4-2021

Fact-Checking of Claims from the English Wikipedia Using Evidence in the Wild

Aalok Sathe
University of Richmond

Fact-Checking of Claims from the English Wikipedia Using Evidence in the Wild

Aalok Sathe

Honors Thesis

Under supervision of: Dr Joonsuk Park

Submitted to:

Department of Math and Computer Science
University of Richmond

Richmond, VA 23173, USA

April 2021

Abstract

Automated fact checking is a task in the domain of Natural Language Processing that deals with the verification of claims using evidence. Fact checking is becoming increasingly important as large amounts of human-generated information accumulate online. In the recent past, our society has witnessed large-scale spread of disinformation via the internet that has time and again led to noticeable disruptions in the fabric of society. Fact-checking would help mitigate the spread of disinformation by allowing large magnitudes of content to be automatically evaluated for disinformation.

In this work, we construe and tackle multiple subtasks of fact checking using labeled data from WikiFactCheck-English (Sathe et al., 2020), a dataset of 124k triples consisting of a claim, context and an evidence document extracted from English Wikipedia articles and citations, as well as 34k manually written claims that are refuted by the evidence documents. We provide support vector machine and logistic regression-based baselines, as well as attempt state-of-the-art results using large pretrained transformer-based transfer learning approaches (specifically, BERT) that take our performance from a baseline accuracy of 68% to about 78%. Furthermore, we adapt a novel semi-supervised attention-based multiple-instance learning approach to learn item-level fact verification from document-level labeled data, leading to future possibilities in weakly supervised learning of fact-checking models. We also demonstrate that transfer learning from Natural Language Inference, a sentence-level inference task, leads to the best overall transfer performance in a low-resource data constrained setting, but no overall advantage given sufficient training data.

We demonstrate that claims often require and benefit from more than 1 sentence to support them, and that BERT can learn to attend to multiple evidence sentences to make the correct fact checking inference.

Acknowledgements

The completion of my honors project and the writing of this thesis has been possible thanks to a large number of wonderful and inspiring people. In this section, I will attempt to non-exhaustively recognize some of them.

I thank Dr. Jon Park, my advisor, for his guidance throughout the years as I worked under his mentorship. I thank him for his continued trust and support during highs as well as lows in research over the years.

I am very grateful and lucky to have received mentorship from many professors at UR. I have immensely enjoyed working with my research mentors including Drs. Prateek Bhakta, Taylor Arnold, Heather Russell (Math & CS), Cindy Bukach, Matthew Lowder (Psychology), and Dieter Gunkel (Linguistics). I am also grateful for conversations on academics and life in general from all of them as well as Drs. Jory Denny, Arthur Charlesworth, Jeremy LeCrone, Barry Lawson, and Della Dumbaugh. Thanks to all of them, I have grown as a computer science student during my formative years in college, and have gotten introduced to a rigorous yet fun discipline.

I thank my parents and family for supporting me in my every academic pursuit and encouraging me to keep going. I thank my friends, whose encouragement and camaraderie made this journey enjoyable and easier.

Finally, I thank the Math & CS department, including Ms Kathy Rothert and all the professors, for building a welcoming and inclusive culture, and Dr. Bhakta in particular for leading ICPC practice, Putnam, and weekly board game nights—some of the things that made this department feel like home for four years.

Table of Contents

List of Figures

Chapter 1

Introduction

Fact checking: motivation Fact checking is a problem under the domain of Natural Language Processing (NLP) and Information Retrieval. Fact-checking is a text-based problem, conceived as a classification task. Fact-checking is a popular task, given its applicability to some of the pressing issues faced by societies with high digital penetration and access to the internet. In recent years, large social media platforms on the internet, as well as message boards and public-facing information sources, have witnessed a proliferation of misinformation. Misinformation appears truthful, and is hard for even humans to identify. Misinformation also tends to spread faster than truthful information, and can appear more believable at first glance (Vosoughi et al., 2018). This compounds the hardness of the task for machines, which, thus far, have had limited success in the processing of text involving realistic data requiring world knowledge and common-sense reasoning.

Machine learning approaches to Fact checking Many researchers have worked on various characterizations of the fact checking task. Typical approaches include constructing a dataset based on fact and misinformation content, training classifiers on the data, and evaluating them. In the past, people have extracted such datasets using human fact-checking websites, headlines and ledes of news articles, hand-crafted text, and text extracted from large internet-based knowledge bases including Wikipedia. Approaches include regression, support vector machines, convolutional neural networks, recurrent neural networks, transformer neural networks, and ensemble methods. In more recent approaches, like in any NLP application, a representation of the input is obtained using an intermediate network. This representation may be in the form of word embeddings, recurrent encodings of entire sentences, as well as contextualized embeddings using convolutional networks. These representations are further fed to classifiers, which may range from regression to feedforward neural networks.

In recent years, we have seen the advent of large corpus-based training methods and newer model architectures, most noticeably, the transformer family of architectures. These advancements enable taking advantage of vast amounts of unsupervised text data obtained using the internet, via sources such as Wikipedia and Common Crawl. Using these large corpora, researchers have demonstrated that unsupervised or semi-supervised *pretraining* on unrelated tasks, can lead to significant transfer improvements in general-purpose NLU in many downstream tasks. Such methods have also shown large increases in performance on specific applications, with large pretrained models now playing a key role in the pipelines of many such applications. In general, the current state of the art in Natural Language Understanding has come to be dominated by the practice of utilizing large, unsupervised, pretrained language models *finetuned* to a specific domain.

1.1 Fact Checking Formalism

There are many varieties of the general problem of fact checking that came about over the years, proposed by various researchers and organizations. Typically, fact-checking involves verifying the truth value of a claim or a statement. This verification may be done as a standalone task, i.e., on the basis of the structure of the claim itself (untruthful claims may exhibit certain patterns that are useful for deceiving non-skeptical humans), or using an external knowledge base. The verification may be also done in the context of a particular fact, or assertion, i.e., verification of the truthfulness of a claim assuming a certain other statement were true.

