in 2019-2020) focus on developing language model based approaches. These works are classified in separate categories in the timeline (Table 2), see columns "Simple" (simple classification layer), "Graph", "Seq2seq", so that the reader can identify each component easily.

Other Neural Models: Alternative neural models have also been proposed [14, 54, 56, 69, 95, 118]; these cannot be classified in any of the aforementioned categories. Specifically, Chakrabarty et al. [14] rely on bidirectional LSTMs and perform operations combining the claim and evidence representations (e.g., element-wise product, concatenation) similar to Conneau et al. [20]. Other studies use the DA model [54, 69], similar to the one that was exploited in the baseline model. Other methods such as Convolution Neural Networks (CNNs) with attention mechanisms [95, 118] and transformer networks [56] have also been used. Finally, the work of Tokala et al. [103] relies on the BIDAF model [84], where the input is the claim and all the retrieved evidence sentences. These

works and the ESIM-based models are classified in separate categories in the timeline (Table 2), see columns "ESIM", "LSTM/CNN", "DA".

3.2 Joint models

Unlike all the aforementioned studies presented so far in Section 3, which consider the FEVER subtasks in a pipeline setting, there has been a significant amount of work that handles the FEVER subtasks in a joint setting. The main motivation of joint methods is that in the pipeline setting, there are errors that are flowing from one component to the other, while in the case that two or more subtasks are considered together, decisions can be possibly corrected due to the interaction between the components. For instance, Yin and Roth [117] proposed the use of a multitask learning architecture with CNNs and attentive convolutions [118] in order to extract coarse-grained (i.e., sentence-level) and fine-grained (i.e., with attention over the words) sentence representations of the claim and evidences so as to perform the tasks of sentence selection and claim verification in a joint setting. Similar to this work, Hidey and Diab [36] train the sentence selection and claim verification subtasks in a multitask fashion. Specifically, they use the ESIM model for the representation of the claim and the evidence sentences, pointer networks [105] for the sentence selection subtask and an MLP-based architecture for claim verification. A newer version of this system, which uses adversarial instances to improve the performance, has also been proposed in the work of Hidey et al. [35]. Nie et al. [62] perform an experimental study, where the NSNM model [63] is compared in three different setups. In particular, a pipeline setting, a multitask setting and a newly introduced so-called "compounded label set" setting are being compared. This compounded label set setting is a combination of all the labels of the sentence selection and claim verification subtasks.

Unlike the previous line of research that trains the models in a multitask learning fashion, there are works that address some of the subtasks (i.e., the retrieval steps) in an unsupervised manner. Specifically, Jobanputra [42], Lee et al. [46] formulate the problem as a closed task. In the work of Lee et al. [46], the last entity of the claim is masked out and the missing entity is filled in with a language model. That way, a new evidence is being created (eliminating the need for a sentence retrieval module), which is then input together with the claim into an MLP to predict the claim verification label. Similar to that work, Jobanputra [42] also relies on language models and eliminates the sentence retrieval step by masking parts of the claim and generating potential evidence sentences. Finally, the work of Lewis et al. [47] is an end-to-end system that is performing the three steps at once. Specifically, they design a new retrieval system that is able to retrieve relevant documents and then generate text based on the retrieved documents. The model relies on a retriever that outputs a distribution of scores over the documents and a generator that takes also the previous generated tokens and the highest scoring documents into account. In the case of a classification task such as FEVER, where the goal is to predict a label based on a given claim, the sequence-to-sequence model (generator) has as goal (in the decoder part) to predict the tokens of the labels (e.g., SUPPORTED). This model has been exploited in other tasks such as question answering and question generation. These works are classified in separate categories in the timeline (Table 2), see columns "Supervised" and "Generated".

4 RESULTS & DISCUSSION

In this section, we describe the experimental results of the methods presented in Section 3 and we compare their performance. Similar to the previous section, we present the results per subtask along with the corresponding discussion. The performance of the joint models is also presented in the corresponding subsections.

Model	Pre-calculated Features	κ	Fully Supported	Oracle Accuracy
Thorne et al. [99]	✓	5	55.30	70.20
Yin and Roth [117]	✓	5	89.63	93.08
Hidey and Diab [36]	✓	5	90.70	-
Hanselowski et al. [32]	✓	7	-	93.55
Chakrabarty et al. [14]	✓	3	**94.40**	-
Zhou et al. [122]	✓	7	-	93.33
Nie et al. [63]	✓	5	-	92.42
Tokala et al. [103]	✓	5	-	**94.05**

Table 3. Results of the document retrieval task in terms of the oracle accuracy and the fully supported evaluation metrics in the dev set. The pre-calculated features column indicates whether a model uses external NLP tools or hand-crafted features. The symbol κ is the number of the retrieved documents. The best performing models per column are highlighted in bold font. Missing results are not reported in the original papers.

4.1 Document Retrieval

In Table 3, we present the results of the various document retrieval components that were extensively presented in Section 3.1.1. We evaluate the performance of the models based on the two commonly used evaluation metrics (i.e., fully supported and oracle accuracy) for the document retrieval step of the FEVER task as introduced in the work of Thorne et al. [99, 101] and presented in Section 2.5.2. In Table 3, the pre-calculated features column indicates whether external NLP tools (e.g., dependency parser) or hand-crafted features are exploited. In the k column, the number of retrieved documents per claim is presented. We report the k's of each work that lead to best results or those that are used in the results of the pipeline systems. The model of Thorne et al. [99] is the baseline model as presented in Section 2.4. It is worth mentioning that in the various works, there is no consistent report on the various metrics. The results are reported on the dev set since there is no ground truth data for this subtask on the test set. Specifically, the evaluation metrics on the competition platform[4] for the test set assess the performance only of the subtasks two (i.e., sentence retrieval) and three (i.e., claim verification). Almost all of the systems presented in Table 3 rely either on mention- or keyword-based approaches, except for the baseline model that relies on TF-IDF features to obtain the most relevant documents. The works of Hanselowski et al. [32] and Nie et al. [63] are those that most of the recent neural methods (see Section 3.1.1 for more details) rely upon. The document retrieval presented in the work of Zhou et al. [122] reproduces the results of the Hanselowski et al. [32] model. In terms of oracle accuracy score, the model of Chakrabarty et al. [14] is the best performing one; however, the models are not directly comparable since the oracle accuracy is measured based on a different number of retrieved documents. We observe that all the models [14, 36, 117], which rely on mention-based document retrieval, achieve higher performance compared to the baseline model with respect to the fully supported evaluation metric. The same holds for the oracle accuracy evaluation metric. The keyword-based model of Nie et al. [63] also scores better compared to the baseline model. The gap between the keyword-based model of Nie et al. [63] is relatively small in terms of the fully supported evaluation metric compared to the mention-based approaches. The model of Tokala et al. [103] performs on par with the mention- and keyword-based approaches since it relies on exact match techniques and the BIDAF model [84]. We expect that there are no significant differences between the terms that are identified between the mention- and keyword-based (or the exact match) approaches since all of these approaches are able to identify the core parts of the document collections leading to similar performance as

---

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

	Model	Pre-calculated Features	Dev			Test		
			Precision	Recall	F_1	Precision	Recall	F_1
TF-IDF	Thorne et al. [99]	✓	-	-	17.20	11.28	47.87	18.26
	Taniguchi et al. [95]	✓	-	-	-	11.37	29.99	16.49
	Chakrabarty et al. [14]	✓	-	78.04	-	23.02	75.89	35.33
ESIM-based	Hanselowski et al. [32]	✗	-	87.10	-	23.61	85.19	36.97
	Zhou et al. [122]	✗	24.08	86.72	37.69	23.51	84.66	36.80
	Nie et al. [63]	✗	36.49	86.79	51.38	42.27	70.91	52.96
Language models	Nie et al. [64]	✗	-	-	**76.87**	-	-	**74.62**
	Soleimani et al. [89]**	✗	38.18	88.00	53.25	-	-	38.61
	Liu et al. [53]	✗	27.29	**94.37**	42.34	25.21	**87.47**	39.14
	Subramanian and Lee [92]	✗	-	90.50	-	-	-	-
	Zhong et al. [121]	✗	26.67	87.64	40.90	25.63	85.57	39.45
Other models	Luken et al. [54]	✓	**77.50**	52.30	62.50	**77.23**	47.12	58.53
	Otto [69]	✓	-	-	-	12.09	51.69	19.60
	Yoneda et al. [119]**	✓	-	84.54	35.84	22.16	82.84	34.97
	Stammbach and Neumann [91]**	✗	25.10	89.80	39.30	-	-	-
Joint models	Yin and Roth [117]*	✗	53.81	57.73	50.59	49.91	44.68	47.15
	Hidey and Diab [36]	✗	-	-	-	18.48	75.39	29.69
	Nie et al. [62]	✗	-	-	-	-	-	50.28

Table 4. Results of the sentence retrieval task in terms of the precision, recall, and F_1 evaluation metrics in the dev and the test set. The pre-calculated features column indicates whether a model uses external NLP tools or hand-crafted features. The best performing models per column are highlighted in bold font. The single star symbol (*) denotes that the results of a model are reported on the dev and test sets defined in Thorne et al. [99] (i.e., 9,999 dev and 9,999 test instances) and not on the dev and test sets of the shared task [101]. The double star symbol (**) indicates whether a model uses the title of the Wikipedia pages as external information. Missing results are not reported in the original papers. It is worthwhile mentioning that all models that do not belong in the category "Joint Models" (see the vertical heading in the first column) are pipeline models.

indicated in Table 3. Thus, all of these systems form a good baseline for the document retrieval step. It is worth mentioning that in all of the presented models, the *k* (i.e., the number of retrieved documents) used in each study is relatively small (i.e., 3-7) since otherwise the number of sentences for the next subtask (i.e., sentence retrieval) would be relatively large. A large number of selected sentences has the immediate effect of significantly slowing down the training and inference at the sentence retrieval subtask.

