A Review on Fact Extraction and VERification: The FEVER case

Giannis Bekoulis[1,2] **Christina Papagiannopoulou** **Nikos Deligiannis**[1,2]
[1]ETRO, Vrije Universiteit Brussel, 1050 Brussels, Belgium
[2]imec, Kapeldreef 75, 3001 Leuven, Belgium
{gbekouli,ndeligia}@etrovub.be

Abstract

Fact Extraction and VERification (FEVER) is a recently introduced task which aims to identify the veracity of a given claim based on Wikipedia documents. A lot of methods have been proposed to address this problem which consists of the subtasks of (i) retrieving the relevant documents (and sentences) from Wikipedia and (ii) validating whether the information in the documents supports or refutes a given claim. This task is essential since it can be the building block of applications that require a deep understanding of the language such as fake news detection and medical claim verification. In this paper, we aim to get a better understanding of the challenges in the task by presenting the literature in a structured and comprehensive way. In addition, we describe the proposed methods by analyzing the technical perspectives of the different approaches and discussing the performance results on the FEVER dataset.

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 already occurred 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 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. Since the main units of a document are sentences, the goal of the fact checking system is to 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 (Bowman et al., 2015). However, we should note that an NLI system (i.e., used to validate the veracity of 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 that, 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 focus on the recently introduced problem of FEVER (Thorne et al., 2018a). 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. (2018a) and a competition (i.e., shared task) has been organised, 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) (Kipf and Welling, 2017), language models (Devlin et al., 2019) – have been presented in top-tier NLP venues to resolve the task which is far from being solved (i.e., score of 70.60 percentage points).

This work aims at summarizing the methods introduced to resolve the FEVER task, which has the specific characteristic that it comprises multiple subtasks. Unlike the work of Thorne and Vlachos (2018) 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. Note that 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, by presenting 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 providing an easy way to compare systems and experiments, and investigating the contributions of each individual model component in the overall performance.

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

Figure 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 core methods that are exploited in several studies 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 preprocessed (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 for each claim to return the SUPPORTED or REFUTED label along with the corresponding evidence while in the case of the NEI label, no evidence is returned. In 16.82% of the claims, more than one evidence sentences are needed to conclude about the veracity of the claim. 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 number 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 has 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 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 (Fleiss, 1971) 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 Preliminaries

In the literature, the FEVER task has been mostly treated as a series of three subtasks, namely document retrieval, sentence retrieval and claim verification (Thorne et al., 2018a; Nie et al., 2019a; Yoneda et al., 2018). In this section, we describe the methods that have been mostly used to solve the aforementioned subtasks.

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. (2018b).

2.3.1 Document Retrieval

Document retrieval is the task which aims at matching a query against a collection of unstructured documents and return the most relevant articles (Chen et al., 2017a).

DrQA: Several approaches, which have been exploited to partly solve the FEVER task, rely on the DrQA component (Chen et al., 2017a) 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.

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. The tasks of sentence retrieval and claim verification are commonly framed as NLI problems and are treated with NLI methods.

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 (Bowman et al., 2015), the Multi-Genre Natural Language Inference (MultiNLI) Corpus (Williams et al., 2018), and the cross-lingual NLI (XNLI) Corpus (Conneau et al., 2018). Several approaches have been proposed to solve the NLI tasks, however the most fundamental neural models that have been exploited in the context of the FEVER task are the Decomposable

Attention (DA) (Parikh et al., 2016), Enhanced Long Short-Term Memory (LSTM) for Natural Language Inference (ESIM) (Chen et al., 2017b), Bidirectional Encoder Representations from Transformers (BERT)-based NLI (Devlin et al., 2019).

DA: The model has been proposed in the work of Parikh et al. (2016) and unlike the trend of using LSTMs, DA solely relies on word embeddings in order not to increase the complexity by $\mathcal{O}(d^2)$ 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. (2015) (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. (2017b) rely on LSTM models to perform the NLI task. In particular, the model exploits the use of bidirectional LSTMs (Hochreiter and Schmidhuber, 1997) (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 (Parikh et al., 2016) 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) (Peters et al., 2018), OpenAI GPT (Radford et al., 2018) and BERT (Devlin et al., 2019). In the context of the FEVER task, several models (Liu et al., 2020; Soleimani et al., 2020) rely on the pre-trained BERT model. The BERT relies on WordPiece embeddings (Wu et al., 2016) and on Transformer networks (Vaswani et al., 2017). 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 WordPiece 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.

