Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 6874–6882
Marseille, 11–16 May 2020

Automated Fact-Checking of Claims from Wikipedia

Aalok Sathe*, Salar Ather*, Tuan Manh Le*, Nathan Perry†, and Joonsuk Park*
*Department of Math and Computer Science, University of Richmond, VA, USA
†Department of Computer Science, Williams College, MA, USA
{aalok.sathe, salar.ather, tuan.le}@richmond.edu, nmp2@williams.edu, park@joonsuk.org

Abstract

Automated fact checking is becoming increasingly vital as both truthful and fallacious information accumulate online. Research on fact checking has benefited from large-scale datasets such as FEVER and SNLI. However, such datasets suffer from limited applicability due to the synthetic nature of claims and/or evidence written by annotators that differ from real claims and evidence on the internet. To this end, we present WIKIFACTCHECK-ENGLISH, 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. This is the largest fact checking dataset consisting of real claims and evidence to date; it will allow the development of fact checking systems that can better process claims and evidence in the real world. We also show that for the NLI subtask, a logistic regression system trained using existing and novel features achieves peak accuracy of 68%, providing a competitive baseline for future work. Also, a decomposable attention model trained on SNLI significantly underperforms the models trained on this dataset, suggesting that models trained on manually generated data may not be sufficiently generalizable or suitable for fact checking real-world claims.

Keywords: fact-checking, fact-verification, natural language inference, textual entailment, corpus

1. Introduction

The advancements in information technology have led to a rapid accumulation of textual content available online. While this has many positive implications, we are faced with the challenge of sifting truth from falsehood. Fact checking, the task of determining whether a given claim is true or false, has thus become an active area of research recently (Vlachos and Riedel, 2014).

Initial approaches to fact checking were manual, as done on websites such as `PolitiFact.com`, `FactCheck.org`, and `Snopes.com`. As expected, manual fact checking is not scalable due to the limitations imposed by the speed and capacity of human fact checkers (Sharma et al., 2019). As importantly, it is susceptible to human biases (Ciampaglia et al., 2015).

Researchers began to address the shortcomings by automating the fact checking process. Luckily, fact checking can leverage existing areas of research, such as information retrieval (IR) and natural language inference (NLI) / textual entailment (TE) (Vlachos and Riedel, 2014). To support the training of effective fact checkers and their components, a few large-scale dataset have been created and published: The Fact Extraction and Verification (FEVER) corpus is a collection of claims that support, refute, or are in unverifiable relationships (labeled as NOT ENOUGH INFORMATION) to Wikipedia articles (Thorne et al., 2018). The Stanford Natural Language Inference (SNLI) corpus consists of pairs of hypotheses and premises, each of which is labeled as ENTAILMENT, CONTRADICTION, or NEUTRAL (Bowman et al., 2015). Lastly, the MultiNLI corpus builds on the SNLI corpus to bring multiple genres together to address the domain- and genre-specificity of the existing and widely-used corpora (Williams et al., 2017).

While the aforementioned datasets are the first large-scale datasets that support active research on their respective tasks, they suffer from a common issue: the claims and evidence have been written and curated by annotators rather than crawled from the wild. Even though the annotation processes were carefully designed and conducted, the resulting claims are significantly different from the real-world claims. As shown in Table 1, the claims are self-contained, short and syntactically simple unlike the real-world claims, such as those extracted from the English Wikipedia (labeled as WIKIFACTCHECK-ENGLISH in the table).

Dataset	Example claim
FEVER	Oliver Reed was a film actor.
SNLI	Some men are playing a sport.
MultiNLI	People formed a line at the end of Pennsylvania Avenue.
WikiFactCheck-English	The hindwings are uniform grey with a narrow marginal line.

Table 1: Comparison of claims from existing large-scale datasets and WIKIFACTCHECK-ENGLISH

In this work, we present WIKIFACTCHECK-ENGLISH[1], a novel large-scale corpus for fact checking. It consists of 124,821 triples consisting of a real-world claim, its context and a cited evidence document extracted from the English Wikipedia; 34,783 of the triples are accompanied by a manually written claim that is refuted by the given evidence document (See an example entry in Table 2). This dataset was designed to be as realistic as possible, so that the fact checkers trained on this dataset can handle real claims effectively. In particular, the claims are real, often requiring a context to be fully comprehended, and the evidence is embedded in real documents from various domains.