4.2 Sentence Retrieval

In Table 4, we present the results of the various sentence retrieval systems described in Section 3.1.2. The models in Table 4 are grouped similar to the way that are presented in the aforementioned subsection. The pre-calculated features column indicates whether the models use external NLP tools (e.g., named entity recognizers to detect mentions) or hand-crafted features. We present results both on the dev and the test sets using the precision, recall and F_1 evaluation metrics described in Section 2.5.2. Bold font indicates the best performing model per column. The single star symbol (*) denotes that the results of a model are reported on the dev and test sets defined in Thorne et al. [99] (i.e., 9,999 dev and 9,999 test instances) and not on the dev and test sets of the shared task [101] (see Table 1 for the statistics of the dataset of the shared task). The double star symbol (**) indicates whether a model uses the title of the Wikipedia pages as external information. Because this is a commonly used feature in this task that resolves co-reference issues, we do not include it in the pre-calculated features column.

In Table 4, we observe that the different models optimize over different evaluation metrics (e.g., precision, recall). For instance, the system of Luken et al. [54] optimizes over the precision evaluation

metric, the system of Nie et al. [64] optimizes over the F_1 score, while most works optimize over recall. This is because the recall metric measures the number of the correctly retrieved evidences over the total number of the ground truth evidences. This is of great importance since the core evaluation of the task (i.e., the FEVER score) requires at least one correctly retrieved evidence group along with the correct label for the claim in order to evaluate a claim as correct. Thus, retrieving more evidence groups maximizes the chance of retrieving a correct evidence group out of the retrieved evidence groups. However, the organizers have imposed the restriction of taking into account only the five highest scoring evidence groups. This restriction alleviates the issue of returning the full set of evidence groups that would lead to the problems of (i) a perfect recall and (ii) transforming the FEVER score to the label accuracy metric. Therefore, a high recall at the sentence retrieval subtask helps at increasing the performance of the model in terms of the FEVER score on the subtask of claim verification (next subtask in the pipeline).

Model	Dev			Test		
	P@5	R@5	F_1@5	P@5	R@5	F_1@5
Pointwise	27.66	95.91	42.94	23.77	85.07	37.15
Pairwise	27.29	94.37	42.34	25.21	87.47	39.14

Table 5. The performance of a BERT-based model trained on the sentence retrieval task using the pointwise and the pairwise loss functions on the dev and the test sets (see the work of Liu et al. [53]) in terms of Precision (P), Recall (R), and F_1 scores. The results are reported on the 5 highly ranked evidence sentences (i.e., @5). Note that in the original paper the results on the pointwise loss are not reported due to page limitations and the pointwise results are obtained from their Github codebase.

In Table 4 some of the results, especially in the dev set, are missing while for the test set, the results are available through the competition leaderboard. We observe that the ranking of the models (i.e., which model performs better compared to another model) in terms of their performance remains the same for the dev and the test set. However, for most of the models, the performance decreases in the test set. In terms of the recall, the ESIM-based and the language model based models perform better compared to the rest of the models. Exceptions are the models of Yoneda et al. [119] (i.e., the system which has been ranked as second in the shared task and relies on hand-crafted features, external tools and a logistic regression classifier) and the model of Stammbach and Neumann [91] (which is a combination of an ESIM-based and a BERT-based system). It is worth mentioning the experiment of Liu et al. [53] which indicates that using language models instead of ESIM-based for sentence retrieval leads to an improvement of 3 percentage points on the test set in terms of the recall evaluation metric and to 1 percentage point improvement on the claim verification subtask in terms of the FEVER score (this is not presented in Table 4). The two types of loss functions (i.e., pointwise and pairwise, see the language model based part in Section 3.1.2) that have been exploited for the sentence retrieval task, have been more extensively studied in the work of Soleimani et al. [89] and in the work of Liu et al. [53]. The experimental study of Soleimani et al. [89] suggests that there is a little variation in terms of recall between the pointwise and the pairwise models, even in the case that hard negative mining [82] is used in order to select more difficult instances. On the other hand, in the work of Liu et al. [53] (see Table 5), we observe that there is variation between the two losses on the dev and on the test set. Moreover, we observe that the pointwise loss performs better on the dev set while it performs worse on the test; this suggests that the pointwise loss overfits faster (while tuning the parameters) on the dev set. We hypothesize that this is because in the pairwise loss all the pairs of positive and negative examples are used while in the pointwise loss only a ratio of negative examples is used (i.e., five negative examples for each positive). The

ratio in the pointwise loss is used since otherwise we would have a highly imbalanced dataset, which can negatively affect performance. This has also been verified in our experiments, presented in Section 9, where the performance of the pointwise loss decreases by 1 percentage point in terms of the FEVER score both on the dev and the test set. The lack of a consistent conclusion from the two experimental studies (i.e., the one of Soleimani et al. [89] and the one of Liu et al. [53]) suggests that the impact of the loss function on the sentence retrieval task needs further investigation.

	Model	Pre-calculated Features	Dev Label Accuracy	Dev FEVER	Test Label Accuracy	Test FEVER
ESIM-based	Hanselowski et al. [32]	✗	68.49	64.74	65.46	61.58
	Yoneda et al. [119]**	✗	69.66	65.41	67.62	62.52
	Nie et al. [63]	✓	69.72	66.49	68.21	64.21
Language models	Nie et al. [64]	✗	75.12	70.18	72.56	67.26
	Chernyavskiy and Ilvovsky [19]**	✗	-	-	71.72	67.68
	Zhou et al. [122]	✗	74.84	70.69	71.60	67.10
	Portelli et al. [77]	✗	**84.33**	-	-	-
	Soleimani et at. [89]	✗	74.59	72.42	71.86	69.66
	Liu et al. [53]**	✗	78.29	76.11	74.07	70.38
	Zhong et al. [121]	✓	79.16	-	76.85	70.60
	Zhao et al. [120]**	✗	78.05	74.98	72.39	69.07
	Chen et al. [16]	✓	81.12	**78.94**	**76.91**	73.43
	Ye et al. [116]	✓	-	-	75.96	72.30
	Subramanian and Lee [92]	✗	75.77	73.44	74.64	71.48
	Stammbach and Ash [90]**	✗	77.48	74.98	76.60	**74.27**
Other models	Thorne et al. [99]	✓	51.37	31.27	48.84	27.45
	Chakrabarty et al. [14]	✗	58.77	50.83	57.45	49.06
	Luken et al. [54]	✓	44.70	43.90	50.12	43.42
	Malon [56]**	✗	-	58.44	61.08	57.36
	Taniguchi et al. [95]	✗	-	-	47.13	38.81
	Otto [69]	✗	-	-	54.15	40.77
	Tokala et al. [103]	✗	-	-	69.98	66.72
Joint models	Yin and Roth [117]*	✗	78.90	56.16	75.99	54.33
	Hidey and Diab [36]	✗	-	-	59.72	49.94
	Nie et al. [62]	✗	-	-	66.21	62.69
	Lee et al. [46]	✓	-	-	57.00	-
	Lewis et al. [47]	✗	74.50	-	72.50	-
	Hidey et al. [35]	✓	76.74	73.17	72.47	68.80

Table 6. Results of the claim verification task in terms of the label accuracy and the FEVER score evaluation metrics in the dev and the test set. The pre-calculated features column indicates whether a model uses external NLP tools or hand-crafted features. The best performing models per column are highlighted in bold font. The single star symbol (*) denotes that the results of a model are reported on the dev and test sets defined in Thorne et al. [99] (i.e., 9,999 dev and 9,999 test instances) and not on the dev and test sets of the shared task [101]. The double star symbol (**) indicates whether a model uses the title of the Wikipedia pages as external information. Missing results are not reported in the original papers. All models that do not belong in the category "Joint Models" (see the vertical heading in the first column) are pipeline models.

4.3 Claim Verification

In Table 6, the results of the claim verification subtask are presented in terms of the label accuracy and the FEVER score evaluation metrics. As in the above tables, the models are grouped based on the way that the groups have been formulated in Section 3.1.3. The pre-calculated features column indicates as before whether the models use external tools or hand-crafted features. Bold font indicates the best performing model per column. The single star symbol (*) denotes as above that the results of the model are reported on a different dev and test sets compared to the dev and test sets of the shared task [101]. The double star symbol (**) indicates whether the model uses the title of the Wikipedia pages as external information.

As we observe, the models have a better performance in the dev set compared to their performance on the test set. This is because the test set is blind and the number of submissions to Codalab is limited. Thus, the competition participants can only check the performance of their model in the test set by submitting the prediction file on the competition platform. On the other hand, the dev set is publicly available, and therefore, it is likely that some of the systems overfit on the dev set. Based on the results of Table 6, the systems that use language models have better performance both in terms of label accuracy and FEVER score in the dev and test sets compared to the rest of the models. This is because pre-trained language models have a superiority over the rest of the methods, since they have been trained on large corpora and thus they already incorporate prior knowledge. The ESIM-based models, which are the three highly-ranked models of the shared task, are the second best performing group of models in terms of both metrics, although there is a gap of 4-8 percentage points in terms of the label accuracy evaluation metric and 3-6 percentage points in terms of FEVER score. In addition, the joint model of Yin and Roth [117] performs well on the label accuracy (similar to the language model based approaches), however, the FEVER score drops dramatically due to the low recall of their model in the sentence retrieval task. Recall that in the task of sentence retrieval task in the work of Yin and Roth [117], they optimize over the F_1 score, which favors both precision and recall unlike most of the works that optimize only over recall (see Table 4). Note that the presented results of Yin and Roth [117] are reported on the splits defined in the work of Thorne et al. [99] and not on the splits of the shared task [101]. In general, joint models have shown an improved performance in a number of tasks (e.g., entity-relation extraction [9, 59], POS tagging-dependency parsing-chunking-semantic relatedness-textual entailment [34]) since the error propagation between the various sequential tasks is alleviated. However, this is not the case for the proposed joint architectures for the FEVER problem except for the model of Hidey et al. [35]. We hypothesize that this is due to the fact that in the FEVER problem, there are no annotated (i.e., gold) sentences for the NEI class and thus the different strategies of selecting examples (e.g., by randomly selecting sentences of the returned documents for that class) are not that beneficial. On the other hand, in the pipeline setting, a sentence retrieval model is trained on the sentences of the SUPPORTED and REFUTED classes and this model can later on be used to retrieve sentences that are exploited as potential evidence sentences along with the corresponding claim to train the model on the NEI class for the claim verification subtask.