2.4 Baseline Model

Along with the FEVER dataset, Thorne et al. (2018a) 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. (2020); Zhong et al. (2020)) 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 (Yin and Roth, 2018).

2.4.1 Document Retrieval

For this subtask, Thorne et al. (2018a) exploited the DrQA module, which has been extensively described in Section 2.3.1, and used cosine similarity to obtain the k most similar documents to the claim based on the TF-IDF word representation.

2.4.2 Sentence Selection

For sentence selection, in the proposed three-step model, Thorne et al. (2018a) obtained the most similar sentences from the retrieved documents (see previous subtask) by using either DrQA or unigram

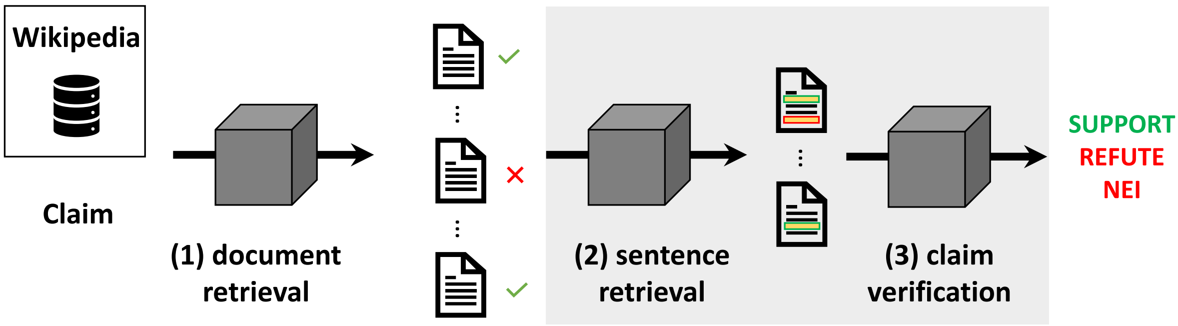

Figure 2: A three-step pipeline model for the FEVER task. It consists of three components, namely document retrieval, sentence retrieval and claim verification. The input to the first component is Wikipedia and a given claim sentence. The output of the first component is a set of Wikipedia documents 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.

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

2.4.3 Claim verification

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 (Parikh et al., 2016) – which was described in Section 2.3.2 – has been used as a state-of-the-art system in NLI (Bowman et al., 2015) (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.

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 (Thorne et al., 2018b) 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 based on the work of Thorne et al. (2018a) that defined 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 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. Note that a claim is considered correct in the case that at least a complete evidence group is identi-

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

fied. 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.

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 (Hanselowski et al., 2018; Zhong et al., 2020) 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 (Yin and Roth, 2018; Nie et al., 2020). For a detailed description of these joint architectures, see Section 3.2. Table 2 is 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. Note that 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.

Mention-based methods

Hanselowski et al. (2018) 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. (2018). 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. The component (ii) presented in the work of Hanselowski et al. (2018) 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. (2018) is also followed by Zhou et al. (2019); Chernyavskiy and Ilvovsky (2019); Stammbach and Neumann (2019); Zhao et al. (2020); Liu et al. (2020); Soleimani et al. (2020), sometimes with minor modifications. Another piece of work that the value of named entities is being exploited is the work of Malon (2018); Chakrabarty et al. (2018); Hidey and Diab (2018); Yin and Roth (2018). The work of Chakrabarty et al. (2018), 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. Note that the works of Malon (2018); Chakrabarty et al. (2018) 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. (2019a) ranked at the first position at the FEVER competition. For that, they presented a three-stage model that relies on Neural Semantic Matching Network (NSNM), i.e., a variation of ESIM (Chen et al., 2017b) (see Section 2.3.2). 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