In fact checking, as will be construed in the rest of this manuscript, a *claim*, c is a textual string, usually consisting of one meaningful sentence or statement making an assertion about something. Occasionally, the claim may be longer than a single sentence, or simply a meaningful phrase, rather than a complete sentence. The *context* of a claim is the text preceding the claim in the original document the claim was extracted from. The context is typically more than one sentence, but less than a paragraph. The task involves the *evidence document*, $E_c = (e_1, e_2, ..., e_{|E_c|})$, which is a document containing justification for either supporting or refuting the claim. Here, $e_1, e_2, ...$ are individual sentences of the evidence document. An *instance* of fact checking is a collection of a claim, context, evidence, and a gold label. The *gold label* associated with an instance is the true status of the claim with respect to the evidence: whether it is *supported* or *refuted*.

1.2 Research Contribution

In this work, we build on past work in fact-checking and more generally in natural language understanding (NLU) by (1) tackling fact-checking using a new dataset consisting of 124k+ claims, context, and evidence extracted using the English Wikipedia; (2) extending baseline performance using state-of-the-art methods; (3) implementing a novel dot-product attention mechanism to learn item-level inference using document-level labels; and (4) utilizing context in an information retrieval component. In doing so, we provide competitive empirical results and a novel approach to learning from partially-labeled data in a semi-supervised manner. To our knowledge, our dataset is constructed in a novel manner to tackle existing issues with datasets constructed in the past. Our extension of baseline results uses transformer language models, making it one of the early uses of transformers in fact-checking. Furthermore, our novel approach provides new avenues of dataset construction and methods of training from massive amounts of unlabeled data.

We provide a competitive result on the WikiFactCheck-English dataset using transformer-based language models (BERT). We improve on multiple subtasks of the fact checking task, including the sentence retrieval task using contextualized semantic similarity measures for retrieval, as well as the inference subtask. We improve on inference in the single sentence as well as generalized case, by allowed for semi-supervised learning of unlabeled sentence-level data from document-level annotations. We investigate the significance of transfer learning from related tasks to Fact Checking, and investigate the use of context of a claim in performing fact checking.

1.3 Outline

In the next section (Background), we will go into the task as well as some background information in more depth. In the section after that (Related Works), we provide an overview of past research on fact checking as it relates to our contribution. We include a summary of other approaches, including datasets and methods, and the outcomes. In the Methods section, we provide the technical details of our approach and contribution, including machine learning and neural network methods, as well as the specific adaptation to our task. We explain the specific questions we investigate and how we address them. In the Experiments section, we describe our experimental setup, and describe the results, and how they relate to our question and contribution. Finally, in Discussion, we consider the implications of our approach, and discuss extensions to this work.

Chapter 2

Background

2.1 Fact Checking Subtasks

Fact-checking is a complex task involving many moving parts. Thanks to this, many subtasks that are otherwise independently studied and tackled can be formulated as subtasks of fact-checking. Consequently, fact checking can benefit from research in these areas, and the subtasks can find a practical application domain. Furthermore, research on integrating various subtasks can lead to futher advancements within these subtasks. Part of work attempts to integrate certain subtasks in a novel way and demonstrate the effectiveness of them in fact checking. From common knowledge, as well as observation, we find that typically 1-3 sentences, $e_i, e_j, e_k \in E_c$ lend sufficient evidence to support or refute the claim c. However, the typical evidence document E_c of a claim contains hundreds of sentences. The crux of the task, then, is to identify few sentences in E_c that may be used to decide the truth value of claim c.

1 Document Retrieval/Information Retrieval subtask A subtask of Fact Checking is Document Retrieval/Information Retrieval (IR). In IR, we are trying to retrieve a subset of optimal documents D' from a very large set of possibilities D with respect to a query q, such that the relevance of D' given q is maximized,

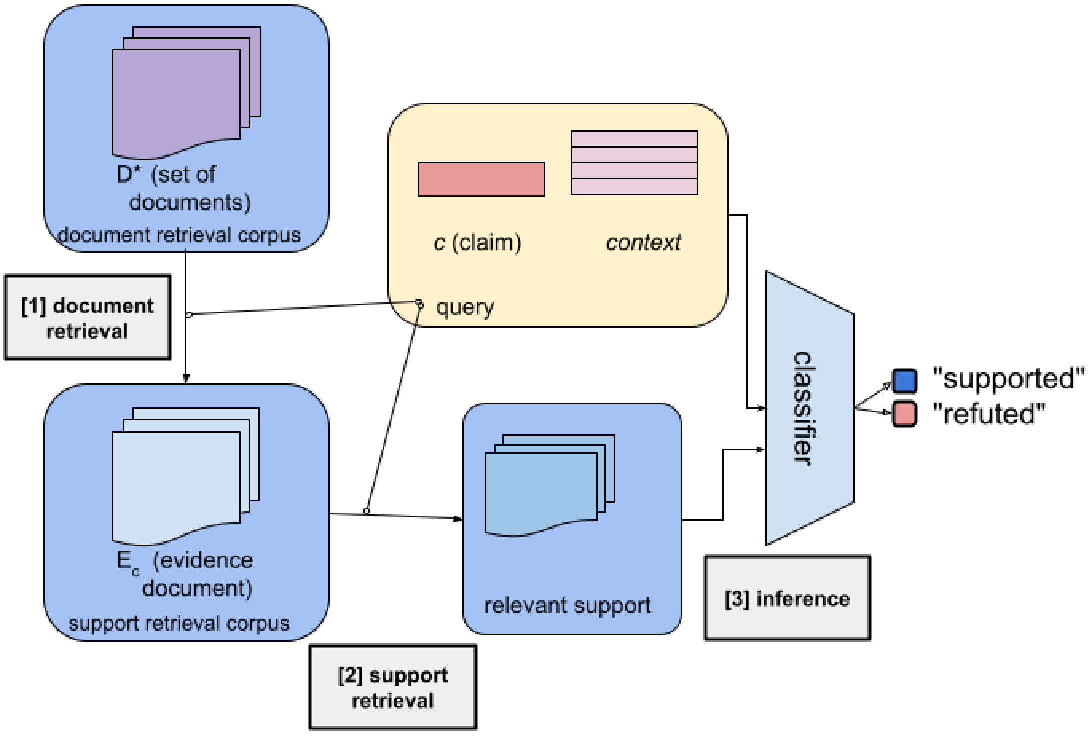

Figure 2.1: High-level depiction of the WikiFactCheck pipeline

according to some ranking metric. IR is a subtask of fact-checking: in order to garner appropriate evidence to determine the support for a claim, one must search for evidence-containing documents. In our framing of the task, there is a set of evidence documents $E^\star$. A step in the task would be to determine the appropriate document $E_c \in E^\star$ given a claim c. Fact checking in the wild faces a choice of millions of documents from all over the internet, any subset of which could lend support to a claim. The crucial execution of this task is a necessity for the success of other subtasks building on top of it. In this work, we restrict ourselves to a single document to draw upon.