Based on the results of Table 6, the model proposed in the work of Stammbach and Ash [90] performs best in terms of the FEVER score in the test set. This is because this model relies on multiple combined modules on the downstream components (i.e., document and sentence retrieval), but its main benefit comes from the fact that it relies on the GPT-3 pre-trained language model [13]. The models presented in the work of Zhong et al. [121] and Chen et al. [16] lead to the best performance in terms of label accuracy both in the dev and test sets. This is due to the fact that these models rely on external tools. The first one uses the semantic role labeling tool of AllenNLP[5] for constructing the graph of the claim and the evidence sentences. That way the graph neural network used in this work is able to take into account the structure of the semantic roles (due to the use of the external tool) instead of extracting that information from the raw claim and evidence sentences during training. The work of Chen et al. [16] uses again AllenNLP tools to extract the central phrases in a claim. The contribution of the semantic roles and central phrases is also evident from that other works that rely on graph neural networks (see Liu et al. [53], Zhou et al. [122]) achieve lower performance in terms of label accuracy and FEVER score. The gap between the model of Liu et al. [53] and one that uses semantic roles [121] on the test set is smaller in terms of FEVER score; we hypothesize that this is because the former model is able to extract more relevant sentences (i.e.,

---

higher recall) at the sentence retrieval subtask. Moreover, this is also evident from that the model of Chen et al. [16] performs similar to the model of Zhong et al. [121] in terms of label accuracy on the test set, while it performs 3% better in terms of FEVER score. We believe that this is because the model of Chen et al. [16] relies on the sentence retrieval of Liu et al. [53]. We should note that the use of external tools (e.g., semantic role labeling) is beneficial in a number of tasks (e.g., entity and relation extraction where a dependency parser has been exploited to improve the relation extraction task, see Miwa and Bansal [59]). However as presented in the work of Bekoulis et al. [9, 10], the performance of a model can be significantly reduced when the external tool (e.g., a parser) has been trained on data coming from different domains (e.g., news data) or languages (e.g., English) and is applied on data from another domain (e.g., biological data) or language (e.g., Dutch). In addition, the models of Liu et al. [53], Soleimani et al. [89], Zhao et al. [120], Zhong et al. [121], which optimize for high recall in the sentence retrieval task, obtain almost similar performance on the FEVER score on the test set (i.e., a variation of 1 percentage point). Moreover, the models of Liu et al. [53], Zhao et al. [120], Zhong et al. [121] are far more complex compared to the plain BERT-based model of Soleimani et al. [89]; thus, it is clear that the main benefit (compared to prior studies, e.g., Hanselowski et al. [32]) comes from the pre-trained models.

5 RELATED TASKS

In this section, we present tasks that are strongly related to automated fact checking. Note that this is a non extensive list of related tasks; however, this list serves as indication of problems related to automated fact checking that can be tackled by similar methodologies.

Adversarial Examples for FEVER: The second version of the FEVER shared task has also been organized [102]. The goal was to generate adversarial examples able to improve the performance of the FEVER task. Specifically, there were three phases, namely (i) Build-it, where the goal was to develop a system for solving the FEVER task, (ii) Break-it, where the goal was to generate adversarial instances to fool the Builder system, and (iii) Fix-it, where the goal was to combine the Builder system with the generated adversarial instances of the Breaker system in order to improve the performance of the model. It is worth mentioning that the authors of the shared task have also introduced metrics for evaluating the quality of the generated adversarial instances. Niewinski et al. [65] submitted the winning solution for the Breakers system and they used a language model based architecture along with a targeted vocabulary for generating adversarial examples.

Fake News Detection: Fake news detection is strongly related to automated fact checking and some of the previous works consider automated fact checking as a constituent of fake news, see, for example, this review paper on fake news detection [67]. Fake news detection refers to the task of assessing the validity of full articles, claims or social media posts. Thus, the type of the input (e.g., full article) depends on the dataset and the downstream application. It is common in applications that assess the validity of fake news (e.g., posts, articles) to use fake/non-fake as prediction labels [72]. However, in other applications, there are more fine-grained labeling schemes, since an article can be partially fake (see e.g., the work of Mitra and Gilbert [58]). For solving the fake news detection problem several approaches have been proposed ranging from feature-based models [22] to neural network architectures [61, 80]. Note that there are several surveys that have been proposed for related tasks to fact verification and specifically for fake news detection [11, 81, 87, 123].

Rumour Detection: According to the work of Zubiaga et al. [124], a rumour is a statement, which has not been officially verified at the time that the statement is posted. Specifically, in this task, binary classifiers are usually used to predict whether a statement is a rumour or not. For instance,

one can see the PHEME dataset [125], where the input are threads of tweets from Twitter users and the goal is to classify them as true or false (i.e., rumour, non-rumour). The use of the interactions among the users' profiles has also been exploited in the literature (see Do et al. [28]) to improve the model performance on the rumour detection task.

Stance Detection: Stance detection is a closely related problem to fact checking and according to the study of Zubiaga et al. [124], stance detection could be used after the rumour detection module in rumour classification systems. Specifically, the rumour detector is responsible to identify whether a statement is rumour or not, while the stance detector is responsible for identifying the stance of the author of a text against a given statement (e.g., FAVOR, AGAINST) that has been classified as rumour [45].

6 RELATED DATASETS

Similar to the FEVER dataset, several other datasets have been proposed for verifying the truthfulness of claims in several contexts.

SCIFACT: Recently, a dataset that follows the paradigm of the FEVER dataset in the context of validating scientific claims has been proposed by Wadden et al. [107]. Specifically, in that work, Wadden et al. [107] constructed a set of 1,409 scientific claims and the veracity of each claim was evaluated against a corpus of 5,183 scientific abstracts. The baseline model proposed for solving this task includes a pipeline of three subtasks similar to FEVER, namely, abstract retrieval (similar to document retrieval), rationale selection (similar to sentence selection) and label prediction (similar to claim verification).

LIAR: The LIAR dataset has been introduced in the work of Wang [110] for fake news detection. In particular, the author of this work created a dataset consisting of 12.8k statements manually annotated from the POLITIFACT.COM website. Unlike the FEVER dataset, which has been constructed using only Wikipedia documents, the problem described in the LIAR dataset is closely related to fake news detection since the dataset is constructed using only news content (e.g., tweets, interviews, Facebook posts). An instance consists of a statement, the person who made the statement, and the context (e.g., president elections). The label of the instance falls into one of the six predefined fine-grained classes (false, barely-true, etc.). The baseline model proposed in that work consists of a combined architecture of CNNs and bidirectional LSTMs to predict the label given the statement and the metadata.

LIAR-PLUS: A variation of the original LIAR dataset is the dataset introduced in the work of Alhindi et al. [3]. Unlike the LIAR dataset, the LIAR-PLUS dataset extracts automatically the justification for each statement. To do so, the authors of the LIAR-PLUS select the summary section of the articles or last sentences (if there is no summary section) as justifications. By exploiting this information the model performance increases for all the examined architectures.

TABFACT: All the datasets presented so far rely on extracting information from raw text. However, none of these studies consider structured or semi-structured information. Chen et al. [18] introduced a dataset called TABFACT, which includes 117,854 manually labeled claims based on 16,753 tables from Wikipedia. The goal of the task is to predict the veracity of the claim and two labels have introduced for that (i.e., ENTAILED, REFUTED). The challenge in this dataset is that it is not straightforward to extract information from a Wikipedia table. The authors of this work have

proposed a Transformer- and a BERT-based solution to solve the problem.

FTFY: The FTFY dataset proposed in the work of Hidey and McKeown [37] contains contrastive claims of Reddit posts. Specifically, the authors of this work crawled posts from Reddit that received "Fixed That For You" (FTFY) responses. These responses are edits of the original post where the person who responds modifies part of the original comment. They propose a methodology for automatically generating pairs of contrastive claims using a sequence-to-sequence model, which are not trivially contrastive (i.e., not negation of the original claim). Recall that this was also an issue that the authors of the FEVER dataset tried to alleviate, as described in Section 2.2.

MultiFC: Unlike the FEVER dataset, which relies on claims generated from Wikipedia, the MultiFC dataset proposed by Augenstein et al. [7] relies on 26 fact checking websites. In particular, they constructed a dataset of 34,918 claims. The claims were crawled from various domains, where each domain has a different number of labels; this is also the challenging aspect of this dataset. In this work, they also propose a multi-task learning approach, which takes into account the relation between the labels of the different domains.

FakeCovid: A recently constructed dataset inspired by the COVID-19 pandemic is the Fake-Covid [86]. This dataset includes 5,182 news articles related to COVID-19, coming from 92 fact checking websites in 40 languages. Unlike the aforementioned datasets, this one exploits the use of multilingual sources. The authors also provided a BERT-based classification model for solving the task.

Others: Several other datasets have been proposed in the context of fact checking and the related to that tasks, such as fake news detection, rumour detection, etc. A non extensive list of papers includes PHEME [125], SOME-LIKE-IT-HOAX [94], SYMMETRIC FEVER [83], CLEF-2019 tasks 1 & 2 [4, 33] and CLIMATE-FEVER [27]. Additional datasets have introduced in the works of Derczynski et al. [24], Pérez-Rosas et al. [72], Popat et al. [76], Shu et al. [88]. Note also that the FEVER dataset has been included in the ERASER [26] and KILT [74] benchmarks.