To demonstrate the difficult yet feasible nature of train-

[1] The data and relevant code will be made available at the url `http://github.com/WikiFactCheck-English`

Field	Content
id	115724
claim	The hindwings are uniform grey with a narrow marginal line.
context	Eupoca sanctalis is a moth in the Crambidae family. It is found from central Costa Rica south to northern Colombia. The apical, subapical and tornal areas of the forewings are brown and the medial area is light brown. The antemedial and subterminal lines are white.
refuted	The hindwings are uniform blue with a broad marginal line.
url	http://digitalcommons.unl.edu/cgi/viewcontent.cgi?article=1054&context=systentomologyusda

Table 2: Example entry from WIKIFACTCHECK-ENGLISH data. The code for crawling and processing evidence files can be found in the public project repository.

ing fact checkers for real-world claims, we tackle the NLI subtask— determining whether a given claim is supported or refuted by a given evidence document. We employ various combinations of learning algorithms and features for this; we achieve a peak accuracy of 68.0% using a logistic regression model trained with existing and novel features. We also show that a pre-trained decomposable attention (DA) model designed for SNLI by Parikh et al. (2016) only achieves 58.4% accuracy. This provides an empirical support for the need for a dataset with real claims.

Our main contributions are two-fold. First, we present a large-scale dataset of real claims, context, and evidence documents extracted from the English Wikipedia, as well as manually written claims refuted by the evidence documents. This is the largest fact checking dataset of real claims and evidence documents to date; it will allow the development of fact checking systems that can effectively process claims that occur in the real world. Second, we design and analyze classifiers trained on our dataset for the NLI task, achieving a 68.0% accuracy on a held-out test set. This will serve as a meaningful baseline for future studies in this area.

In the remainder of the paper, we discuss related works (Section 2), describe the dataset and how it was constructed (Section 3), present NLI systems trained on the dataset (Section 4), analyze the experiment results (Section 5), and conclude with future work (Section 6).

2. Related Work

Fact checking The fact checking task was formally described in 2014 to overcome the limitations of manual fact annotation online in websites mentioned previously. This was done by creating a novel dataset of statements by important persons and/or political figures, their ratings as decided by journalists, and URLs to the evidence provided supporting or refuting the statements made (Vlachos and Riedel, 2014). This resulted in a dataset that is somewhat domain-specific, and has a particular type of construction, since the statements are spoken and designed to mislead people into believing a claim. A similar dataset and task is the Emergent dataset (Ferreira and Vlachos, 2016), featured online in the 'fakenewschallenge'[2]. The task involved determining whether a news headline maintains a stance in line with the rest of the article, or goes against it.

The more recent FEVER dataset was based on introductory sentences from Wikipedia, and evidence from within other related pages on Wikipedia (those that were linked to). Annotators performed mutations to generate positive and negative examples, as well as provided sentence-level annotations about what sentences from what other articles supported a particular claim. retrieval to pick the appropriate evidence.

Unfortunately, these datasets are not suitable for building systems that can process real-world claims. The size is too small or the claims deviate from real claims as they are written by annotators. In this work, we seek to address the issues by creating a large dataset of real claims. We carefully design the extraction to keep the claims and evidence realistic.

NLI/TE A core component of fact checking systems is NLI/TE, where the goal is to identify relationships between two spans of text. The first time textual entailment (TE) was formalized and made a public challenge on a large scale was with the PASCAL RTE challenge in 2005 (Dagan et al., 2010). The challenge stated the pressing issue that is the "variability of semantic expression" across Natural Language Understanding (NLU) tasks and subdomains, such as Information Retrieval (IR), Text Summarization, Question Answering (QA), and Machine Translation (MT). The crucial part of NLU that is common across all of these is the semantic component. The PASCAL challenge stated the difficulty of evaluation methods in MT, since there is no one single correct translation of some expression, but rather many possible manifestations of some underlying *meaning*. There have been many challenges. The seventh RTE challenge included two orders of magnitude examples more than the first one, at 21,000 (Ghuge and Bhattacharya, 2014).