2 Sentence Retrieval/Support Retrieval subtask Given a claim c and an evidence document E_c, there can be many possible sentences $e_1, ... \in E_c$ that may or may not lend evidence to support or refute c. Sifting through these to arrive at the correct few is a challenge, and is crucial to next steps in the fact checking task. Whereas more than one sentence e_a, e_b, e_c may be necessary to make the determination of the truth of c, the structure in which such an inference may be made is not fixed, and can involve a multi-step argument in terms of the e_is. For instance, it may be that $e_a \implies e_b$; $e_a \wedge e_c \implies c$. On the other hand, it could also be the case that $e_a \implies e_b \implies e_c \implies c$. A simpler case could be that $e_a \wedge e_b \wedge e_c \implies c$. There are many possible ways to structure an argument leading to c making it hard to extract sentences from E_c. The challenge is then to extract the appropriate sentences that will allow the construction of a valid argument in favor or opposition of c. Because textual writing is generally topically organized, E_c is likely to have many topically similar sentences, not all of which may be relevant. Therefore, sentence retrieval may benefit from the semantic representation of sentences.

3 Natural Language Inference: the NLI subtask A subtask of Fact Checking is Natural Language Inference (NLI). In NLI, we have a premise (p) and a hypothesis (h), and the task is to determine the truth value of h with respect to p. NLI involves assigning labels "entailment", when $p \implies h$, "contradiction", when $p \implies \neg h$, or "neutral", when $\neg(p \implies h)$. To extend the inference task from its single-premise version to be compatible with the dynamic nature of fact checking, we allow for multiple premises, $p_1, ..., p_n$, to either support or refute h, which in our case will be the same as the claim c. As mentioned in a prior part, the inference subtask must make a proper inference regardless of the argument structure needed by the premises.

In this work, we focus on WikiFactCheck-English, a dataset consisting of claims, context, and evidence triples extracted from the English Wikipedia, and a corresponding formulation of the fact checking task involving support retrieval and natural language inference (NLI) (Sathe et al., 2020).

2.2 Transformers

2.2.1 Motivation

Until recently, recurrent neural network (RNN) and its variant long short-term memory (LSTM) models were the state-of-the-art and standard for natural language understanding thanks to their ability to encode arbitrarily long sequences (with truncated backpropogation as necessary). However, RNN-based encoder-decoder architectures rely on the use of latent vectors leading to an information bottleneck and the inability to maintain long-distance dependencies due to the limitation of the latent encoding vector. For this reason, recent uses of RNN-based encoder-decoder architectures utilized *attention*, i.e., a weighted sum of all previously computed hidden representations over the input sequence to obtain a latent representation at each decoding timestep, or at classification-time depending on the task.

2.2.2 Self-Attention and Contextualized Embeddings

An extension of attention over an RNN encoder is *self-attention*, which entails getting rid of the time-dependant encoding of the input sequence. Instead, a self-attention mechanism computes the representation of an input token by averaging over all input tokens queried by itself (Vaswani et al., 2017). An advantage of this is the generalizability in terms of architecture it affords. Self-attention allows all inputs to be interpreted in context, and relaxes constraints on information content in latent vectors by doing away with them. Self-attention additionally enables wider parallelism allowing for better and more efficient utilization of modern GPU architectures. This allows scaling along data, significantly reducing training times. A simple illustrative example of this is the dot-product attention.

Say we have input tokens $T_1, T_2, \ldots$. Let the embeddings be given by $\mathbf{emb}(\cdot)$, parameterized by an embedding layer. Then, the attention corresponding to

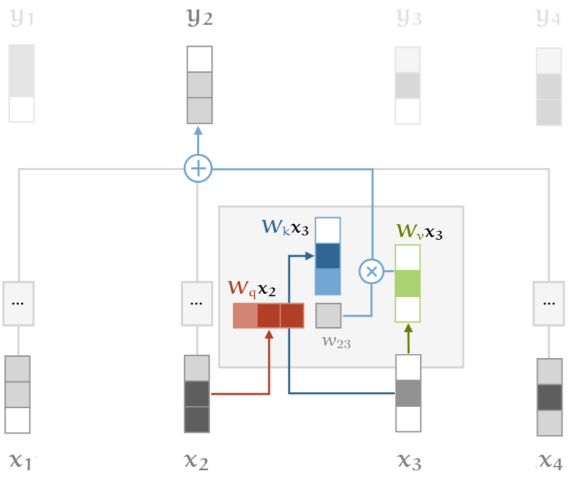

Figure 2.2: Figure taken from Bloem (2019) illustrating self-attention mechanism in transformers. x_i represent input embeddings; y_i are contextualized embeddings.

tokens i, j will be given by:

$$a'_{i,j} = \mathbf{emb}(\mathbf{T_i}) \cdot \mathbf{emb}(\mathbf{T_j}) \tag{2.1}$$

Now in order to compute the contextualized representation of T_is, first we compute a **softmax** over the attention values to obtain a normalized collection of weights for our weighted sum.

$$a_{i,\star} = \mathbf{softmax}(a'_{i,\star}) \tag{2.2}$$

Then, each contextualized vector $\mathbf{x}$ is simply a weighted average of the input embeddings.

$$\mathbf{x}_i = \sum_{j=1,2,\ldots,n} a_{i,j} \cdot \mathbf{emb}(T_j) \tag{2.3}$$

A more nuanced form of attention than dot-product attention is *self-attention*, which utilizes key, query, value (K, Q, V) transformations of input embeddings to compute attention weights, illustrated in figure 2.2.