7   COMPETITIONS

In this section, we list some of the most recent competitions related to the FEVER task. In particular, along with the FEVER competition, several other competitions have been organized, which are related to fact checking applications (e.g., fake news, fact checking based on table data (see TABFACT in Section 6)). Specifically, in the context of FEVER, a second shared task (**FEVER 2.0**) [102] has been organized and the goal is to define Builders (systems that solve the first task), Breakers (generate adversarial instances to break prior methods that solve the FEVER problem) and Fixers (improving systems by exploiting adversarial instances), as also described in Section 5. Another competition has been organized for the newly developed dataset **SCIFACT** [107][6]. In this competition, the goal is to identify the validity of a claim based on scientific abstracts. **CLEF 2020 CheckThat!** is a competition that runs the last 3 years (i.e., since 2018) and, in its last edition, the goal for Task 1 was to rank claims (of a political debate) based on whether these claims are interesting to be annotated and for Task 2 was to rank the evidence and check the veracity of the claims (in an Arabic dataset). A competition has also been organized[7] [18] based on the **TABFACT** dataset (described in Section 6), where the goal is to verify the validity of a claim based on Wikipedia tables. In the

---

[6]https://scifact.apps.allenai.org/leaderboard
[7]https://competitions.codalab.org/competitions/21611

same line, the shared task SemEval 2021 (Task 9) **SEM-TAB-FACT**[8] aims at identifying whether a table supports a given claim and providing also the required evidence for that. The **Fake News Challenge**[9] has been organized in 2017 and attracted a lot of attention (i.e., 50 participants). The goal was to identify the stance (i.e., agree, discuss, disagree, be unrelated) of a specific article with respect to a headline.

8 EXPLAINABLE FACT CHECKING MODELS

Although explainable models is an area of increasing interest in NLP [49, 108], there are only few studies that focus on explaining the outcome of fact checking models. Specifically, Atanasova et al. [5] exploit the use of a joint architecture that simultaneously generates explanations using the extractive summarization model of Liu and Lapata [50] and predicts the veracity of a claim using a classification layer. Similar to the previous work, the work of Stammbach and Ash [90] relies also into abstractive summarization techniques for generating explanations. Specifically, this work, which relies on GPT-3 [13], generates explanations based on the FEVER dataset. The model generates a summary based on the context (i.e., evidences) and the claim. This summary has also been used as an input to the FEVER system (instead of the evidence sentences) and the system performed well on the dev set. This indicates that the summarized context (i.e., evidences) is of high quality. Along the same line, Glockner et al. [31] proposed a new framework that is able to automatically extract rationales. In particular, they also exploit the rationales to solve the task at hand (i.e., they also demonstrate the effectiveness of their method in the FEVER dataset). The rationales that lead to best performance obtain a high ranking (i.e., score). In another work, Ahmadi et al. [2] propose the use of a set of rules on knowledge graphs (known for their structured representation of information, i.e., entities and their corresponding relations) to extract interpretable results. Unlike the aforementioned studies that propose new methods for generating explanations, in the benchmark study of DeYoung et al. [26], they define different metrics for evaluating the quality of alignment between human and machine generated rationales. Nadeem et al. [60] developed an end-to-end system for fact checking that is able to provide explanations based on the stance scores. Specifically, the sentences with the highest stance scores are highlighted in the user interface and provided as the explanations of the model. The work of Paranjape et al. [70] has identified that there should be an equilibrium between the level of explanation conciseness and the task accuracy. Experimental results indicate that retaining a trade-off between these two tasks leads to performance improvement on a standard benchmark (ERASER [26]), as well as agreement with human generated explanations.

9 EXPERIMENTAL STUDY

In this section, we conduct an experimental study on the FEVER dataset. Specifically, inspired by Table 5 presented in Section 4.2, we perform an analysis on the various loss functions for the sentence retrieval subtask. This is because based on the results presented in Table 5, a concrete conclusion is missing about the type of loss function that is appropriate for sentence retrieval. As we observe in Table 5, the pointwise loss function performs better on the dev set, while the pairwise loss function performs better on the test set and the rationale for this is explained at the end of Section 4.2. Another motivation for this experimental study is that the performance of the loss functions used in the sentence retrieval task of the FEVER dataset is not a well-studied problem. Although, as analyzed in Section 3.1.2, several methods have been proposed for the sentence retrieval task, none of these studies focuses on experimenting with different ranking

---

[8]https://sites.google.com/view/sem-tab-facts/
[9]http://www.fakenewschallenge.org/

criteria. To investigate further the benefit of the examined loss functions, we conduct experiments both on the sentence retrieval and claim verification subtasks.

9.1 Methods

At this point, we present the various loss functions that we have investigated for the sentence retrieval task. We have exploited loss functions commonly used in computer vision (e.g., angular loss) and for document ranking. For representing the input sentences, we use a BERT-based model similar to previous work (see e.g., Liu et al. [53]).

Pointwise: In this setting, we use the cross-entropy loss and the input to our model is the claim along with an evidence sentence. The goal of the sentence retrieval component paired with the pointwise loss is to predict whether a candidate evidence sentence is an evidence or not for a given claim. Thus, the problem of sentence retrieval is framed as a binary classification task.

Pairwise: In our work, we also exploit the pairwise loss, where the goal is to maximize the margin between the positive and the negative examples. Specifically, we use the pairwise loss presented in Eq. (1) which is similar to the margin based loss presented in the work of Wu et al. [112]. The pairwise loss is:

$$\mathcal{L}^{pairwise}(p, n) = [-y_{ij}(f(x_p) - f(x_n)) + m]_+, \tag{1}$$

where $y_{ij} \in \{-1, 1\}$, $f(x)$ is the representation that we obtain from the BERT-based model, m is the margin and the indices p and n indicate a pair of a positive and a negative example. In order to obtain a claim aware representation of the (positive-negative) instances, we concatenate the claim with the corresponding evidence.

Triplet: Unlike the pairwise loss that considers only pairs of positive and negative examples, the triplet loss [112] uses triplets of training instances. Specifically, given an anchor sample a, the goal is the distance $D_{ij} = \|f(x_i) - f(x_j)\|_2$ to be greater between the anchor and a negative example than the distance between the anchor and a positive example. The triplet loss is given by

$$\mathcal{L}^{triplet}(a, p, n) = [D_{ap}^2 - D_{an}^2 + m]_+, \tag{2}$$

where m is the margin and the indices a, p and n indicate the triplet of the anchor, a positive and a negative example. As anchor we use the claim, and similar to the pairwise loss, we concatenate the claim with the corresponding evidence for the positive and the negative examples.

Cosine: We have also experimented with the cosine loss. Specifically, we exploit positive and negative samples by using the following formula:

$$\mathcal{L}^{cos}(p, n) = y_{ij}(1 - \cos(f(x_p), f(x_n))) + (1 - y_{ij})[(\cos(f(x_p), f(x_n)) - m)]_+, \tag{3}$$

where $y_{ij} \in \{0, 1\}$ and cos indicates the cosine distance between the positive and the negative samples.

Angular: The angular loss [109] uses triplets of instances (i.e., similar to the triplet loss) while imposing angular constraints between the examples of the triplet. Specifically, the formula is given by

$$\mathcal{L}^{ang}(a, p, n) = [D_{ap}^2 - 4\tan^2 r D_{nc}^2]_+, \tag{4}$$

where $f(x_c) = (f(x_a) - f(x_p))/2$ and r is a fixed margin (angle).

Evidence-Aware: Unlike all the aforementioned loss functions, we propose an evidence-aware model that relies on transformers, similar to the retrieval model proposed in the work of Pobrotyn et al. [75]. This model exploits the use of self-attention over the potential evidence sentences. Unlike (i) the pointwise loss that does not take into account the relations between the evidence sentences and (ii) the distance-based losses (e.g., pairwise, triplet) that considers only pairs of sentences, the evidence-aware model considers subsets of evidence sentences simultaneously at the training phase. Specifically, the input to the evidence-aware model is a list of BERT-based representations of the evidence sentences. Thus, the model is able to reason and rank the evidence sentences while taking into account all the other evidence sentences in the list. On top of the evidence-aware model, we exploit a binary cross-entropy loss similar to the one presented in the case of the pointwise loss, framing again the problem as a binary classification task.

9.2 Setup

We evaluate our models on the FEVER dataset [99] presented in Table 1. We use the BERT-based model [25] in all of our experiments to guarantee a fair comparison among the various loss functions. Our code will be available upon acceptance of the manuscript. The input to our sentence retrieval component is the output of the document retrieval task presented in the work of Hanselowski et al. [32] and used also in the work of Liu et al. [53]. For the conducted experiments in the sentence retrieval task, in all the loss functions except for the evidence-aware one, we present results using all the potential evidence sentences as retrieved from the document retrieval step. For the evidence-aware model, we conduct experiments using either 5 or 10 negative examples per positive instance during training. In addition, the overall (positive and negative) maximum number of instances that are kept is 20. This is because, unlike the other models where the evidences are considered individually or in pairs, in the evidence-aware model, we have to limit the instances that are considered simultaneously in the list. We experiment also with a limited number of instances in the other settings to have a fair comparison among the different setups. For the distance-based losses (e.g., triplet, pairwise), we conduct additional experiments only in the best performing model when all instances are included (i.e., the pairwise loss). We also present results on the claim verification task with all of the examined architectures. For the claim verification step, we use the model of Liu et al. [53]. We evaluate the performance of our models using the official evaluation metrics for sentence retrieval (precision, recall and F_1 using the 5 highly ranked evidence sentences) and claim verification (label accuracy and FEVER score) in the dev and test sets.

9.3 Results

Table 7 presents our results on the sentence retrieval and claim verification tasks using the various examined loss functions. The "# Negative Examples" column indicates the number of negative evidences that are randomly sampled for each positive instance during training and the "# Max Instances" column indicates the maximum number of instances that we keep for each claim. The ✓ symbol denotes that we keep all the instances from this category (i.e., "# Negative Examples" or "# Max Instances"). Regarding the number of maximum instances, we keep as many as possible from the positive samples (i.e., if we have 5 positive samples and the number of maximum instances is 20 we keep all of them, while if the number of positive samples is 25 we keep 20 of them), and then we randomly sample from the negative instances. The settings (✓, ✓) for the pairwise loss and settings (5, ✓) for the pointwise loss reported in Table 7, are similar to the ones reported in Table 5.