There has been attempts to create datasets automatically using news sources (Burger and Ferro, 2005): the MITRE dataset, for example, extracted positive claims (only) from news article headlines and their text. This dataset lacked any negative examples, thus making it not very helpful for training NLI models. A dataset along the same lines was constructed by LCC as part of their 'GroundHog' submission to RTE 2006, using just headlines and the first paragraphs of news articles (Hickl et al., 2006). Positive examples were chosen using the first sentence of the first paragraph, based on the naive assumption that they would be very much related. In 2015, the Stanford NLP group came up with the first ever large annotated NLI dataset that could feasibly be used for deep machine learning methods (Bowman et al., 2015). This dataset was built using captions underneath images from Flickr, obtained using the Flickr30k dataset. Annotators were asked to mutate the claims to cre-

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

ate positive, negative, and neutral examples.

The MultiNLI dataset (Williams et al., 2017) seeks to address the genre-specificity introduced by the use of a single source of a specific domain used by SNLI. MultiNLI featured hypothesis and supporting texts from many different domains. Both datasets, however, like most RTE/NLI datasets, feature sen2sen textual entailment, which is uncharacteristic of how textual entailment or language inference works in the real world, which is more congruent to the fact checking task. Additionally, these large annotated datasets have their drawbacks as well, when it comes to artificial example generation, though it may be based on existing non-synthetic examples from the real world, by showing that a text classifier that did not even look at the evidence could still do relatively well on the two datasets in the three-way classification task (Gururangan et al., 2018). This suggests specific cues existing in the sentence-level annotations secretly reveal information about its attitude.

SciTail (Khot et al., 2018) is a domain-specific, completely unsynthetic dataset that state-of-the-art models for SNLI performed poorly on, suggesting poor cross-domain transferability, and difficulty of the task on natural vs. synthetic data.

By introducing WIKIFACTCHECK-ENGLISH, we aim to provide a large-scale annotated fact-checking corpus that resembles the real world closely. The experiment results show that a model trained on SNLI significantly underperforms models trained on this dataset, confirming the difference in the types of claims. Our dataset also supports fact checking and related tasks like information retrieval and textual entailment.

3. Data

Wikipedia is a massively collaborative repository of public knowledge. By design, it is encouraged that claims are be backed by sufficient evidence. A lot of the linked evidence ends up being primary articles and reports from all over the internet. Wikipedia makes all of its articles available to download as a data dump on a regular basis[3] in the form of a collection of XML *pages* each corresponding to an article, adding up to more than 60 gigabytes of raw content. We used a data dump from November 2017 of the English version of Wikipedia.

3.1. Components

Each instance in the dataset consists of four components: a claim, a context, an evidence file, and a refuted claim.

3.1.1. Claim

We extract claims by parsing wikitext using a MediaWiki markup parser called `mwparserfromhell`[4]. A typical Wikipedia citation looks as follows (extracted from the Wikipedia article on *Albedo*):

> The proportion reflected is not only determined by properties of the surface itself, but also by the spectral and angular distribution of solar radiation reaching the Earth's surface.[x]
>
> [x]: `http://curry.eas.gatech.edu/Courses/6140/ency/Chapter9/Ency_Atmos/Reflectance_Albedo_Surface.pdf`

This method works well to pick out straightforward claim-citation pairs. However, Wikipedia has many varieties of citations, and we must make modifications to our algorithm to accommodate them accurately.

We can have multiple citations for the same sentence. For instance, consider a sentence with citations of the form $S^{[x][y]}$. In such a case, we pick the earliest valid claim-evidence pair: (S, x), if the evidence at x is valid (described later). A simple modification to the existing method lets us include such examples.

Although we describe a way to pick claims and citations from sentences with citations spread over the sentence, they are not guaranteed to be well-formed clauses. Humans have added these annotations and are forced to judge where a citation must go. For instance, here is one entry from the Wikipedia article on *Albedo*:

> Many small objects in the outer Solar System[18] and asteroid belt have low albedos down to about 0.05.[19]

These two citations are not consistently placed. The claim before citation '[18]' doesn't tell us much, even though we know the intention was to provide evidence for a part of a claim; it should instead have been at the end of the sentence, similarly to the way '[19]' is placed. We therefore only pick a sentence if a part-of-speech tagger (Honnibal and Montani, 2017) can find a sentence phrase (SP) at the root of the text from the beginning till a citation mark.