The self-attention mechanism is applied multiple times in different *attention heads* to allow for the possibility of picking up on nuanced relations between

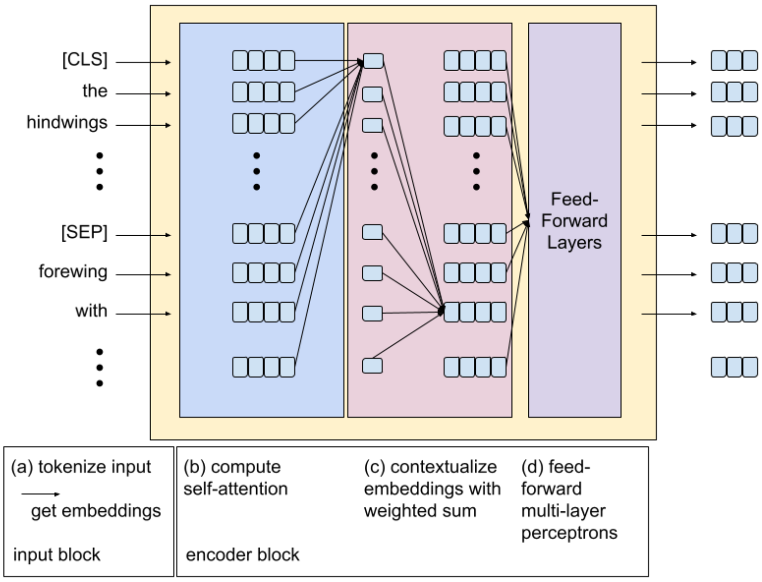
[CLS]
the
hindwings
[SEP]
forewing
with
Feed-Forward Layers
(a) tokenize input
get embeddings
input block
(b) compute self-attention
(c) contextualize embeddings with weighted sum
(d) feed-forward multi-layer perceptrons
encoder block

Figure 2.3: Schematic illustration of BERT in action with sample input from WikiFactCheck-English: a claim (c) and a candidate support sentence (e_i). BERT computes contextualized representations of each token from c and e_i, including special tokens `[CLS]`, `[SEP]`.

input tokens. Attention heads are followed by multi-layered perceptrons (MLPs) to complete an encoder block. A transformer consists of multiple encode blocks followed by a number of decoder blocks. We will skip the implementation details of the above-mention concepts, as well as a description of the decoder block, as that will not be necessary for our purposes.

In this paper, we use Bidirectional Encoder Representations from Transformers (BERT) models (Devlin et al., 2018) that have been pretrained on Masked Language Modeling and Next Sentence Prediction tasks using data from the datasets BookCorpus and WikiText. We utilize the easy-to-extend open-source implementations provided by Huggingface (Wolf et al., 2020), and modify to support our architecture. Figure 2.3 sketches a schematic of the BERT model architecture used in this paper. BERT accepts tokenized input with special tokens. Among these are `[CLS]`, a dummy token used for pooled representation fed into a classifier, and `[SEP]`, a separator token placed between input sequences.

2.2.3 Transfer Learning and Finetuning

Many instances of using pretrained BERT to finetune on a downstream task have shown promising performance, including attaining state-of-the-art results on a large number of tasks. It is systematically seen than the approach of starting with a pretrained BERT, finetuning it to a task, and possibly further finetuning it gives promising results (Rogers et al., 2021). In addition to introducing BERT in the WikiFactCheck pipeline, we also sought to identify the effects of various intermediate task finetuning, and compare the results between various tasks, including, no task, i.e., the vanilla pretrained BERT.

Chapter 3

Related works

In this section we review various relevant resources and machine learning approaches to fact-checking.

3.1 Resources and Datasets related to Fact-checking

Fact checking was introduced as a task close to the year 2014 to overcome obvious limitations with manual fact-checking in various websites. Researchers did this by creating a dataset consisting of statements made by prominent persons. The ratings (labels) were judged by journalists, and URLs to evidence was provided (Vlachos and Riedel, 2014). The procedure above supplied a somewhat domain-specific dataset where the statements are misleading by design.

A newer dataset, FEVER, was built using introductory sentences from Wikipedia. Annotators mutated the sentences to generate positive as well as negative sentences. Participants also provided a sentence-level annotation.

A common limitation of these efforts is the limited size and the homogeneous and synthetic nature of the data thanks to the way it was sourced.

In this work, we use WikiFactCheck-English (Sathe et al., 2020), which was designed to address some such issues by creating a large dataset of real-world claims and evidence.

NLI is a subtask of our fact-checking pipeline, as illustrated in figure 2.1. In the following, we provide an overview of work relating to NLI.

In NLI one must identify relationships between two text sequences. PASCAL intiated the Recognizing Textual Entailment (RTE) challenge in 2005 (Dagan et al., 2010). The challenge highlighted the common underlying paradigm across many kinds of tasks, which may be broadly be termed together as Natural Language Understanding (NLU). In these tasks, there is variation in the kind of semantic expression. However, being able to tackle any one task in theory should mean being able to tackle any other task. As we will see later, crucial to fact-checking is this semantic variability of expression through text.

For a comprehensive review, see Sathe et al. (2020).

3.2 Machine Learning Approaches to Fact-checking

Zhong et al. (2020) reiterate that fact-checking is a complex task where more than one sentence may support or reject a claim. It is therefore important to consider multiple possible bases for the inference step. In the mentioned work, the authors take an approach of created semantic graphs and reason over these structures to address the FEVER task (Thorne et al., 2018). Whereas we will use an approximation of the argument structure of inference, it is worthwhile to note the complexity of inference that may be required.

Soleimani et al. (2020) tackled the FEVER task using a two-step pipeline. The authors employed BERT in the evidence retrieval component to rank the evidence by relevance to the claim to be verified. The authors also employed BERT in the second component to use the retrieved evidence to verify a claim.

Lee et al. (2020) took an interesting approach to the otherwise traditional pipeline of fact-checking in FEVER- and WikiFactCheck-like tasks. The authors considered verifying claims independent of the available evidence, which in the case of FEVER would be a subset of the data used in the pretraining of BERT. The authors queried BERT as though a knowledge base and used its masked predictions to decide if the claim was true. Whereas it is a novel approach, in practice, it doesn't provide a promising performance gain. On WikiFactCheck-English, such an approach did not go beyond baseline approaches previously reported.

In this work, we add to the currently limited but growing body of work of using transformer-based models, particularly BERT, to perform fact checking.

Chapter 4

Methods

In this section, we describe our approach to Fact Checking as it relates to our dataset, WikiFactCheck-English. As previously mentioned, we specifically address the subtasks of support retrieval and natural language inference.