The evidence-aware model (see the setting with 5 negative examples and 20 maximum instances denoted as (5, 20)) is the best performing one both in the dev and test set in terms of FEVER score. The pairwise loss performs best in terms of label accuracy on the test set. However, the most important evaluation metric is the FEVER score, since it takes into account both the label accuracy

Loss	# Negative Examples	# Max Instances	Dev					Test				
			P@5	R@5	F_1@5	LA	FEVER	P@5	R@5	F_1@5	LA	FEVER
Angular	✓	✓	26.90	93.93	41.82	77.22	74.81	24.36	86.14	37.98	72.30	68.30
Cosine	✓	✓	27.02	93.85	41.96	77.50	75.10	**24.83**	86.73	**38.61**	72.49	68.81
Triplet	✓	✓	26.99	94.24	41.96	77.51	75.32	24.74	**86.86**	38.51	72.76	69.31
Pairwise	✓	✓	26.88	93.90	41.79	78.05	75.61	24.44	86.17	38.08	72.92	69.34
	5	✓	26.76	93.23	41.58	77.21	74.74	24.53	85.90	38.17	72.05	68.22
	10	✓	26.77	92.99	41.57	77.58	75.04	24.62	86.15	38.29	72.65	68.93
	5	20	27.11	94.13	42.10	77.53	75.37	24.75	86.67	38.51	72.87	69.25
	10	20	27.09	94.40	42.10	78.05	75.79	24.74	86.84	38.51	**73.02**	69.38
Pointwise	✓	✓	25.77	91.96	40.26	77.94	75.12	22.28	82.61	35.01	71.63	67.63
	5	✓	27.74	95.93	43.04	78.43	76.71	23.99	85.67	37.48	72.54	68.71
	5	20	27.39	95.25	42.54	78.49	76.58	23.79	85.24	37.19	72.55	68.64
Evidence-Aware	5	20	**28.52**	**97.16**	**44.09**	**78.67**	**77.38**	24.70	86.81	38.46	72.93	**69.40**
	10	20	28.50	96.82	44.04	78.26	76.78	24.76	86.83	38.53	72.70	68.46

Table 7. Results of the (i) sentence retrieval task in terms of Precision (P), Recall (R), and F_1 scores and (ii) claim verification task in terms of the label accuracy (LA) and the FEVER score evaluation metrics in the dev and the test sets. The "# Negative Examples" column indicates the number of negative evidences that are randomly sampled for each positive instance (i.e., evidence) in the training phase. The "# Max Instances" column indicates the maximum number of instances that we keep for each claim (e.g., evidences in the case of the pointwise loss and positive-negative pairs in the case of the pairwise loss) in the training phase. The ✓ symbol denotes that we use all the examples of the category (i.e., "# Negative Examples" or "# Max Instances"). The best performing models per column are highlighted in bold font. Note that the sentence retrieval metrics for both the dev and the test set are reported on the 5 highly ranked evidence sentences (i.e., @5).

and the predicted evidence sentences. The pointwise loss is the worst performing one when using all the evidence sentences. This is because when we use all the potential evidences, the number of negative samples is too large and we have a highly imbalance problem leading to low recall and FEVER score in both the dev and test set. Note that the evidence-aware model relies on the pointwise loss (i.e., the worst performing one). However, a benefit of the evidence-aware model (0.7% in terms of FEVER score) is reported (see pointwise (5, 20)). This showcases the important effect of ranking potential evidences simultaneously using self-attention. From the distance-based loss functions (e.g., triplet) except for the pairwise, we observe that the angular and the cosine loss have worse performance compared to the pairwise and the triplet loss when using all the instances. We hypothesize that this is because the norm-based distance measures fit best for scoring pairs using the BERT-based representations.

The evidence-aware model is the best performing one (5, 20), while using only a small fraction of the overall training instances. This is because the evidence-aware model is able to take into account all possible combinations of the sampled evidences while computing attention weights. However, the same model in the (10, 20) setting showcases a reduced performance. This is due to the fact that the pointwise loss influences the model in a similar way as in the pointwise setting, leading to a lower performance (due to class imbalance). For the pairwise loss, we observe that the performance of the model when sampling constrained evidence sentences (see (5, 20), (10, 20) settings) is similar to the performance of the model when we do not sample evidence sentences. In addition, it seems that when one constrains the number of negative samples should also constrain the overall number of instances in order to achieve the same performance as in the non-sampling setting. We hypothesize that this is due to the fact that when we have a limited number of instances, it is better to have

a more balanced version of the dataset. Therefore, we conclude that the evidence-aware model achieves high performance by using few examples, and it can be therefore used even in the case that we have a small amount of training instances. In the case of the pairwise loss, it is important to sample instances, otherwise it becomes computationally intensive when we take all the possible combinations between the positive and negative training instances into account. Finally, note that it is crucial to sample negative sentences to control: (i) the computational complexity in the case of the distance-based loss functions, (ii) the memory constraints in the case of the evidence-aware model and (iii) the imbalance issue in the case of the pointwise loss.

10 CHALLENGES AND FUTURE RESEARCH DIRECTIONS

Bias in the FEVER dataset: As indicated in the work of Schuster et al. [83], there is a bias issue in the FEVER dataset that may affect the performance of the fact checking systems. Specifically, they observed that by only using the claim statement without the evidence sentences, the performance of their BERT-based system was slightly worse (8 percentage points) compared to that of an NSNM system (see Section 3), which uses also the predicted evidence sentences as input. Although, these 8 percentage points difference between the two systems might be seen for some cases as a substantial improvement, this result indicates that an important part of the input is neglected. To alleviate this issue, Schuster et al. [83] created a new test dataset, the so-called "SYMMETRIC TEST SET". In this dataset, for each claim-evidence pair, they create a claim-evidence pair that contradicts the original one. By taking all the possible combinations (of claims and evidences), the authors of Schuster et al. [83] were able to generate four claim-evidence pairs. In the same work, new regularization methods to improve the performance on the new test set were introduced; for more details, we refer to [83]. In the same line of research, Thorne and Vlachos [98] also rely on regularization methods to improve the performance on the SYMMETRIC dataset. In another work, Karimi Mahabadi et al. [43] proposed the use of product of experts [38] and focal loss [48] in order to mitigate the bias in NLI systems and reported results on the FEVER SYMMETRIC test set. Finally, the quality of the dataset is also discussed in the work of Derczynski et al. [23], which proposes two metrics that ensure the quality of the annotation of the FEVER dataset. In the work of Shah et al. [85], augmentation techniques have been proposed in order to expand the FEVER training set using sentence-based modification approaches. The method has also been validated on the "SYMMETRIC TEST SET". Similar to Schuster et al. [83], the work of Pratapa et al. [78] also illustrates that LMs are able to conclude about the veracity of a claim without taking the evidences into account (i.e., based on the knowledge these models have while training on large corpora). This is why they construct a new anonymized dataset, where the entities are masked out and the evidences are the only facts that are available to the model.

Adversarial instances: Generating adversarial instances has been proposed in order to identify and improve the vulnerabilities of the various systems. As also described in Section 5, a shared-task has been organized regarding that matter [102] and a baseline system along with scoring metrics for evaluating adversarial attacks can be found in Thorne and Vlachos [97], Thorne et al. [100]. Several works have been proposed for introducing adversarial attacks in various datasets (see e.g., Atanasova et al. [6], Hidey et al. [35]), showcasing that generating adversarial instances is an interesting research direction that can help robustifying the fact checking systems.

Artificially generated dataset: Another issue with the FEVER dataset is that the claims are artificially constructed by Wikipedia content, a source which the documents are also coming from. This way, we have a controlled environment that is far from realistic fact checking scenarios. For instance, in a realistic situation, we might assess the validity of claims coming from social media (e.g., Twitter) or news sources. Therefore, for practitioners that are interested to build systems for

fact checking applications (such as fake news), it is not advisable to test their system only on the FEVER dataset. Although the dataset is artificially constructed, it can have an advantage in some cases in transfer learning scenarios. For instance, the FEVER dataset has been exploited in [107], where a model pre-trained on FEVER and finetuned on the SCIFACT dataset was shown to deliver performance improvements in terms of label accuracy. However, there was no improvement in the case of sentence selection since the pre-trained FEVER model has not been trained on a scientific corpus. Another example regarding the poor generalization performance of the FEVER dataset in transfer learning scenarios can be found in the work of Suntwal et al. [93].

Multilinguality: The FEVER dataset is available only in the English language. However, in case one is interested in identifying the veracity of a claim might need to consider multi-lingual content [66]. This is, for instance, applicable in countries with more than one official languages. Thus, a multi-ligual dataset that can be used in the aforementioned scenario would be useful.

Single-hop nature of the dataset: It is worth mentioning once more (see also Section 2.1) that in 16.82% of the FEVER claims, more than one evidence sentences are needed to conclude about their veracity. This indicates that models explicitly designed to solve more complex tasks, where multiple evidence sentences are needed to conclude about the veracity of the claim, have limited improvement compared to their full potential. For instance, see the model of Zhao et al. [120] in which the overall performance is similar to the rest of the state-of-the-art models. However, in a more difficult subset the improvement of that model is almost 20% in terms of FEVER score. Recently, fact checking datasets have been proposed that require more hops (sentences or documents) to validate the veracity of a claim, see for instance Jiang et al. [41], Ostrowski et al. [68].

Too many negative instances: Based on our analysis on the sentence retrieval task (see our analysis in Section 9.2), we observed that there are a lot of negative sentences that are coming from the document retrieval step. This leads to issues, such as an imbalance dataset (in the pointwise loss) and a high computational complexity (in the pairwise loss) when using all the negative samples. To circumvent these issues, following also previous research [53, 89], we randomly sample negative examples leading to lower computational time and improved performance. However, more sophisticated techniques than random sampling should be investigated to select examples that are more informative. So far, hard negative mining to select more difficult examples has been studied in the work of Soleimani et al. [89], although the benefit of that technique is limited.