Exclusion Not all claims are as well-formed. We exclude any entry with a special character in plaintext that does not have an ASCII encoding. We exclude any claims with some portion of a table, chart, or list appearing in them.

3.1.2. Context

A unique feature of our dataset is providing the context of each claim. Context may be necessary in case a claim has an unresolved reference. For instance, see the entry in Table 2—it is unclear whose 'hindwings' are being discussed here. In more extreme cases, the claim starts with a pronoun. However, it may be necessary to know that before determining whether the claim is supported or not, because there may be multiple lines in the original evidence source that could be talked about. In order to collect context, we aggregate all the sentences before a claim to the nearest previous paragraph break, or previous claim, whichever occurs earlier

3.1.3. Evidence

Based on the assumption that a cited document supports the respective claim in Wikipedia, we crawled the cited documents for the extracted claims. We chose to only download PDF references to ensure content quality. Every reference that returned an `application/pdf` MIME type to a HEAD HTTP request was downloaded, deduplicated, converted to text, and stored.

[3] `http://dumps.wikimedia.org/enwiki`

[4] `http://mwparserfromhell.readthedocs.io`

As of the initial crawl, there were 277,194 evidence files. This number makes manual filtering infeasible. Thus, we apply an automated filter based on the following criteria:

- **Size of content** Only those evidence documents with size between 2 kb and 1 mb are retained.

- **Language** Evidence documents for which a significant portion is in the English language (cutoff 90% and above) are retained, the rest are discarded.

- **Number of sentences in a paragraph** From observation, we found that measuring the number of sentences in a paragraph is a good proxy for whether the evidence document has any well-formatted paragraphs or not.

3.1.4. Refuted claim

To train fact checking systems, we also need claims that are refuted by the evidence documents to serve as negative examples. Thus we generated claims that are refuted by the evidence.

Automated annotation We have considered the following automated approaches, though these were not used to generate the final dataset. One way to generate refuted claims is to perform automatic claim negation using rule-based 'not' insertion based on syntax and part-of-speech (Bilu et al., 2015). However, this would result in a handful of negation words appearing in the refuted claims by design, causing classifiers to exploit this pattern.

Another automated approach is to pick a different random claim from the dataset. However, a "refuted" claim chosen this way is likely to be topically dissimilar from the evidence file, rendering it not a useful negative example for the classifier. More importantly, there is no guarantee that the claim is actually refuted.

Manual annotation These shortfalls of automated methods lead us to manually generate refuted claims with help from human annotators. As it would take much longer to read evidence files and generate a refuted claim, the annotators were asked to read claims and write new claims that are *definitely* false assuming the respective original claim is true. We restricted generating refuted claims corresponding to only the claims with 15 words or fewer. We chose this limit for a number of reasons: the claims got tedious to work with and annotators produced unreliable results with too long claims, based on our pilot data collection (described below). Additionally, we investigated the distribution of claim lengths, and picked 15 as our threshold since 15 or fewer tokens comprised a significant chunk of claims.

For one phase of writing refuted claims, a pilot study was done using Amazon Mechanical Turk (MTurk). Workers were given five sentences in one human intelligence task (HIT), and were asked to construct corresponding sentences that would be definitely false given the original ones. The workers were additionally asked to retain the same subject, and stick to the original topic. The workers were also asked to keep the new sentence length (in terms of number of tokens) close to the original ones (within one or two tokens of each other). We observed suboptimal annotation performance from MTurk workers. In a large part, this was due to poor understanding of negating the logic of a sentence.