4.1 Fact Checking Pipeline

We describe the detailed pipeline for the present work in figure 4.1. The following parts describe components of the pipeline.

4.1.1 Sentence Retrieval

Recall that the sentence retrieval/support retrieval subtask of fact-checking involves retrieving sentence(s) from the evidence document E_c that lend support to or refute a claim c. In this paper, we focus on retrieving the top k sentences $(e_1, e_2, ..., e_k) \in E_c$.

Baseline

We explored various approaches of retrieving the top 1 sentence using simple textual similarity. These included: Levenshtein distance (LD) and Cosine similarity. Empirically, LD worked better for sentence retrieval (Sathe et al., 2020). In the absence of true annotations of supporting text, we are forced to compare approaches in this subsection using only the overall results.

Current approach

To improve upon the baseline, we utilized Sentence-BERT or SiameseBERT (SBERT) (Reimers and Gurevych, 2019). SBERT is a model architecture for computing representations over sentences rather than tokens. A flavor of SBERT is trained to jointly compute representations of sentence pairs, and a classifier on top of it is trained to compute sentence pair semantic textual similarity as a scalar. Because our claims and evidence are drawn from distinct sources that may be dissimilar in their style and choice of words, we believe that an approach involving semantics would work better as compared with a purely textual approach (such as LD or cosine similarity). In order to compute semantic similarity, we use a pretrained SBERT architecture finetuned to the `sts-b` (semantic textual similarity) corpus part of a popular NLU benchmark for generalized language understanding and evaluation, GLUE (Wang et al., 2018).

As illustrated in figure 4.1, we rank sentences in E_c by similarity with $[context|c]$ (context concatenated with the claim c). We pick the top k sentences, where $k = 1, 3, 5$. In the case of $k = 1$, we have a direct comparison with the baseline approach from Sathe et al. (2020).

4.1.2 Natural Language Inference

As shown in figure 4.1, the next step in the WikiFactCheck pipeline is to perform classification on retrieved support and the claim and context.

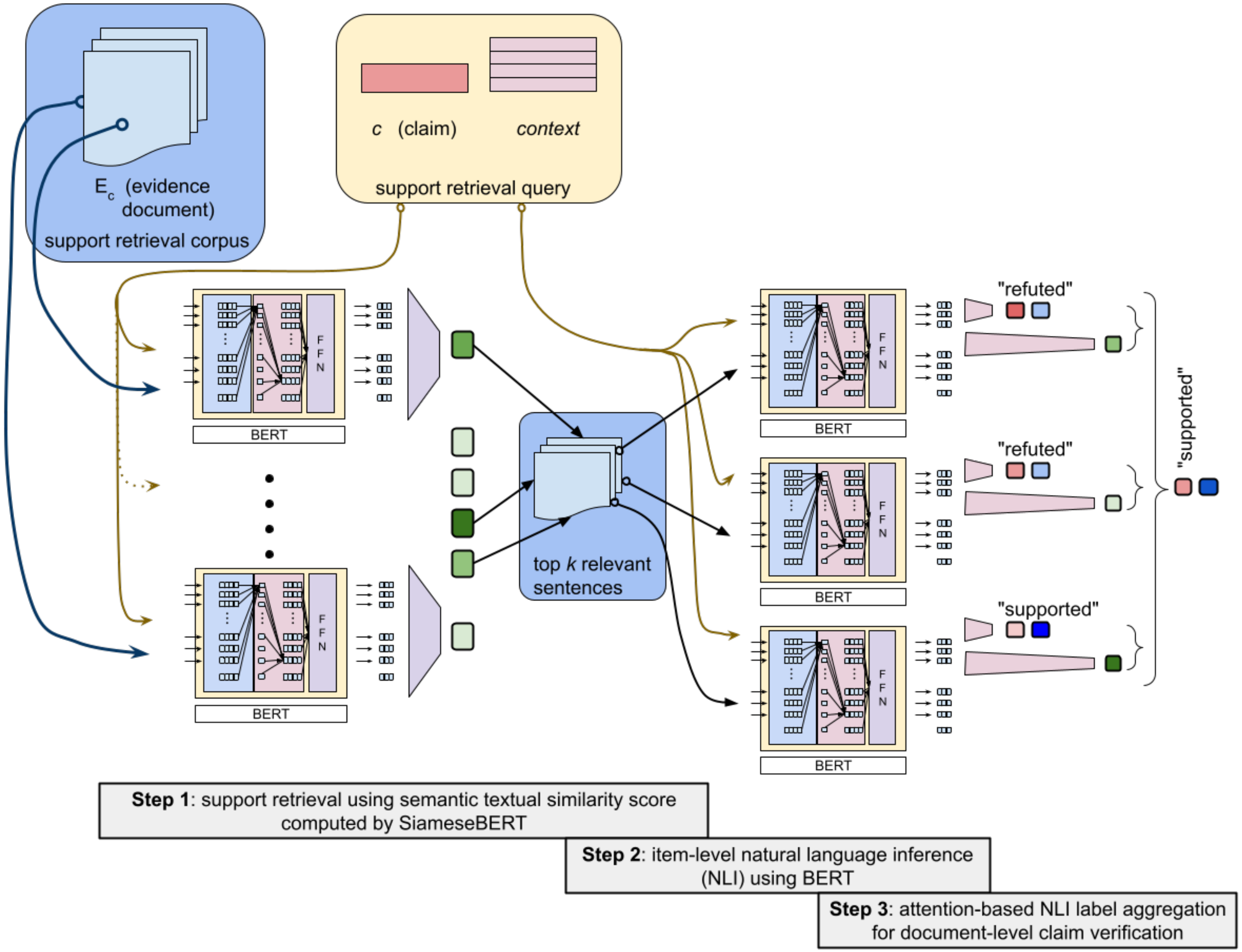
E_c (evidence document)
support retrieval corpus
c (claim)
context
support retrieval query
BERT
F F N
top k relevant sentences
"refuted"
"refuted"
"supported"
"supported"
Step 1: support retrieval using semantic textual similarity score computed by SiameseBERT
Step 2: item-level natural language inference (NLI) using BERT
Step 3: attention-based NLI label aggregation for document-level claim verification

Figure 4.1: WikiFactCheck pipeline outlining the tasks addressed in this paper. Darker shades represent numerically higher weights or values in case of model outputs.