Complex vs simple models: For the FEVER task, we observe that most of the recent research works (e.g., Liu et al. [53]) focus on creating new complex architectures for the claim verification task. We conducted a small scale experiment (that is not present in Table 7), where we replaced our model for claim verification (recall that we rely on the method of Liu et al. [53]) with a BERT-based classifier. We observed that the benefit for the pointwise loss was 0.2%, a benefit of 0.1% for the triplet loss and a drop of 1% in the performance of the cosine loss when using the model of Liu et al. [53] instead of the BERT-classifier (in our early experiments on the dev set). This indicates that models relying on complex architectures without any extra knowledge have a limited benefit in the performance of the FEVER score. Note that this gap can become larger in other datasets (see e.g., the performance of the model of Liu et al. [52, 53] on SCIFACT [107]). On the other hand, the use of pre-trained BERT in the FEVER task [89] gave a significant boost over the ESIM models. Moreover, the use of semantic role labeling [121] (which is an external source of knowledge) also improved the performance on the task, leading to performance similar to that of the $BERT_{large}$ model of Liu et al. [53]. Finally, the combination of various models along with the use of the GPT-3 pre-trained

language model [13] (i.e., the main benefit is coming from that model) in the work of Stammbach and Ash [90] leads to a substantial improvement. Our experimental evidence shows that starting from a simple architecture and using one of the best performing setups in Table 7 together with a BERT-based classifier for the claim verification task can deliver fair performance in the FEVER dataset. However, as noted above this simple classification architecture might not perform as well as more complex architectures for every dataset. It is also evident that an interesting direction is the use of models that bring additional knowledge (e.g., new pre-trained models such as GPT-3 [13] or by identifying other beneficial tasks such as coreference resolution [116]), combinations of existing models or models that are able to perform complex reasoning operations in order to extract different type of knowledge from the data. Finally, as indicated by our performance gain, we motivate future researchers to work also on the sentence retrieval subtask, as the improvement in this subtask leads to similar improvements with architectures proposed for the claim verification subtask.

Scalability: Computational efficiency is important for fact-checking systems when used in real life situations such as verification of claims or fake news detection. Such systems should be able to process a huge volume of data and be able to respond fast and accurate to the queries of the users. Examples of such systems include architectures that scale well with data or the number of users, low complexity architectures, distributed training and distributed inference architectures [1].

Interpretability/Explainability: It is also crucial for fact checking systems to be interpretable (i.e., design models that their predictions can be interpreted by e.g., attention or visualising filters) and explainable (i.e., design systems for post-hoc analysis of the predictions of the model). Although there are explainable models in the NLP community, there is little work regarding interpretability/explainability for fact checking (as one can observe in Section 8) and the work for explainable models for the FEVER dataset is rather limited. Another direction is to study approaches for assessment of interpretability and explainability in the context of fact checking.

11 CONCLUSION

In this paper, we focused on the FEVER task where the goal is to identify whether a sentence is supported or refuted by evidence sentences or if there is not enough info available, relying solely on Wikipedia documents. The aim of our work has been to summarize the research that has been done so far on the FEVER task, analyze the different approaches, compare the pros and cons of the proposed architectures and discuss the results in a comprehensive way. We also conducted a large experimental study on the sentence retrieval subtask and drew diverse conclusions useful for future research. We envision that this study will shed some light on the way that the various methods are approaching the problem, identify some potential issues with existing research and be a structured guide for new researchers to the field.

ACKNOWLEDGMENTS

REFERENCES

[1] Lusine Abrahamyan, Yiming Chen, Giannis Bekoulis, and Nikos Deligiannis. 2021. Learned Gradient Compression for Distributed Deep Learning. *arXiv preprint arXiv:2103.08870* (2021).

[2] Naser Ahmadi, Joohyung Lee, Paolo Papotti, and Mohammed Saeed. 2019. Explainable Fact Checking with Probabilistic Answer Set Programming. In *Conference on Truth and Trust Online*. London, UK, 11.

[3] Tariq Alhindi, Savvas Petridis, and Smaranda Muresan. 2018. Where is Your Evidence: Improving Fact-checking by Justification Modeling. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*. Association for Computational Linguistics, Brussels, Belgium, 85–90. https://doi.org/10.18653/v1/W18-5513

[4] Pepa Atanasova, Preslav Nakov, Georgi Karadzhov, Mitra Mohtarami, and Giovanni Da San Martino. 2019. Overview of the CLEF-2019 CheckThat! Lab: Automatic Identification and Verification of Claims. Task 1: Check-Worthiness.. In *Proceedings of CLEF (Working Notes)*. CEUR Workshop Proceedings, Lugano, Switzerland, 15.

[5] Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020. Generating Fact Checking Explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 7352–7364. https://doi.org/10.18653/v1/2020.acl-main.656

[6] Pepa Atanasova, Dustin Wright, and Isabelle Augenstein. 2020. Generating Label Cohesive and Well-Formed Adversarial Claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 3168–3177. https://doi.org/10.18653/v1/2020.emnlp-main.256

[7] Isabelle Augenstein, Christina Lioma, Dongsheng Wang, Lucas Chaves Lima, Casper Hansen, Christian Hansen, and Jakob Grue Simonsen. 2019. 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, Hong Kong, China, 4685–4697. https://doi.org/10.18653/v1/D19-1475

[8] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference for Learning Representations*. San Diego, USA, 15.

[9] Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018. Adversarial training for multi-context joint entity and relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 2830–2836. https://doi.org/10.18653/v1/D18-1307

[10] Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018. Joint entity recognition and relation extraction as a multi-head selection problem. *Expert Systems with Applications* 114 (2018), 34–45.

[11] Alessandro Bondielli and Francesco Marcelloni. 2019. A survey on fake news and rumour detection techniques. *Information Sciences* 497 (2019), 38–55. https://doi.org/10.1016/j.ins.2019.05.035

[12] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, 632–642. https://doi.org/10.18653/v1/D15-1075

[13] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 1877–1901. https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[14] Tuhin Chakrabarty, Tariq Alhindi, and Smaranda Muresan. 2018. Robust Document Retrieval and Individual Evidence Modeling for Fact Extraction and Verification.. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*. Association for Computational Linguistics, Brussels, Belgium, 127–131. https://doi.org/10.18653/v1/W18-5521

[15] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to Answer Open-Domain Questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, 1870–1879. https://doi.org/10.18653/v1/P17-1171

[16] Jiangjie Chen, Qiaoben Bao, Jiaze Chen, Changzhi Sun, Hao Zhou, Yanghua Xiao, and Lei Li. 2020. LOREN: Logic Enhanced Neural Reasoning for Fact Verification. *arXiv preprint arXiv:2012.13577* (2020).

[17] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for Natural Language Inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, 1657–1668. https://doi.org/10.18653/v1/P17-1152

[18] Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2019. TabFact: A Large-scale Dataset for Table-based Fact Verification. In *International Conference on Learning Representations*. New Orleans, USA, 25.

[19] Anton Chernyavskiy and Dmitry Ilvovsky. 2019. Extract and Aggregate: A Novel Domain-Independent Approach to Factual Data Verification. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*. Association for Computational Linguistics, Hong Kong, China, 69–78. https://doi.org/10.18653/v1/D19-6612

[20] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *Proceedings of the 2017 Conference on*

Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Copenhagen, Denmark, 670–680. https://doi.org/10.18653/v1/D17-1070

[21] Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating Cross-lingual Sentence Representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 2475–2485. https://doi.org/10.18653/v1/D18-1269

[22] Nadia K Conroy, Victoria L Rubin, and Yimin Chen. 2015. Automatic deception detection: Methods for finding fake news. *Proceedings of the Association for Information Science and Technology* 52, 1 (2015), 1–4.

[23] Leon Derczynski, Julie Binau, and Henri Schulte. 2020. Maintaining Quality in FEVER Annotation. In *Proceedings of the Third Workshop on Fact Extraction and VERification (FEVER)*. Association for Computational Linguistics, Online, 42–46. https://doi.org/10.18653/v1/2020.fever-1.6

[24] Leon Derczynski, Kalina Bontcheva, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Arkaitz Zubiaga. 2017. 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, Vancouver, Canada, 69–76. https://doi.org/10.18653/v1/S17-2006

[25] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[26] Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. ERASER: A Benchmark to Evaluate Rationalized NLP Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 4443–4458. https://doi.org/10.18653/v1/2020.acl-main.408

[27] Thomas Diggelmann, Jordan Boyd-Graber, Jannis Bulian, Massimiliano Ciaramita, and Markus Leippold. 2020. CLIMATE-FEVER: A Dataset for Verification of Real-World Climate Claims. In *Tackling Climate Change with Machine Learning Workshop at NeurIPS 2020*.

[28] Tien Huu Do, Xiao Luo, Duc Minh Nguyen, and Nikos Deligiannis. 2019. Rumour Detection Via News Propagation Dynamics and User Representation Learning. In *2019 IEEE Data Science Workshop (DSW)*. IEEE, Minneapolis, USA, 196–200.

[29] Joseph Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin* 76, 5 (1971), 378–382.

[30] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A Deep Semantic Natural Language Processing Platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*. Association for Computational Linguistics, Melbourne, Australia, 1–6. https://doi.org/10.18653/v1/W18-2501

[31] Max Glockner, Ivan Habernal, and Iryna Gurevych. 2020. Why do you think that? Exploring Faithful Sentence-Level Rationales Without Supervision. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 1080–1095. https://doi.org/10.18653/v1/2020.findings-emnlp.97

[32] Andreas Hanselowski, Hao Zhang, Zile Li, Daniil Sorokin, Benjamin Schiller, Claudia Schulz, and Iryna Gurevych. 2018. UKP-Athene: Multi-Sentence Textual Entailment for Claim Verification. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*. Association for Computational Linguistics, Brussels, Belgium, 103–108. https://doi.org/10.18653/v1/W18-5516

[33] Maram Hasanain, Reem Suwaileh, Tamer Elsayed, Alberto Barrón-Cedeno, and Preslav Nakov. 2019. Overview of the CLEF-2019 CheckThat! Lab: Automatic Identification and Verification of Claims. Task 2: Evidence and Factuality. In *Proceedings of CLEF (Working Notes)*. CEUR Workshop Proceedings, Lugano, Switzerland, 15.