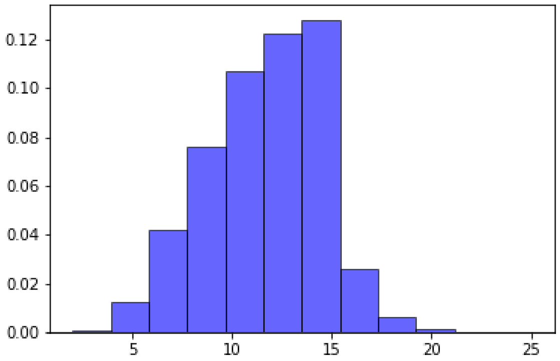

Figure 1: Densities of claim and refuted claim lengths (in tokens) in the annotated portion of WIKIFACTCHECK-ENGLISH data

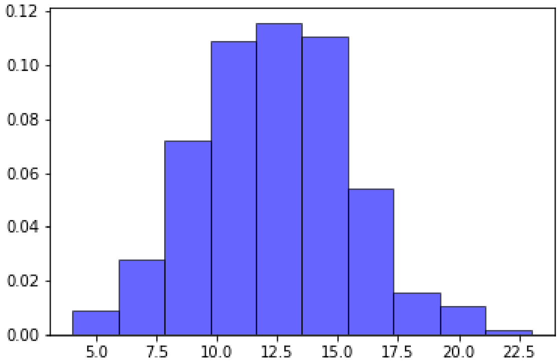

Figure 2: Densities of most relevant evidence sentence (e_c) lengths (in tokens) in the annotated WIKIFACTCHECK-ENGLISH data

Many of the poorly formed refuted claims were modified in some way but they were not necessarily false given the positive examples. Instead, they were only empirically false or unlikely to be true presumably based on the MTurk workers' real-world experiences. We resolved to hand-picking a handful of MTurk workers who performed well (getting $\geq$ 4 out of 5 correct) on the pilot, and gave more claims to annotate. In addition, undergraduate students at the University of Richmond were involved in generating refuted claims.

As opposed to machine-generated refuted claims, hand-annotations allow us to construct refuted claims by modifying original claims by understanding what they are about, while, at the same time, making their refuted versions sound as natural and original as possible.

3.2. Resulting dataset

The final dataset contains 124,821 entries as shown in Table 3. Each entry has an id, claim, context, and an evidence url, in the least. Among these, 34,783 entries have claims with 15 or fewer tokens. These entries also have corresponding refuted claims, manually constructed by human annotators based on certain guidelines to keep them similar

Entries without refuted claims		Entries with refuted claims			Total
		Training set	Test set		
		24,348	10,435		
90,038	+	34,783		=	124,821

Table 3: Summary of the train/test split and database size

Domain	Count
www.hpo.ncdcr.gov	1512
www.dhr.virginia.gov	1090
grfx.cstv.com	824
eci.nic.in	814
www.mapress.com	809
www.dtic.mil	737
www.americanradiohistory.com	623
www.la84foundation.org	527
www.researchgate.net	509
ec.europa.eu	472
www.nps.gov	471
www.gov.uk	466
sora.unm.edu	449
www.ncbi.nlm.nih.gov	410

Table 4: Top 15 domains most frequently appearing as sources of evidence documents (Wikipedia citations)

and relevant. Among the annotated entries, we set aside a held-out test set (30%). The dataset is provided in the jsonl format for ease of usage. The mean number of tokens in a claim throughout the dataset including the portion without annotations available is 19.99. The same metric for claims in just the annotated portion is about 11.62; and the annotated refuted claims are comparable in length (about 12 tokens each) to their positive counterparts (about 11 tokens each; see Figures 1 and 2). The average length of the context in terms of number of tokens was about 81 (spanning multiple sentences).

We found that the top sources of evidence for claims came from government and educational websites, public knowledge bases, and research article hosting sites, which are generally considered credible sources of information (see Table 4). We ran topic modeling using Latent Dirichilet allocation to discover the most common topics in our data for various numbers of topics from 15 through 25. For this we used context and claim concatenated together for each entry, and used all entries from the annotated portion. The groupings that were found corresponding to numbers, punctuation, other symbols, etc. were ignored. For the rest, we came up with a few possible topic names based on our judgment using the highest weighted words of the respective grouping. What follows is a list of the salient topic names we derived from these groupings (separated by semicolons): education, research; planning, development; moths, insects; law, government; historic buildings; production (industry); rivers, lakes; sports, football (soccer); US presidents; phylogenetics; film, literature; print media; climate; war; population; software, computing; aviation; storm, natural calamities. Topics that are rarer in occurrence may not have made it to the list above, but still be present in the dataset.