Baseline

Our first baseline is as reported in Sathe et al. (2020) combined with the baseline approach to support retrieval. This baseline utilizes support vector machines (SVM) and logistic regression (LR) using bag-of-words and special handcrafted features on the top 1 relevant evidence sentence together with the claim. With SVMs, we achieved 66% accuracy and 69.6% F1. With LR, we obtained 68% accuracy and 70.8% F1. It is noteworthy that these performances are despite an approximating assumption: that the top-1 most textually similar sentence contains sufficient information to make an inference about the claim. To relax this assumption, we allow for multiple-sentence inference in the current work.

Current approach

In this paper, we will break this step down into two parts: natural language inference (NLI) and attention-based aggregation. Because our dataset only contains ground-truth labels at the document-level, we do not possess item-level labels for NLI with each sentence from E_c. We therefore utilize the multiple instance learning (MIL) technique (Angelidis and Lapata, 2018; Ilse et al., 2018) to learn item-level labels in a semi-supervised manner by aggregation over multiple labels and computing loss at the document level.

Let **BERT** denote the pretrained BERT model to use.

Let $\mathbf{BERT}(\texttt{[CLS]}, T_1, T_2, \dots)$ represent the contextualized representations of input tokens $\texttt{[CLS]}, T_1, T_2, \cdots$. We initialize a binary ("supported"/"refuted") NLI classifier over the contextualized inputs such that:

$$\mathbf{y} = ReLU\left(W \cdot \mathbf{BERT}(\texttt{[CLS]}, ...)_0 + b_W\right) \tag{4.1}$$

where $\mathbf{y}\langle,\rangle$ is the output of an MLP from the `[CLS]` token of the input sequence. The `[CLS]` token is a special token whose representation is meant to capture relevant features of the input for classification (Devlin et al., 2018).

Notice that for a claim c and top-k relevant support $e_1, e_2, \ldots, e_k$, we will have k input pairs $\langle e_i, [context|c] \rangle$ or $\langle e_i, c \rangle$ (either with or without context). For each such pair, we will have an NLI prediction $\mathbf{y}_i$. However, we only have the ground truth $\hat{y}$ corresponding to $\langle E_c, c \rangle$. Therefore, we will use another MLP to transform the same representation used for NLI to compute attention over the items.

$$\mathbf{a}' = Sigmoid\,(V \cdot \mathbf{BERT}(\texttt{[CLS]}, ...)_0 + b_V) \tag{4.2}$$

Similar to the concept of attention as used within transformers, we use the attention thus computed to aggregate outputs like so:

$$\mathbf{a} = \mathbf{softmax}(\mathbf{a}') \tag{4.3}$$

Here, $\mathbf{a}_i$ is the attention weight for aggregating the predictions for $\langle c, e_i \rangle$.

Then our aggregated document-level predicted label is

$$\mathbf{y}^\star = \mathbf{a} \otimes \mathbf{y} \tag{4.4}$$

where $\mathbf{y}^\star$ is a two-item vector containing predictions for two labels. Loss is computed using Cross Entropy loss with the appropriate ground truth label. In case $k = 1$, the attention weight for the single prediction is trivially 1, and this automatically reduces to a vanilla NLI case without any need for backpropogation.

4.1.3 Implementation and Experimentation

We built off of the implementation given by the open-source transformers library 'Huggingface' (Wolf et al., 2020) and PyTorch. We used 'Weights and Biases' to document our experimental setup and hyperparameters to understand what the most important factors were during a pilot experiment (Biewald, 2020).

Chapter 5

Experiments and Results

In this section, we outline the questions at hand based on our set up in the methods section. We then outline how we perform experiments to address these questions. Finally, we discuss results.

1. Do claims draw on more than one sentence to be accurately classified?
   (a) Can we learn aggregate fact checking without item-level annotations?
   (b) How many supporting sentences are a good amount of support?
2. Does using context help classify a claim?
3. Does intermediate task transfer learning help fact checking?
   (a) What tasks are helpful?

5.1 Experimental Setup

Data We initially used a small subset of training data to understand the behavior of the model and importance of various factors. Our initial experimentation used fewer than 2k examples at training time (out of 24k in the training set). After hyperparameter tuning, we used at mos 10k examples to train the pipeline.

Our pipeline was constrained by the time taken due to hardware-related issues, and so we were unable to scale to the full 24k examples available for training.

Intermediate task finetuning We experiment with `bert-base-uncased`, i.e., the simple pretrained variant of BERT (in the 'base' size). This variant is the vanilla variant without any *finetuning* to an NLU task. All BERTs used have been pretrained on the same large unsupervised corpora. When finetuning occurs, it is performed in addition to pretraining (Pruksachatkun et al., 2020). We also use BERT models finetuned on MultiNLI.

Using context We experimented with using context along with claim in the NLI step by concatenating context and claim together. However, pilot results revealed that using context consistently hampered performance and the model was unable to reach the level of performance of claim-only NLI. We elected to not use context in our extended experiments.

Similar sentences We used $k \in \{1, 3, 5\}$ for the aggregate NLI step.

Training procedure We used learning rates $10^{-5}, 50^{-5}, 10^{-4}$ during hyperparameter search with Adam optimizer and linear learning rate decay. We settled on 10^{-5} as the optimal learning rate for later use. We use a constant 50 warmup steps based on initial pilot experimentation. We train for 3 epochs and update gradients every 4 steps, picked based on pilot experiments with $1, 4, 8, 16$ update steps. For evaluation, we compute accuracy, F1, precision, and recall on the training and validation sets.

In addition to computing typical metrics, we computed correlations and relative parameter importance using linear models, with help from (Biewald, 2020).

5.2 Results

Table 5.1 shows evaluation metrics for several runs of the pipeline on validation data. We see at least a 10-point climb across the board compared with results reported in Sathe et al. (2020). This improvement comes after two modifications to the pipeline; (1) using BERT-based sentence similarity for ranking relevant sentences, (2) using BERT-based NLI in an individual and aggregate fashion. Whereas it is not possible to attribute the climb accurately to any single factor, it is likely the combination of changes and new experimentation that has contributed to the increase in performance.

We observe that using MNLI-finetuned BERT affords an advantage early in the training process, but this advantage subsequently disappears as multiple pre-trained BERTs converged to similar accuracy values.