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

	Model	Document Retrieval			Sentence Retrieval				Claim Verification			Joint
		Mention	Keyword	Other	TF-IDF	ESIM	LM	Other	ESIM	LM	Other	
2018	Hanselowski et al. (2018)	✓				✓			✓			
	Thorne et al. (2018a)			✓	✓							
	Yoneda et al. (2018)			✓				✓	✓			
	Yin and Schütze (2018a)			✓	✓						✓	
	Hidey and Diab (2018)	✓		✓								✓
	Chakrabarty et al. (2018)	✓			✓						✓	
	Malon (2018)	✓									✓	
	Luken et al. (2018)		✓					✓			✓	
	Yin and Roth (2018)	✓										✓
	Taniguchi et al. (2018)			✓	✓						✓	
2019	Nie et al. (2019a)		✓			✓			✓			
	Nie et al. (2019b)		✓				✓			✓		
	Ma et al. (2019)		✓									
	Zhou et al. (2019)	✓										
	Chernyavskiy and Ilvovsky (2019)	✓			✓					✓		
	Stammbach and Neumann (2019)	✓				✓	✓			✓		
	Jobanputra (2019)									✓		✓
2020	Zhao et al. (2020)	✓					✓			✓		
	Liu et al. (2020)	✓					✓			✓		
	Soleimani et al. (2020)	✓					✓			✓		
	Zhong et al. (2020)		✓				✓			✓		
	Portelli et al. (2020)		✓		✓					✓		
	Lee et al. (2020)									✓		✓
	Lewis et al. (2020)									✓		✓
	Nie et al. (2020)		✓			✓			✓			✓

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.

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 (Ma et al., 2019; Nie et al., 2019b; Zhong et al., 2020; Portelli et al., 2020) exploit the document retrieval module developed by Nie et al. (2019a). The work of Luken et al. (2018) is also designed to extract part-of-speech tags, dependencies, etc. by using the CoreNLP parser (Manning et al., 2014) 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 (2018a) relies on the work of the baseline model (Thorne et al., 2018a) as presented in Section 2.4. Similar to the baseline model, the work of Hidey and Diab (2018) exploit DrQA along with hand-crafted features and neural methods for the document retrieval task. Yoneda et al. (2018) 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. (2018) aims for high precision using exact matching techniques. In addition, as we observe in Table 2, most of the works that have been developed for the competition shared task (2018) focus on hand-crafted 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.

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.

TF-IDF
For the sentence retrieval task, several pipeline methods in the literature rely on the sentence retrieval component of the baseline method (Thorne et al., 2018a). Specifically, these methods (Chernyavskiy and Ilvovsky, 2019; Portelli et al., 2020; Taniguchi et al., 2018; Yin and Schütze, 2018a), use a TF-IDF vector representation along with a cosine similarity function (see Section 2.4 for a detailed description). However, there are some attempts that exploit additional representations such as the ELMo embeddings (Chakrabarty et al., 2018).

ESIM-Based
An important line of research (Hanselowski et al., 2018; Nie et al., 2019a; Zhou et al., 2019) for the sentence selection subtask of the FEVER task includes the use of ESIM-based models (Chen et al. (2017b), see also Section 2.3.2 for more details on the ESIM architecture). Those works formulate the sentence selection subtask as an NLI problem where the claim is the "premise" sentence and the potential evidence sentence as a "hypothesis". Hanselowski et al. (2018) 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. (2019) exploits the evidences retrieved by the model of Hanselowski et al. (2018). Similar to Hanselowski et al. (2018), Nie et al. (2019a) uses 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). Similar to the work of Hanselowski et al. (2018), Nie et al. (2019a) 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. (2018) that train a pairwise hinge loss, Nie et al. (2019a) exploit a cross-entropy loss for training their model.

Language Model Based
Similar to the ESIM-based methods, language model based methods (Nie et al., 2019b; Zhong et al., 2020; Soleimani et al., 2020; Liu et al., 2020; Zhao et al., 2020) 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 in Section 2.3.2. It is however worth mentioning that the models developed for the sentence retrieval component do not rely only on BERT but also on RoBERTa (Liu et al., 2019b) and XLNet (Yang et al., 2019). For the language model based sentence retrieval two types of losses have been exploited (i) pointwise loss, where a cross-entropy classifier is used to predict 0 or 1, depending on whether the claim and the potential evidence sentence are related, and (ii) 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 (Soleimani et al., 2020; Nie et al., 2019b). Unlike the previous works that use the pointwise loss, the work of Zhong et al. (2020) use the RoBERTa and XLNet models pre-trained models. For loss of type (ii), the proposed architectures rely only on the BERT pre-trained language model (Soleimani et al., 2020; Zhao et al., 2020; Liu et al., 2020). Due to the high number of negative examples with respect to the number of positive examples in the sentence retrieval subtask, Soleimani et al. (2020) proposed to use hard negative mining similar to Schroff et al. (2015) 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. Note that, as we observe in Table 2, most recent works (i.e., developed in 2019-2020) focus on developing language model based approaches.