4. Natural Language Inference (NLI)

To demonstrate the difficult yet feasible nature of training fact checking systems for real-world claims, we consider the NLI subtask: Given a claim c and a corresponding evidence document E_c, determine whether c is *supported* or *refuted* by E_c.[5]

Based on the observation that a claim is often highly similar to a sentence in the respective evidence file (we will look at an example in Section 4.1), we develop a two-step pipelined approach:

1. **Passage Retrieval** Extract sentences e_c from the document E_c, that are likely to be most relevant to c.

2. **Support Verification** Classify c as supported or refuted by e_c.

4.1. Passage retrieval

One way to approach passage retrieval is to identify a single most relevant sentence. Typically, there exists a sentence in the evidence E_c that directly supports or refutes the claim c, and such sentence is similar to c as in the following example. This is distinct from other unrelated sentences in the evidence document, where they may only have a few to no words in common:

> **Correct label:** supported
> c**:** In contrast with lunar outpost missions, lunar sorties will be of relatively brief duration.
> e_c**:** The targeting for the outpost mission differs from the sortie mission due to the proximity to the Lunar pole.
> **unrelated**$_1$**:** These techniques applied to the Lunar sortie and the Lunar outpost missions.
> **unrelated**$_2$**:** The current polar outpost mission design targets the CEV/LSAM to a 90<B0> inclined parking orbit.

To capture this, we use normalized Levenshtein distance between each sentence in the evidence document E and c and picked the one with the highest score (‘e_c’). We used Levensthein distance because it worked the best in a preliminary study in which other measures like cosine similarity were tested.

[5] There are also other formulations of NLI, such as the one used in the SNLI dataset benchmark task

4.2. Support Verification

Unlike the previous step, determining whether or not c is supported by e_c is not as straight forward. Thus, learning from the data is necessary.

4.2.1. Models

We employ both feature-based and neural network models.

Feature-based models We use a Linear SVC Support Vector Machine (SVM) classifier with up to 5000 iterations to converge. To find optimal C values (model hyperparameter) for each case, we perform 5-fold cross validation with grid search between 0.5 and 3 in increments of 0.1. We also use Logistic Regression with a maximum of 5000 iterations.

Neural network model We also employ the Decomposable Attention ('DA') model implemented in AllenNLP (Gardner et al., 2018). The DA model was the state-of-the-art in 2016, constructed as part of the SNLI challenge, evaluated at 86.3% accuracy on the test set of SNLI. Newer models have since surfaced, but DA is still one of the top-performing models. At the same time, it is simple and easy to adapt, as it is not an ensemble approach and does not rely on syntactic parsing. In mapping the labels outputted by DA, we merge NEUTRAL with CONTRADICTION to form the negative class in our experiments.

4.2.2. Features

In addition to the existing word pairs feature, we develop several novel features.

Word pairs The Cartesian product of tokens in the claim c and most relevant evidence sentence e_c. This is a widely used extension of unigrams to cases in which information needs to be drawn from two spans of text (Marcu and Echihabi, 2002; Park and Cardie, 2012). With enough data, this featureset can identify relationships between pairs of words. For example, the pair (NOT, NO) turned out to be useful for this task, since it captures a shared sentiment between c and e_c)

Sentiment We designed three features relating to sentiments of words. For each one, we use the SentiWordNet (Esuli and Sebastiani, 2006) in NLTK (Loper and Bird, 2002) and define the sentiment of token as the difference between the positive and negative sentiment scores.

- **Average Sentiment (AVE_SEN)** Sentiment score of individual tokens, averaged over c. Factual claims tend to have a neutral sentiment, as they typically state objective aspects of a topic. Also, it is less likely to have a negative sentiment than to have a positive sentiment. This is because it is more common to describe what is the case, rather than what is not the case. These can be partially captured by the average sentiment of the tokens.

- **Max Sentiment (MAX_SEN)** The highest sentiment score among those of tokens in c. The rationale is the same as AVE_SEN, but taking the max minimizes the chance of the sentiment queue being lost when most of the tokens in a given c is neutral.

- **Sentiment Difference (DIFF_SEN)** The difference between the average sentiment scores of c and e_c. If c and e_c share the same sentiment, they are more likely to be stating the same information, which in turn means at that e_c is likely to support c. On the other hand, if they have drastically different sentiments, it could mean that e_c refutes c.