Correlations revealed that accuracy and F1 scores are positively correlated with k across multiple runs. Instances with $k = 3, 5$ tend to outperform single-sentence-support instances. This suggests that WikiFactCheck-English has a non-trivial amount of examples that need more than a one-step single-sentence inference, but might require more nuanced, and multi-step inference. We also witness that using a Multiple Instance Learning and attention-based approach allows us to perform inference over multiple sentences despite not having fine-grained labels to train this inference directly.

We see noisiness in the training process compared with typical GLUE task training (Wang et al., 2018). We believe this may simply be due to the complex nature of our task, and because of our modifications to the typical BERT-training architecture. Nevertheless, overall, we see convergence to a fact checking model.

finetuning model	***k***	**acc**	**f1**	**pre**	**rec**
bert-base-uncased[1]	5	76.2[1]	75.12[1]	80.29[1]	71.83[1]
bert-base-uncased	3	**0.786**	**0.770**	**0.834**	**0.715**
bert-base-uncased	1	0.778	0.757	**0.834**	0.693
aloxatel/bert-base-mnli	5	**0.780**	**0.760**	**0.835**	0.697
aloxatel/bert-base-mnli	3	0.773	0.755	0.819	0.701
aloxatel/bert-base-mnli	1	0.778	**0.760**	0.827	**0.703**

Table 5.1: Results from experiments on extended ($n = 10{,}000$) amount of data. [1]Run did not finish executing due to hardware malfunction.

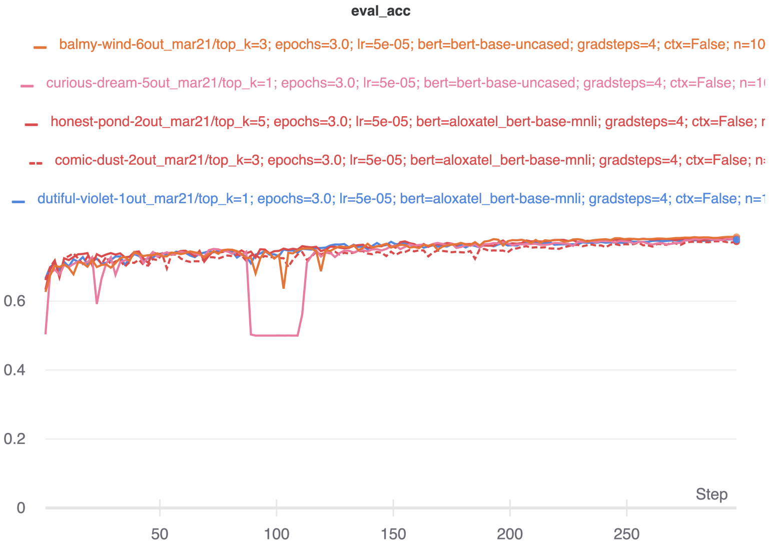

Figure 5.1: Accuracy from several runs on the validation set

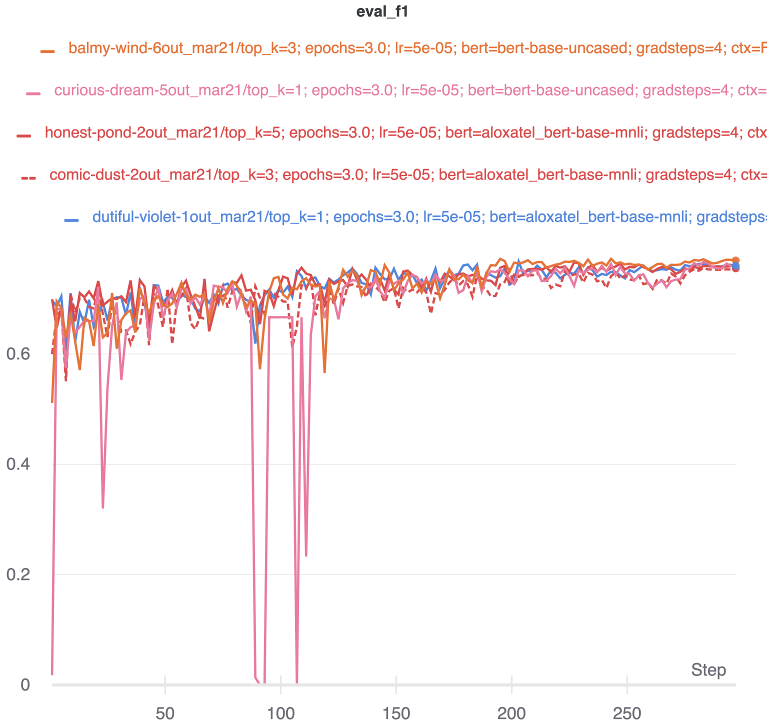

Figure 5.2: F1 from several runs on the validation set

Chapter 6

Conclusion and Future Work

We see that BERT-based pipeline significantly pushes the performance on Wikifactcheck, even with half the data withheld compared to SVM and LR baselines. This highlights the promise of large pretrained language models' use in downstram tasks, and particularly tasks other than NLU, but more nuanced, multi-step tasks as well, for example, fact checking.

Beyond simply commenting on an overall performance gain, we also saw some interesting specific results. We see that $k > 1$ helps inference. This is an interesting finding, and one that matches intuition, that a single sentence likely has more than one sentence as their actual support. It may also indicate that BERT is able to pick up on argument structure beyond single sentences with proper set up and training. However, common intuition also says that it is likely not too many sentences that will form an argument for a claim. To investigate whether a large k would hurt rather than help fact checking NLI step, we would have to perform more experiments. However, until then, this result is promising, and suggests that using BERT may allow for more experimentation in semi-supervised and unsupervised settings, since datasets need not be as granualar. This might also encourage newer dataset creation that relies less on annotators.

We observe that intermediate task training helps, but significantly so with less data. When we introduced an order of magnitude more data, this advantage disappeared at the end of training. We believe the advantage conferred is likely to be temporary and more starkly visible in a resource-constrained manner. This is in line with existing findings that intermediate task training is helpful towards similar tasks, and greatly improves few-shot learning. In the future, we would like to explore more tasks for intermediate fine-tuning.