Other Methods
A model able to combine both the ESIM-based and the language model based sentence retrieval components is the two-step model of Stammbach and Neumann (2019). This work relies on the model of Nie et al. (2019a) 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 (Luken et al., 2018; Otto, 2018; Yoneda et al., 2018; Stammbach and Neumann, 2019) that make use of different strategies for sentence retrieval. Luken et al. (2018) use the root, the nouns and the names 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. (2018), Otto (2018) also relies on nouns and named entities extracted from the claim by using the spaCy NLP library (Honnibal and Johnson, 2015). This work is able to directly retrieve evidences using the Solr indexer[3] without relying on a document retrieval component. Unlike previous studies (Luken et al., 2018; Otto, 2018), Yoneda et al. (2018) 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.

3.1.3 Claim Verification

In general, most of the claim verification methods use neural model components. In the baseline method, as discussed in Section 2.4, 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.

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

ESIM-based
Hanselowski et al. (2018) 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 relevant classes (e.g., SUPPORTED, REFUTED, NEI). The winning system of the FEVER task proposed by Nie et al. (2019a) 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. (2018) 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. (2018) used an MLP classifier on top of the prediction score of each evidence sentence .

Language Model Based
Language models have been also successfully applied on the claim verification subtask of FEVER. Soleimani et al. (2020) 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 (2018) (i.e., default label is NEI and if there is a SUPPORTED label then the label of the claim is also SUPPORTED). BERT-based models have been also adopted by multiple studies e.g., Nie et al. (2019b); Stammbach and Neumann (2019); Chernyavskiy and Ilvovsky (2019); Portelli et al. (2020); Stammbach and Ash (2020). Zhou et al. (2019) proposed a BERT-based method that makes use of GNNs (Kipf and Welling, 2017). By using GNNs, where evidences are nodes in a graph, they are able to exchange information between the nodes, performing reasoning in order to obtain the final class verification label. Similar to the work of Zhou et al. (2019) is the work of Liu et al. (2020) where they exploit the use of kernel attention (Xiong et al., 2017) both at sentence and

token level to propagate information among the evidence nodes. A graph-based approach is also exploited in the work of Zhong et al. (2020) 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. Finally, Zhao et al. (2020) 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. Note also that as we observe in Table 2 most recent works (i.e., developed in 2019-2020) focus on developing language model based approaches.

Other Neural Models

Some additional neural models that cannot be classified in any of the aforementioned categories have been proposed (Chakrabarty et al., 2018; Yin and Schütze, 2018a; Luken et al., 2018; Malon, 2018; Taniguchi et al., 2018; Otto, 2018). Specifically, Chakrabarty et al. (2018) rely on bidirectional LSTMs and perform operations among the claim and evidence representation (e.g., element-wise product, concatenation) similar to Conneau et al. (2017). Other studies use the DA model (Luken et al., 2018; Otto, 2018) similar to the one that was exploited in the baseline model (for details, see Section 2.3.2). Other methods such as Convolution Neural Networks (CNNs) have been also used in combination with attention mechanisms and transformer networks have been used by Yin and Schütze (2018a); Taniguchi et al. (2018) and Malon (2018), respectively.

3.2 Joint models

Unlike all the aforementioned studies presented so far in Section 3 that 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 (2018) proposed the use of a multitask learning architecture with CNNs and attentive convolutions (Yin and Schütze, 2018b) in order to extract coarse-grained (i.e., sentence-level) and fine-grained (i.e., with attention over the words) sentence representations of claim and evidences to perform the tasks of sentence selection and claim verification in a joint setting. Similar to this work, Hidey and Diab (2018) 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 (Vinyals et al., 2015) for the sentence selection subtask and an MLP-based architecture for claim verification. A newer version of this system that use also adversarial instances to improve the performance has also been proposed in the work of Hidey et al. (2020). Nie et al. (2020) perform an experimental study, where the NSNM model (Nie et al., 2019a) 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 train the models in a multitask learning fashion, Jobanputra (2019); Lee et al. (2020) formulate the problem as a clozed task. In the work of Lee et al. (2020), the last entity of the claim is masked out and the missing entity is being filled in with a language model. That way, a new evidence is being created (eliminating the need for a sentence retrieval module) and along with the claim, the claim verification label using an MLP is predicted. Similar to that work, Jobanputra (2019) also rely on language models and eliminate the sentence retrieval step. Finally, the work of Lewis et al. (2020) is an end-to-end system that is performing the three steps at once. Specifically, for the retrieval an off-the-shelf retriever (Karpukhin et al., 2020) is being used and in the claim verification substask, they use a sequence-to-sequence model that has as goal in the decoder part to predict the tokens of the labels (e.g., `SUPPORTED`).