Antonymy A claim c that is supported by its evidence e_c would likely use adjectives and verbs that are similar in meaning to those used in e_c. A claim that is refuted, too, would have a relatively large number of synonyms, just from being topically similar and talking about the same subject. The crucial differentiating factor, then, might be the existence of antonyms: a claim is more likely to be refuted by e_c if it contains even one antonym of a word in e_c. On the other hand, a claim c that is supported by e_c should tend to have no antonyms.

Along this intuition, we define two features capturing the existence of antonyms. Since it is more convenient to compute how similar two sets of words are, rather than how 'opposite' the two sets are, we compute the antonymy score of c and e_c by measuring similarity scores between tokens in c and antonyms of tokens in e_c, and vice versa. Here, we extracted the top antonym synsets (corresponding to the POS tag of the token) to extract antonyms and used path similarity to measure the similarity (Miller, 1995). The path similarity is a metric based on the length of the shortest path from one synset to another via hyponymy and hypernymy relations in WordNet. The scores range from 0, the least similar, to 1, the most similar (identical).

- **Average antonymy (AVE_ANT)** Average of the similarity between tokens from the pairs in the Cartesian product $(token\_antonyms(c) \times \text{tokens}(e_c)) \cup (\text{tokens}(c) \times token\_antonyms(e_c))$. Here, let $token\_antonyms(c)$ and $token\_antonyms(e_c)$ be the set of antonyms of tokens in c and e_c, respectively. This feature can be useful when multiple words mean the opposite, while none them are strong antonyms.

- **Max antonymy (MAX_ANT)** Maximum of the similarity between tokens from the pairs in the Cartesian product $(token\_antonyms(c) \times \text{tokens}(e_c)) \cup (\text{tokens}(c) \times token\_antonyms(e_c))$. This feature can capture whether strong or obvious antonyms exist, which would mean that e_c is likely to refute c.

5. Experiment Results & Analysis

We randomly split the corpus into a training set (70%) and a test set (30%), comprised of only all the entries that have annotated negative examples. Entries without annotations are not included in this split or in the experiments below.

The results reported in Table 5 show feature-based models outperforming the DA model by a noticeable margin. And upon further analysis, we find that the novel sentiment and antonymy features are helpful for this task.

Feature-based vs neural network models The DA model pre-trained on SNLI performs significantly worse than most feature-based models. Consider the following instance where the best LR model made a correct prediction, while DA did not:

Model	Featureset	Prec.	Rec.	F1	Acc.
Decomposable Attention (DA)	n/a	0.562	0.763	0.647	0.584
LinearSVC (SVM)	word pairs	0.625	**0.784**	**0.696**	0.657
	word pairs + antonymy	0.635	0.754	0.689	**0.660**
	word pairs + sentiment	0.638	0.731	0.681	0.658
	sentiment + antonymy	0.578	0.604	0.591	0.581
	word pairs + sentiment + antonymy	**0.646**	0.707	0.675	**0.660**
Logistic Regression (LR)	word pairs	0.645	**0.784**	**0.708**	0.676
	word pairs + antonymy	0.651	0.749	0.697	0.674
	word pairs + sentiment	0.658	0.750	0.701	**0.680**
	sentiment + antonymy	0.578	0.600	0.589	0.581
	word pairs + sentiment + antonymy	**0.664**	0.729	0.695	**0.680**

Table 5: Overall classifier evaluation performance on the Natural Language Inference subtask (subtask of the Automated Fact-Checking task). The results for DA are using training on SNLI and evaluation on WIKIFACTCHECK-ENGLISH held-out test set.

> **label:** supported
> c**:** The extratropical system was completely absorbed by the front six hours later.
> e_c**:** The circulation of Shary was completely absorbed within the frontal zone that day.

The sentences are much more complex than a typical sentence from SNLI, both from the perspective of morphology and syntax. In fact, this is expected: It is hard to expect an annotator to write claims about specific subjects in a great detail. As the sentences are drastically different from what the DA model was trained on, the model is not able to make a correct classification. This serves as an empirical support for the main premise of this work: real-world claims are different from synthetic claims, and systems trained using synthetic claims will not process real claims as well. The feature-based models, on the other hand, can leverage various aspects of the sentences, such as the shared sentiment and lack of antonyms to, classify it correctly.