With regards to context, it was our intuition that using context should provide an advantage to perform inference. However, using context adds several sentences to the claim, and that may be throwing off BERT at the NLI stage. To make comparison on equal footing, in the future, we will compare with a BERT that has been given a gibberish context as opposed to actual relevant context to compensate for length.

Bibliography

Angelidis, S. and Lapata, M. (2018). Multiple instance learning networks for fine-grained sentiment analysis. *Transactions of the Association for Computational Linguistics*, 6:17–31.

Atanasova, P., Nakov, P., Màrquez, L., Barrón-Cedeño, A., Karadzhov, G., Mihaylova, T., Mohtarami, M., and Glass, J. (2019). Automatic fact-checking using context and discourse information. *Journal of Data and Information Quality (JDIQ)*, 11(3):1–27.

Biewald, L. (2020). Experiment tracking with weights and biases. Software available from wandb.com.

Bilu, Y., Hershcovich, D., and Slonim, N. (2015). Automatic claim negation: why, how and when. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 84–93.

Bloem, P. (2019). Transformers from scratch.

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

Burger, J. and Ferro, L. (2005). Generating an entailment corpus from news headlines. pages 49–54. Association for Computational Linguistics.

Ciampaglia, G. L., Shiralkar, P., Rocha, L. M., Bollen, J., Menczer, F., and Flammini, A. (2015). Computational fact checking from knowledge networks. *PloS one*, 10(6):e0128193.

Dagan, I., Dolan, B., Magnini, B., and Roth, D. (2010). Recognizing textual entailment: Rational, evaluation and approaches–erratum. *Natural Language Engineering*, 16(1):105–105.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Esuli, A. and Sebastiani, F. (2006). Sentiwordnet: A publicly available lexical resource for opinion mining. In *LREC*, volume 6, pages 417–422. Citeseer.

Ferreira, W. and Vlachos, A. (2016). Emergent: a novel data-set for stance classification. pages 1163–1168.

Gardner, M., Grus, J., Neumann, M., Tafjord, O., Dasigi, P., Liu, N., Peters, M., Schmitz, M., and Zettlemoyer, L. (2018). Allennlp: A deep semantic natural language processing platform. *arXiv preprint arXiv:1803.07640*.

Ghuge, S. and Bhattacharya, A. (2014). Survey in textual entailment.

Gururangan, S., Swayamdipta, S., Levy, O., Schwartz, R., Bowman, S. R., and Smith, N. A. (2018). Annotation artifacts in natural language inference data.

Hickl, A., Williams, J., Bensley, J., Roberts, K., Rink, B., and Shi, Y. (2006). Recognizing textual entailment with lcc's groundhog system. volume 18.

Hidey, C., Chakrabarty, T., Alhindi, T., Varia, S., Krstovski, K., Diab, M., and Muresan, S. (2020). Deseption: Dual sequence prediction and adversarial examples for improved fact-checking. *arXiv preprint arXiv:2004.12864*.

Honnibal, M. and Montani, I. (2017). spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Ilse, M., Tomczak, J. M., and Welling, M. (2018). Attention-based deep multiple instance learning.

Khot, T., Sabharwal, A., and Clark, P. (2018). Scitail: A textual entailment dataset from science question answering. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Lee, N., Li, B. Z., Wang, S., Yih, W.-t., Ma, H., and Khabsa, M. (2020). Language models as fact checkers? *arXiv preprint arXiv:2006.04102*.

Loper, E. and Bird, S. (2002). NLTK: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics*, pages 63–70.

Marcu, D. and Echihabi, A. (2002). An unsupervised approach to recognizing discourse relations. In *ACL*, pages 368–375.

Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Parikh, A. P., Täckström, O., Das, D., and Uszkoreit, J. (2016). A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933*.

Park, J. and Cardie, C. (2012). Improving implicit discourse relation recognition through feature set optimization. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, SIGDIAL '12, pages 108–112, Stroudsburg, PA, USA. Association for Computational Linguistics.

Pérez-Rosas, V., Kleinberg, B., Lefevre, A., and Mihalcea, R. (2017). Automatic detection of fake news. *arXiv preprint arXiv:1708.07104*.

Plummer, B. A., Wang, L., Cervantes, C. M., Caicedo, J. C., Hockenmaier, J., and Lazebnik, S. (2015). Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649.

Pruksachatkun, Y., Phang, J., Liu, H., Htut, P. M., Zhang, X., Pang, R. Y., Vania, C., Kann, K., and Bowman, S. R. (2020). Intermediate-task transfer learning with pretrained models for natural language understanding: When and why does it work? *arXiv preprint arXiv:2005.00628*.

Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Rogers, A., Kovaleva, O., and Rumshisky, A. (2021). A Primer in BERTology: What We Know About How BERT Works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Sathe, A., Ather, S., Le, T. M., Perry, N., and Park, J. (2020). Automated fact-checking of claims from wikipedia. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 6874–6882.

Sharma, K., Qian, F., Jiang, H., Ruchansky, N., Zhang, M., and Liu, Y. (2019). Combating fake news: A survey on identification and mitigation techniques. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(3):21.

Soleimani, A., Monz, C., and Worring, M. (2020). Bert for evidence retrieval and claim verification. In *European Conference on Information Retrieval*, pages 359–366. Springer.

Thorne, J., Vlachos, A., Christodoulopoulos, C., and Mittal, A. (2018). FEVER: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355.*

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need.

Vlachos, A. and Riedel, S. (2014). Fact checking: Task definition and dataset construction. pages 18–22.

Vosoughi, S., Roy, D., and Aral, S. (2018). The spread of true and false news online. *Science*, 359(6380):1146–1151.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2018). Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461.*

Wang, W. Y. (2017). ”liar, liar pants on fire”: A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648.*

Williams, A., Nangia, N., and Bowman, S. R. (2017). A broad-coverage challenge corpus for sentence understanding through inference.

Wolf, T., Chaumond, J., Debut, L., Sanh, V., Delangue, C., Moi, A., Cistac, P., Funtowicz, M., Davison, J., Shleifer, S., et al. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Yin, W. and Roth, D. (2018). Twowingos: A two-wing optimization strategy for evidential claim verification. *arXiv preprint arXiv:1808.03465.*

Zhong, W., Xu, J., Tang, D., Xu, Z., Duan, N., Zhou, M., Wang, J., and Yin, J. (2020). Reasoning over semantic-level graph for fact checking.