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. Note that the

	Pre-calculated Model	Features	Oracle κ	Fully Accuracy	Supported
Thorne et al. (2018a)	✓	5	55.30	70.20	
Yin and Roth (2018)	✓	5	89.63	93.08	
Hidey and Diab (2018)	✓	5	90.70	-	
Hanselowski et al. (2018)	✓	7	-	**93.55**	
Chakrabarty et al. (2018)	✓	3	**94.40**	-	
Zhou et al. (2019)	✓	7	-	93.33	
Nie et al. (2019a)	✓	5	-	92.42	

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.

performance of the joint models is also presented in the corresponding subsections.

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 FEVER task as introduced in the work of Thorne et al. (2018a,b) 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. The model of Thorne et al. (2018a) is the baseline model as presented in Section 2.4. It is worth mentioning that in the various works, there is not 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. Note that the works of Hanselowski et al. (2018) and Nie et al. (2019a) are the systems that most of the recent neural methods (see Section 3.1.1 for more details) rely on. The document retrieval presented in the work of Zhou et al. (2019) reproduces the results of the Hanselowski et al. (2018) model. In terms of oracle accuracy score, the model of Chakrabarty et al. (2018) 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 (Yin and Roth, 2018; Hidey and Diab, 2018; Chakrabarty et al., 2018) that rely on mention-based document retrieval achieve higher performance compared to the baseline model. The same holds for the fully supported evaluation metric. In particular, all the models (Yin and Roth, 2018; Hanselowski et al., 2018; Zhou et al., 2019; Nie et al., 2019a) score higher compared to the baseline. This is again because these models rely on mention- and keyword-based techniques for document retrieval. Note that the gap between the keyword-based model of Nie et al. (2019a) is relatively small in terms of the fully supported evaluation metric compared to the mention-based approaches.

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. (2018a) (i.e., 9,999 dev and 9,999 test instances) and not on the dev and test sets of the shared task (Thorne et al., 2018b) (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. (2018) optimizes over the precision evaluation metric, the system of Nie et al. (2019b) optimizes

[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. (2018a)	✓	-	-	17.20	11.28	47.87	18.26
	Taniguchi et al. (2018)	✓	-	-	-	11.37	29.79	16.49
	Chakrabarty et al. (2018)	✓	-	78.04	-	23.02	75.89	35.33
ESIM-based	Hanselowski et al. (2018)	✗	24.07	86.72	37.69	23.51	84.66	36.80
	Nie et al. (2019a)	✗	36.49	86.79	51.38	-	-	52.96
Language models	Nie et al. (2019b)	✗	-	-	**76.87**	-	-	**74.62**
	Soleimani et al. (2020)**	✗	38.18	88.00	53.25	-	-	38.61
	Liu et al. (2020)	✗	27.29	**94.37**	42.34	25.21	**87.47**	34.14
	Zhong et al. (2020)	✗	26.67	87.64	40.90	25.63	85.57	39.45
Other models	Luken et al. (2018)	✓	**77.50**	52.30	62.50	**77.23**	47.12	58.53
	Otto (2018)	✓	-	-	-	12.09	51.69	19.60
	Yoneda et al. (2018)	✓	22.74	84.54	35.84	22.16	82.84	34.97
	Stammbach and Neumann (2019)**	✗	25.10	89.80	39.30	-	-	-
Joint models	Yin and Roth (2018)*	✗	53.81	57.73	50.59	49.91	44.68	47.15
	Hidey and Diab (2018)	✗	-	-	-	18.48	75.39	29.69
	Nie et al. (2020)	✗	-	-	-	-	-	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. (2018a) (i.e., 9,999 dev and 9,999 test instances) and not on the dev and test sets of the shared task (Thorne et al., 2018b). 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.

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 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, note that 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 problem 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).

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. (2018) (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	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. (2019b)) 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[5].

model of Stammbach and Neumann (2019) (which is a combination of an ESIM-based and a BERT-based system). It is worthwhile mentioning the experiment of Liu et al. (2019b) 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. (2020) and in the work of Liu et al. (2019b). The experimental study of Soleimani et al. (2020) 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 (Schroff et al., 2015) is used in order to select more difficult instances. On the other hand, in the work of Liu et al. (2019b) (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, which suggests that the pointwise loss overfits 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. Due to that fact that from the two experimental studies (i.e., the one of Soleimani et al. (2020) and the one of Liu et al. (2019b)), there is no consistent conclusion, this suggests that the effect of the loss function in the sentence retrieval task needs further investigation.