[34] Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. 2017. A Joint Many-Task Model: Growing a Neural Network for Multiple NLP Tasks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, 1923–1933. https://doi.org/10.18653/v1/D17-1206

[35] Christopher Hidey, Tuhin Chakrabarty, Tariq Alhindi, Siddharth Varia, Kriste Krstovski, Mona Diab, and Smaranda Muresan. 2020. DeSePtion: Dual Sequence Prediction and Adversarial Examples for Improved Fact-Checking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 8593–8606. https://doi.org/10.18653/v1/2020.acl-main.761

[36] Christopher Hidey and Mona Diab. 2018. Team SWEEPer: Joint Sentence Extraction and Fact Checking with Pointer Networks. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*. Association for Computational Linguistics, Brussels, Belgium, 150–155. https://doi.org/10.18653/v1/W18-5525

[37] Christopher Hidey and Kathy McKeown. 2019. Fixed That for You: Generating Contrastive Claims with Semantic Edits. 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)*. Association for Computational Linguistics, Minneapolis, Minnesota, 1756–1767. https://doi.org/10.18653/v1/N19-1174

[38] Geoffrey E Hinton. 2002. Training products of experts by minimizing contrastive divergence. *Neural computation* 14, 8 (2002), 1771–1800.

[39] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural computation* 9, 8 (1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[40] Matthew Honnibal and Mark Johnson. 2015. An Improved Non-monotonic Transition System for Dependency Parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, 1373–1378. https://doi.org/10.18653/v1/D15-1162

[41] Yichen Jiang, Shikha Bordia, Zheng Zhong, Charles Dognin, Maneesh Singh, and Mohit Bansal. 2020. HoVer: A Dataset for Many-Hop Fact Extraction And Claim Verification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 3441–3460. https://doi.org/10.18653/v1/2020.findings-emnlp.309

[42] Mayank Jobanputra. 2019. Unsupervised Question Answering for Fact-Checking. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*. Association for Computational Linguistics, Hong Kong, China, 52–56. https://doi.org/10.18653/v1/D19-6609

[43] Rabeeh Karimi Mahabadi, Yonatan Belinkov, and James Henderson. 2020. End-to-End Bias Mitigation by Modelling Biases in Corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 8706–8716. https://doi.org/10.18653/v1/2020.acl-main.769

[44] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of International Conference on Learning Representations*. Toulon, France, 14.

[45] Dilek Küçük and Fazli Can. 2020. Stance detection: A survey. *ACM Computing Surveys (CSUR)* 53, 1 (2020), 1–37.

[46] Nayeon Lee, Belinda Li, Sinong Wang, Wen-tau Yih, Hao Ma, and Madian Khabsa. 2020. Language Models as Fact Checkers?. In *Proceedings of the Third Workshop on Fact Extraction and VERification (FEVER)*. Association for Computational Linguistics, Online, 36–41. https://doi.org/10.18653/v1/2020.fever-1.5

[47] Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., Online, 18.

[48] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, Venice, Italy, 2980–2988.

[49] Hui Liu, Qingyu Yin, and William Yang Wang. 2019. Towards Explainable NLP: A Generative Explanation Framework for Text Classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 5570–5581. https://doi.org/10.18653/v1/P19-1560

[50] Yang Liu and Mirella Lapata. 2019. Text Summarization with Pretrained Encoders. 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, Hong Kong, China, 3730–3740. https://doi.org/10.18653/v1/D19-1387

[51] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019), 13.

[52] Zhenghao Liu, Chenyan Xiong, Zhuyun Dai, Si Sun, Maosong Sun, and Zhiyuan Liu. 2020. Adapting Open Domain Fact Extraction and Verification to COVID-FACT through In-Domain Language Modeling. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 2395–2400. https://doi.org/10.18653/v1/2020.findings-emnlp.216

[53] Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2020. Fine-grained Fact Verification with Kernel Graph Attention Network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 7342–7351. https://doi.org/10.18653/v1/2020.acl-main.655

[54] Jackson Luken, Nanjiang Jiang, and Marie-Catherine de Marneffe. 2018. QED: A fact verification system for the FEVER shared task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*. Association for Computational Linguistics, Brussels, Belgium, 156–160. https://doi.org/10.18653/v1/W18-5526

[55] Jing Ma, Wei Gao, Shafiq Joty, and Kam-Fai Wong. 2019. Sentence-Level Evidence Embedding for Claim Verification with Hierarchical Attention Networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 2561–2571. https://doi.org/10.18653/v1/P19-1244

[56] Christopher Malon. 2018. Team Papelo: Transformer Networks at FEVER. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*. Association for Computational Linguistics, Brussels, Belgium, 109–113. https://doi.org/10.18653/v1/W18-5517

[57] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics, Baltimore, Maryland, 55–60. https://doi.org/10.3115/v1/P14-5010

[58] Tanushree Mitra and Eric Gilbert. 2015. Credbank: A large-scale social media corpus with associated credibility annotations.. In *ICWSM*. Oxford, UK, 258–267.

[59] Makoto Miwa and Mohit Bansal. 2016. End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, 1105–1116. https://doi.org/10.18653/v1/P16-1105

[60] Moin Nadeem, Wei Fang, Brian Xu, Mitra Mohtarami, and James Glass. 2019. FAKTA: An Automatic End-to-End Fact Checking System. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*. Association for Computational Linguistics, Minneapolis, Minnesota, 78–83. https://doi.org/10.18653/v1/N19-4014

[61] Duc Minh Nguyen, Tien Huu Do, Robert Calderbank, and Nikos Deligiannis. 2019. Fake News Detection using Deep Markov Random Fields. 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)*. Association for Computational Linguistics, Minneapolis, Minnesota, 1391–1400. https://doi.org/10.18653/v1/N19-1141

[62] Yixin Nie, Lisa Bauer, and Mohit Bansal. 2020. Simple Compounded-Label Training for Fact Extraction and Verification. In *Proceedings of the Third Workshop on Fact Extraction and VERification (FEVER)*. Association for Computational Linguistics, Online, 1–7. https://doi.org/10.18653/v1/2020.fever-1.1

[63] Yixin Nie, Haonan Chen, and Mohit Bansal. 2019. Combining fact extraction and verification with neural semantic matching networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. AAAI Press, Honolulu, Hawai, 6859–6866.

[64] Yixin Nie, Songhe Wang, and Mohit Bansal. 2019. Revealing the Importance of Semantic Retrieval for Machine Reading at Scale. 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, Hong Kong, China, 2553–2566. https://doi.org/10.18653/v1/D19-1258

[65] Piotr Niewinski, Maria Pszona, and Maria Janicka. 2019. GEM: Generative Enhanced Model for adversarial attacks. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*. Association for Computational Linguistics, Hong Kong, China, 20–26. https://doi.org/10.18653/v1/D19-6604

[66] Farhad Nooralahzadeh, Giannis Bekoulis, Johannes Bjerva, and Isabelle Augenstein. 2020. Zero-Shot Cross-Lingual Transfer with Meta Learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 4547–4562.

[67] Ray Oshikawa, Jing Qian, and William Yang Wang. 2020. A Survey on Natural Language Processing for Fake News Detection. In *Proceedings of the 12th Language Resources and Evaluation Conference*. European Language Resources Association, Marseille, France, 6086–6093. https://www.aclweb.org/anthology/2020.lrec-1.747

[68] Wojciech Ostrowski, Arnav Arora, Pepa Atanasova, and Isabelle Augenstein. 2020. Multi-Hop Fact Checking of Political Claims. *arXiv preprint arXiv:2009.06401* (2020).

[69] Wolfgang Otto. 2018. Team GESIS Cologne: An all in all sentence-based approach for FEVER. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*. Association for Computational Linguistics, Brussels, Belgium, 145–149. https://doi.org/10.18653/v1/W18-5524

[70] Bhargavi Paranjape, Mandar Joshi, John Thickstun, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. An Information Bottleneck Approach for Controlling Conciseness in Rationale Extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 1938–1952. https://doi.org/10.18653/v1/2020.emnlp-main.153

[71] Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A Decomposable Attention Model for Natural Language Inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, 2249–2255. https://doi.org/10.18653/v1/D16-1244

[72] Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. 2018. Automatic Detection of Fake News. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, Santa Fe, New Mexico, USA, 3391–3401. https://www.aclweb.org/anthology/C18-1287

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

for Computational Linguistics, New Orleans, Louisiana, 2227–2237. https://doi.org/10.18653/v1/N18-1202

[74] Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vassilis Plachouras, Tim Rocktäschel, et al. 2020. KILT: a benchmark for knowledge intensive language tasks. *arXiv preprint arXiv:2009.02252* (2020).

[75] Przemysław Pobrotyn, Tomasz Bartczak, Mikołaj Synowiec, Radosław Białobrzeski, and Jarosław Bojar. 2020. Context-Aware Learning to Rank with Self-Attention. In *Proceedings of the ECOM'20: The SIGIR 2020 Workshop on eCommerce*. Association for Computing Machinery, Online.

[76] Kashyap Popat, Subhabrata Mukherjee, Jannik Strötgen, and Gerhard Weikum. 2016. Credibility Assessment of Textual Claims on the Web. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (CIKM '16)*. Association for Computing Machinery, New York, NY, USA, 2173–2178. https://doi.org/10.1145/2983323.2983661

[77] Beatrice Portelli, Jason Zhao, Tal Schuster, Giuseppe Serra, and Enrico Santus. 2020. Distilling the Evidence to Augment Fact Verification Models. In *Proceedings of the Third Workshop on Fact Extraction and VERification (FEVER)*. Association for Computational Linguistics, Online, 47–51. https://doi.org/10.18653/v1/2020.fever-1.7

[78] Adithya Pratapa, Sai Muralidhar Jayanthi, and Kavya Nerella. 2020. Constrained Fact Verification for FEVER. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 7826–7832. https://doi.org/10.18653/v1/2020.emnlp-main.629

[79] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. (2018).