Most informative features The highest accuracy of 68.0% is achieved by logistic regression with all features. Here, the most informative features were the sentiment and antonymy features. AVE_SEN and DIFF_SEN are effective in identifying the supported cases, and MAX_SEN and MAX_ANT are the most useful for recognizing the refuted cases.

As previously discussed, factual claims tend to have a neutral sentiment, which can be captured by the average of the token sentiments (i.e., AVE_SEN), whereas the max of them (i.e., MAX_SEN) is prone to noise. Another way interpretation is that the lack of negative sentiment in the truthful claim keeps the average sentiment score higher, and it is more rare for a factual claim to have a negative sentiment than a positive one. DIFF_SEN is useful, because When e_c supports c, they typically have similar sentiment, and so the difference is close to zero. Also, MAX_SEN and MAX_ANT can be good queues for the refuted cases, since they capture the presence of a strong emotion and antonym, respectively, both of which characterize refuted relations. Unlike other other sentiment and antonymy features, AVG_ANT is not very helpful. This is because there is usually at most one pair of words with a high antonymy score across the two sentences, and averaging makes the AVG_ANT value less distinguishable from instances without antonyms.

The importance of sentiment as well as antonymy is also shown by word pairs: (NOT, NO) was highly corrected with the supported class, and (LEAST, MOST) with the refuted class.

All features vs word pairs only Employing only the word pairs leads to a higher recall, whereas using all features result in models with a higher precision. This aligns with the expectation that the novel features designed to capture specific characteristics of c and e_c help with precision while interferes with recall. Here is an example where the sentiment and antonymy features help:

> **label:** supported
> c**:** MEA is also related to means-ends chain approach used commonly in consumer behavior analysis.
> e_c**:** (1991) ‘Improvements in means-end chain analysis: Using graph theory and correspondence analysis’, Journal of Advertising Research, Vol.

The classifier assigns a low value to DIFF_SEN from recognizing the congruence of sentiment across these sentences. Also the antonymy features will capture the lack of antonymy. As a result, the claim is correctly classified as supported when all features are used. Such aspects of this example, however, is not recognized by word pairs alone.

However, the novel features are not always helpful. For example, consider the claim and relevant evidence sentence pair:

> **label:** supported
> c**:** This species lives in a number of countries and islands including: Latvia Bulgaria .
> e_c**:** These species are newly recorded to the fauna of Bulgaria.

Classifiers employing all features misclassify this instance, unlike those employing only word pairs. The misclassification is influenced by the high anotnymy score between 'species' and 'fauna'. This in turn was caused by 'vegetation', a synonym of 'species', being an antonym of 'fauna'. This is a result of 'species' being a common hypernym of the two antonyms 'fauna' and 'vegetation'. This is a unique situation that we did not account for in implementing the features. There are also other instances where the baseline classifier correctly classifies the example but classifiers using the novel features do not.

6. Conclusion

The need for reliable automatic fact checking systems will continue to grow in line with the drastic increase in the amount of information available online. While large-scale datasets published in the last few years paved the way for fact checking research, we need more realistic claims and evidence files to build fact checkers that can handle claims in the real world. To this end, we have presented WIKIFACTCHECK-ENGLISH, a large-scale dataset of real claims, their contexts, and evidence documents extracted from the English Wikipedia, along with manually written claims refuted by the evidence documents. This is the largest fact checking dataset of real claims and evidence documents to date.

We have experimented with various combinations of learning algorithms and features for the NLI subtask, achieving a 68.0% accuracy. Our findings suggests that models trained on manually generated claims, e.g. SNLI, may not be effective in fact checking real-world claims. Our corpus and the baseline results illustrate the non-trivial nature of this task, and the manifold increase in difficulty with introduction of real-world data. We anticipate this corpus will provide a useful test bed and benchmark for fact checking systems.

There are several feasible extensions to this work, including building a bigger corpus and developing sophisticated fact checking systems, such as those that make use of the context.

7. Acknowledgements

We thank our annotators for their contributions and anonymous reviewers for their helpful input. We are also grateful towards the University of Richmond and Williams College for their generous support for this project.

8. Bibliographical References

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.

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.

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.

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

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*.

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.

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.

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*.

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

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.