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 (Thorne et al., 2018b). 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 have 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 (2018) 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 (2018), 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 (2018) are reported on the splits defined in the work of Thorne et al. (2018a) and not on the splits of the shared task (Thorne et al., 2018b). In general, joint models have shown an improved performance in a number of tasks (e.g., entity-relation extraction (Miwa and Bansal, 2016; Bekoulis et al., 2018a), POS tagging-dependency parsing-chunking-semantic relatedness-textual entailment (Hashimoto et al., 2017)) since the error propagation between the various sequential tasks is

	Model	Pre-calculated Features	Dev Label Accuracy	Dev FEVER	Test Label Accuracy	Test FEVER
ESIM-based	Hanselowski et al. (2018)	✗	68.49	64.74	65.46	61.58
	Yoneda et al. (2018)**	✗	69.66	65.41	67.62	62.52
	Nie et al. (2019a)	✓	69.60	66.14	68.16	64.23
Language models	Nie et al. (2019b)	✗	75.12	70.18	72.56	67.26
	Chernyavskiy and Ilvovsky (2019)**	✗	-	-	71.72	67.68
	Zhou et al. (2019)	✗	74.84	70.69	71.60	67.10
	Portelli et al. (2020)*	✗	**84.33**	-	-	-
	Soleimani et al. (2020)	✗	74.59	72.42	71.86	69.66
	Liu et al. (2020)**	✗	78.29	**76.11**	74.07	70.38
	Zhong et al. (2020)	✓	79.16	-	**76.85**	70.60
	Zhao et al. (2020)**	✗	78.05	74.98	72.39	69.07
	Stammbach and Ash (2020)**	✗	-	-	76.60	**74.27**
Other models	Thorne et al. (2018a)	✓	-	-	48.84	27.45
	Chakrabarty et al. (2018)	✗	58.77	50.83	57.45	49.06
	Luken et al. (2018)	✓	44.70	43.90	50.12	43.42
	Malon (2018)**	✗	-	58.44	61.08	57.36
	Taniguchi et al. (2018)	✗	-	-	47.13	38.81
	Otto (2018)	✗	-	-	54.15	40.77
Joint models	Yin and Roth (2018)*	✗	78.90	56.16	75.99	54.33
	Hidey and Diab (2018)	✗	-	-	59.72	49.94
	Nie et al. (2020)	✗	-	-	66.21	62.69
	Lee et al. (2020)	✓	-	-	57.00	-
	Lewis et al. (2020)	✗	74.50	-	72.50	-
	Hidey et al. (2020)	✓	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. (2018a) (i.e., 9,999 dev and 9,999 test instances) and not on the dev and test sets of the shared task (Thorne et al., 2018b). 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.

alleviated. However this is not the case for the proposed joint architectures for the FEVER problem except for the model of Hidey et al. (2020). 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 (2020) 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). The model presented in the work of Zhong et al. (2020) lead to the best performance in terms of label accuracy both in the dev and test sets. This is due to the fact that the model relies on the semantic role labeling tool of AllenNLP[6] 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 contribution of the semantic roles is also evident from the fact that other works that rely on graph neural networks (see Zhou et al. (2019); Liu et al. (2020)) achieve lower performance in terms of label accuracy and FEVER score. We should

[6] `https://demo.allennlp.org/semantic-role-labeling`

note that the use of external tools 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 (2016)). However as presented in the work of Bekoulis et al. (2018a,b), 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 performance of the four models of Soleimani et al. (2020); Liu et al. (2020); Zhong et al. (2020); Zhao et al. (2020) that 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). Although, the models of Liu et al. (2020); Zhong et al. (2020); Zhao et al. (2020) are far more complex in terms of the proposed architectures compared to the plain BERT-based model of Soleimani et al. (2020), the main benefit 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 which are the related problems 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 (Thorne et al., 2019). 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 authors of the shared task have also introduced metrics for evaluating the quality of the generated adversarial instances. Niewinski et al. (2019) 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 work consider automated fact checking as a constituent of fake news, see e.g., this review paper on fake news detection (Oshikawa et al., 2020). 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 on the downstream application. It is common in applications, which assess the validity of fake news (e.g., posts, articles), to use fake/non-fake as prediction labels (Pérez-Rosas et al., 2018). 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 (2015)). For solving the fake news detection problem several approaches have been proposed ranging from feature-based models (Conroy et al., 2015) to neural network architectures (Rashkin et al., 2017; Nguyen et al., 2019).