[80] Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. Truth of Varying Shades: Analyzing Language in Fake News and Political Fact-Checking. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, 2931–2937. https://doi.org/10.18653/v1/D17-1317

[81] Estela Saquete, David Tomás, Paloma Moreda, Patricio Martínez-Barco, and Manuel Palomar. 2020. Fighting post-truth using natural language processing: A review and open challenges. *Expert Systems with Applications* 141 (2020), 112943. https://doi.org/10.1016/j.eswa.2019.112943

[82] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. FaceNet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Boston, USA, 815–823.

[83] Tal Schuster, Darsh Shah, Yun Jie Serene Yeo, Daniel Roberto Filizzola Ortiz, Enrico Santus, and Regina Barzilay. 2019. Towards Debiasing Fact Verification Models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 3419–3425. https://doi.org/10.18653/v1/D19-1341

[84] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603* (2016). https://arxiv.org/pdf/1611.01603

[85] Darsh Shah, Tal Schuster, and Regina Barzilay. 2020. Automatic Fact-Guided Sentence Modification. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 05 (Apr. 2020), 8791–8798. https://doi.org/10.1609/aaai.v34i05.6406

[86] Gautam Kishore Shahi and Durgesh Nandini. 2020. FakeCovid-A Multilingual Cross-domain Fact Check News Dataset for COVID-19. In *Proceedings of the CySoc 2020 International Workshop on Cyber Social Threats*. AAAI Press, Online, 9.

[87] Karishma Sharma, Feng Qian, He Jiang, Natali Ruchansky, Ming Zhang, and Yan Liu. 2019. Combating Fake News: A Survey on Identification and Mitigation Techniques. *ACM Trans. Intell. Syst. Technol.* 10, 3, Article 21 (April 2019), 42 pages. https://doi.org/10.1145/3305260

[88] Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu. 2020. FakeNewsNet: A Data Repository with News Content, Social Context, and Spatiotemporal Information for Studying Fake News on Social Media. *Big Data* 8, 3 (2020), 171–188.

[89] Amir Soleimani, Christof Monz, and Marcel Worring. 2020. BERT for Evidence Retrieval and Claim Verification. In *Advances in Information Retrieval*, Joemon M. Jose, Emine Yilmaz, João Magalhães, Pablo Castells, Nicola Ferro, Mário J. Silva, and Flávio Martins (Eds.). Springer International Publishing, Cham, 359–366.

[90] Dominik Stammbach and Elliott Ash. 2020. e-FEVER: Explanations and Summaries for Automated Fact Checking. In *Conference on Truth and Trust Online*. Online, 12.

[91] Dominik Stammbach and Guenter Neumann. 2019. Team DOMLIN: Exploiting Evidence Enhancement for the FEVER Shared Task. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*. Association for Computational Linguistics, Hong Kong, China, 105–109. https://doi.org/10.18653/v1/D19-6616

[92] Shyam Subramanian and Kyumin Lee. 2020. Hierarchical Evidence Set Modeling for Automated Fact Extraction and Verification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 7798–7809. https://doi.org/10.18653/v1/2020.emnlp-main.627

[93] Sandeep Suntwal, Mithun Paul, Rebecca Sharp, and Mihai Surdeanu. 2019. On the Importance of Delexicalization for Fact Verification. 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, Hong Kong, China, 3413–3418. https://doi.org/10.18653/v1/D19-1340

[94] Eugenio Tacchini, Gabriele Ballarin, Marco L Della Vedova, Stefano Moret, and Luca de Alfaro. 2017. Some like it Hoax: Automated fake news detection in social networks. In *2nd Workshop on Data Science for Social Good, SoGood 2017*. CEUR Workshop Proceedings, Skopje, Macedonia, 1–15.

[95] Motoki Taniguchi, Tomoki Taniguchi, Takumi Takahashi, Yasuhide Miura, and Tomoko Ohkuma. 2018. Integrating Entity Linking and Evidence Ranking for Fact Extraction and Verification. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*. Association for Computational Linguistics, Brussels, Belgium, 124–126. https://doi.org/10.18653/v1/W18-5520

[96] James Thorne and Andreas Vlachos. 2018. Automated Fact Checking: Task Formulations, Methods and Future Directions. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, Santa Fe, New Mexico, USA, 3346–3359. https://www.aclweb.org/anthology/C18-1283

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

[98] James Thorne and Andreas Vlachos. 2020. Avoiding catastrophic forgetting in mitigating model biases in sentence-pair classification with elastic weight consolidation. *arXiv preprint arXiv:2004.14366* (2020).

[99] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. 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)*. Association for Computational Linguistics, New Orleans, Louisiana, 809–819. https://doi.org/10.18653/v1/N18-1074

[100] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2019. Evaluating adversarial attacks against multiple fact verification systems. 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, Hong Kong, China, 2944–2953. https://doi.org/10.18653/v1/D19-1292

[101] James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018. The Fact Extraction and VERification (FEVER) Shared Task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*. Association for Computational Linguistics, Brussels, Belgium, 1–9. https://doi.org/10.18653/v1/W18-5501

[102] James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2019. The FEVER2.0 Shared Task. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*. Association for Computational Linguistics, Hong Kong, China, 1–6. https://doi.org/10.18653/v1/D19-6601

[103] Santosh Tokala, Vishal G, Avirup Saha, and Niloy Ganguly. 2019. AttentiveChecker: A Bi-Directional Attention Flow Mechanism for Fact Verification. 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)*. Association for Computational Linguistics, Minneapolis, Minnesota, 2218–2222. https://doi.org/10.18653/v1/N19-1230

[104] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. Curran Associates, Inc., Long Beach, USA, 5998–6008.

[105] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in neural information processing systems*. Curran Associates, Inc., Montreal, Canada, 2692–2700.

[106] Andreas Vlachos and Sebastian Riedel. 2014. Fact Checking: Task definition and dataset construction. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*. Association for Computational Linguistics, Baltimore, MD, USA, 18–22. https://doi.org/10.3115/v1/W14-2508

[107] David Wadden, Kyle Lo, Lucy Lu Wang, Shanchuan Lin, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or Fiction: Verifying Scientific Claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online, 17.

[108] Eric Wallace, Jens Tuyls, Junlin Wang, Sanjay Subramanian, Matt Gardner, and Sameer Singh. 2019. AllenNLP Interpret: A Framework for Explaining Predictions of NLP Models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*. Association for Computational Linguistics, Hong Kong, China, 7–12. https://doi.org/10.18653/v1/D19-3002

[109] Jian Wang, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin. 2017. Deep metric learning with angular loss. In *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, Venice, Italy, 2593–2601.

[110] William Yang Wang. 2017. "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)*. Association for Computational Linguistics, Vancouver, Canada, 422–426. https://doi.org/10.18653/v1/P17-2067

[111] Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. 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)*. Association for Computational Linguistics, New Orleans, Louisiana, 1112–1122. https://doi.org/10.18653/v1/N18-1101

[112] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. 2017. Sampling matters in deep embedding learning. In *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, Venice, Italy, 2840–2848.

[113] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* (2016).

[114] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-End Neural Ad-Hoc Ranking with Kernel Pooling. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '17)*. Association for Computing Machinery, New York, NY, USA, 55–64. https://doi.org/10.1145/3077136.3080809

[115] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*. Curran Associates, Inc., Vancouver, Canada, 5753–5763.

[116] Deming Ye, Yankai Lin, Jiaju Du, Zhenghao Liu, Peng Li, Maosong Sun, and Zhiyuan Liu. 2020. Coreferential Reasoning Learning for Language Representation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 7170–7186. https://doi.org/10.18653/v1/2020.emnlp-main.582

[117] Wenpeng Yin and Dan Roth. 2018. TwoWingOS: A Two-Wing Optimization Strategy for Evidential Claim Verification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 105–114. https://doi.org/10.18653/v1/D18-1010

[118] Wenpeng Yin and Hinrich Schütze. 2018. Attentive Convolution: Equipping CNNs with RNN-style Attention Mechanisms. *Transactions of the Association for Computational Linguistics* 6 (2018), 687–702. https://doi.org/10.1162/tacl_a_00249

[119] Takuma Yoneda, Jeff Mitchell, Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. UCL Machine Reading Group: Four Factor Framework For Fact Finding (HexaF). In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*. Association for Computational Linguistics, Brussels, Belgium, 97–102. https://doi.org/10.18653/v1/W18-5515

[120] Chen Zhao, Chenyan Xiong, Corby Rosset, Xia Song, Paul Bennett, and Saurabh Tiwary. 2020. Transformer-XH: Multi-Evidence Reasoning with eXtra Hop Attention. In *International Conference on Learning Representations*. Online, 16.

[121] Wanjun Zhong, Jingjing Xu, Duyu Tang, Zenan Xu, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2020. Reasoning Over Semantic-Level Graph for Fact Checking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 6170–6180. https://doi.org/10.18653/v1/2020.acl-main.549

[122] Jie Zhou, Xu Han, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2019. GEAR: Graph-based Evidence Aggregating and Reasoning for Fact Verification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 892–901. https://doi.org/10.18653/v1/P19-1085

[123] Xinyi Zhou and Reza Zafarani. 2020. A Survey of Fake News: Fundamental Theories, Detection Methods, and Opportunities. *ACM Comput. Surv.* 53, 5, Article 109 (Sept. 2020), 40 pages. https://doi.org/10.1145/3395046

[124] Arkaitz Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter. 2018. Detection and resolution of rumours in social media: A survey. *ACM Computing Surveys (CSUR)* 51, 2 (2018), 1–36.

[125] Arkaitz Zubiaga, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Peter Tolmie. 2016. Analysing how people orient to and spread rumours in social media by looking at conversational threads. *PloS one* 11, 3 (2016), e0150989.