Rumour Detection: According to the work of Zubiaga et al. (2018), 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 (Zubiaga et al., 2016), 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. (2019)) 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. (2018), 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 (Küçük and Can, 2020).

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, which follows the paradigm of the FEVER dataset in the context of validating scientific claims, has been proposed by Wadden et al. (2020). Specifically, in that work, Wadden et al. (2020) 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 (2017) for fake news detection. In particular, the author of this work created a dataset, which consists of 12.8k statements manually annotated from the POLITIFACT.COM website. Unlike the FEVER dataset that 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. (2018). 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. (2019) 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 the 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 (2019) 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 that relies on claims generated from Wikipedia, the MutliFC dataset proposed in the work Augenstein et al. (2019) relies on 26 fact checking websites. In particular, they constructed a dataset of 34,918 claims. The claims crawled from various domains, where each domain has a different number of labels and this is also the challenge 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 FakeCovid (Shahi and Nandini, 2020). 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 (Zubiaga et al., 2016), SOME-LIKE-IT-HOAX (Tacchini et al., 2017), CLEF-2019 tasks 1 & 2 (Atanasova et al., 2019; Hasanain et al., 2019). Also, in the works of Popat et al. (2016); Derczynski et al. (2017); Pérez-Rosas et al. (2018); Shu et al. (2020) new datasets are introduced.

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, which are related to fact checking applications (e.g., fake news, fact checking based on table data (see TABFACT in Section 6)) have been organized. Specifically, in the context of FEVER, a second shared task (**FEVER 2.0**) (Thorne et al., 2019) 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** (Wadden et al., 2020)[7]. 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 the 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). Based on the **TABFACT** dataset (described in Section 6), where the goal is to verify the validity of a claim based on Wikipedia tables, a competition has been also organized[8] (Chen et al., 2019). In the same line, the shared task SemEval 2021 (Task 9) **SEM-TAB-FACT**[9] aims at identifying whether a table supports a given claim and providing also the required evidence for that. The **Fake News Challenge**[10] 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.

[7] https://scifact.apps.allenai.org/leaderboard

[8] https://competitions.codalab.org/competitions/21611

[9] https://sites.google.com/view/sem-tab-facts/

[10] http://www.fakenewschallenge.org/

8 Explainable Fact Checking Models

Although explainable models is an area of increasing interest in NLP (Wallace et al., 2019; Liu et al., 2019a), there are only few studies that focus on explaining the outcome of fact checking models. Specifically, Atanasova et al. (2020) exploit the use of a joint architecture that simultaneously generates explanations using the extractive summarization model of Liu and Lapata (2019) and predicts the veracity of a claim using a classification layer. Similar to the previous work, the work of Stammbach and Ash (2020) relies also into abstractive summarization techniques for generating explanations. In another work, Ahmadi et al. (2019) 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. (2020), they define different metrics for evaluating the quality of alignment between human and machine generated rationales. Nadeem et al. (2019) developed an end-to-end system for fact checking, which 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.

9 Bias in the FEVER dataset

As indicated in the work of Schuster et al. (2019), 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 the performance 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, in the work of Schuster et al. (2019), they create a new test dataset the so-called “SYMMETRIC TEST SET” and they introduce regularization methods to improve the performance on the new test set. In the same line of research, Thorne and Vlachos (2020) also rely on regularization methods to improve the performance on the SYMMETRIC dataset. In another work, Karimi Mahabadi et al. (2020) proposed the use of product of experts (Hinton, 2002) and focal loss (Lin et al., 2017) in order to mitigate the bias in NLI systems and they also report results on the FEVER SYMMETRIC test set. Finally, the quality of the dataset is also discussed in the work of Derczynski et al. (2020), where two metrics that ensure the quality of the annotation of the FEVER dataset, are proposed.

10 Conclusion

In this paper we focus 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 is 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. In addition, 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.

References

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*.

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)*, pages 85–90.

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)*.

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*, pages 7352–7364, Online. Association for Computational Linguistics.

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)*, pages 4677–4691.

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.

Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018a. Adversarial training for multi-context joint entity and relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2830–2836.

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

Samuel 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*, pages 632–642.

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)*, pages 127–131.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017a. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017b. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668.

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*.

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)*, pages 69–78, Hong Kong, China. Association for Computational Linguistics.

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*, pages 670–680.

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*, pages 2475–2485.

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):1–4.

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)*, pages 42–46, Online. Association for Computational Linguistics.

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)*, pages 69–76, Vancouver, Canada. Association for Computational Linguistics.

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)*, pages 4171–4186.

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. *Transactions of the Association for Computational Linguistics*.

T. H. Do, X. Luo, D. M. Nguyen, and N. Deligiannis. 2019. Rumour detection via news propagation dynamics and user representation learning. In *2019 IEEE Data Science Workshop (DSW)*, pages 196–200.

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

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)*, pages 1–6.

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)*, pages 103–108.

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. *Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum*.

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*, pages 1923–1933.

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*, pages 8593–8606, Online. Association for Computational Linguistics.

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)*, pages 150–155.

Christopher Hidey and Kathleen 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)*, pages 1756–1767.

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

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

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*, pages 1373–1378.

Mayank Jobanputra. 2019. Unsupervised question answering for fact-checking. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, pages 52–56, Hong Kong, China. Association for Computational Linguistics.

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*, pages 8706–8716, Online. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of International Conference on Learning Representations*.

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

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)*, pages 36–41, Online. Association for Computational Linguistics.

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. *arXiv preprint arXiv:2005.11401*.

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*, pages 2980–2988.

Hui Liu, Qingyu Yin, and William Yang Wang. 2019a. Towards explainable nlp: A generative explanation framework for text classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5570–5581.

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)*, pages 3721–3731.

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

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*, pages 7342–7351.

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)*, pages 156–160.

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*, pages 2561–2571.

Christopher Malon. 2018. Team papelo: Transformer networks at fever. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 109–113.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose 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*, pages 55–60.

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

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)*, pages 1105–1116.

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)*, pages 78–83.

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)*, pages 1391–1400.

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)*, pages 1–7, Online. Association for Computational Linguistics.

Yixin Nie, Haonan Chen, and Mohit Bansal. 2019a. Combining fact extraction and verification with neural semantic matching networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6859–6866.

Yixin Nie, Songhe Wang, and Mohit Bansal. 2019b. 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), pages 2553–2566.

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)*, pages 20–26, Hong Kong, China. Association for Computational Linguistics.

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*, pages 6086–6093.

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)*, pages 145–149.

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*, pages 2249–2255.

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*, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

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)*, pages 2227–2237.

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*, pages 2173–2178.

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)*, pages 47–51, Online. Association for Computational Linguistics.

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

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*, pages 2931–2937.

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)*, pages 815–823. IEEE.

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)*, pages 3410–3416.

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*.

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):171–188.

Amir Soleimani, Christof Monz, and Marcel Worring. 2020. Bert for evidence retrieval and claim verification. In *European Conference on Information Retrieval*, pages 359–366. Springer.

Dominik Stammbach and Elliott Ash. 2020. e-fever: Explanations and summaries for automated fact checking. In *Conference on Truth and Trust Online*.

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)*, pages 105–109, Hong Kong, China. Association for Computational Linguistics.

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*, pages 1–15. CEUR-WS.

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)*, pages 124–126.

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*, pages 3346–3359.

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*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018a. 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)*, pages 809–819.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018b. The fact extraction and verification (fever) shared task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 1–9.

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)*, pages 1–6, Hong Kong, China. Association for Computational Linguistics.

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*, pages 5998–6008.

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

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*.

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*, pages 7–12.

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)*, pages 422–426.

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)*, pages 1112–1122.

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*.

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*, pages 55–64. ACM.

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*, pages 5753–5763.

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*, pages 105–114.

Wenpeng Yin and Hinrich Schütze. 2018a. Attentive convolution: Equipping CNNs with RNN-style attention mechanisms. *Transactions of the Association for Computational Linguistics*, 6:687–702.

Wenpeng Yin and Hinrich Schütze. 2018b. Attentive convolution: Equipping cnns with rnn-style attention mechanisms. *Transactions of the Association for Computational Linguistics*, 6:687–702.

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)*, pages 97–102.

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*.

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*, pages 6170–6180, Online. Association for Computational Linguistics.

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, pages 892–901.

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):1–36.

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):e0150989.