DeSePtion: Dual Sequence Prediction and Adversarial Examples for Improved Fact-Checking

Christopher Hidey,[1]* Tuhin Chakrabarty,[1] Tariq Alhindi,[1] Siddharth Varia,[1]
Kriste Krstovski,[1,3] Mona Diab,[4,5]* and Smaranda Muresan[1,2]
[1]Department of Computer Science, Columbia University
[2]Data Science Institute, Columbia University
[3]Columbia Business School, Columbia University [4]Facebook AI
[5]Department of Computer Science, George Washington University
{chidey, tariq, smara}@cs.columbia.edu, mtdiab@email.gwu.edu
{tuhin.chakrabarty, sv2504, kriste.krstovski}@columbia.edu

Abstract

The increased focus on misinformation has spurred development of data and systems for detecting the veracity of a claim as well as retrieving authoritative evidence. The Fact Extraction and VERification (FEVER) dataset provides such a resource for evaluating end-to-end fact-checking, requiring retrieval of evidence from Wikipedia to validate a veracity prediction. We show that current systems for FEVER are vulnerable to three categories of realistic challenges for fact-checking – multiple propositions, temporal reasoning, and ambiguity and lexical variation – and introduce a resource with these types of claims. Then we present a system designed to be resilient to these "attacks" using multiple pointer networks for document selection and jointly modeling a sequence of evidence sentences and veracity relation predictions. We find that in handling these attacks we obtain state-of-the-art results on FEVER, largely due to improved evidence retrieval.

1 Introduction

The growing presence of biased, one-sided, and often altered discourse, is posing a challenge to our media platforms from newswire to social media (Vosoughi et al., 2018). To overcome this challenge, fact-checking has emerged as a necessary part of journalism, where experts examine "check-worthy" claims (Hassan et al., 2017) published by others for their "shades" of truth (e.g., FactCheck.org or PolitiFact). However, this process is time-consuming, and thus building computational models for automatic fact-checking has become an active area of research (Graves, 2018). Advances were made possible by new open source datasets and shared tasks: the Fact Extraction and Verification Shared Task (FEVER) 1.0 and 2.0 (Thorne et al., 2018; Thorne and Vlachos, 2019), SemEval 2019 Shared Task 8: Fact-Checking in Community Forums (Mihaylova et al., 2019), and LIAR(+) datasets with claims from PolitiFact (Wang, 2017; Alhindi et al., 2018).

Claim: Murda Beatz's real name is Marshall Mathers.
Evidence: [Murda Beatz] Shane Lee Lindstrom (born February 11, 1994), known professionally as Murda Beatz, is a Canadian hip hop record producer and songwriter from Fort Erie, Ontario.
Label: REFUTES

Figure 1: Example from FEVER 1.0 Dataset

The FEVER 1.0 shared task dataset (Thorne et al., 2018) has enabled the development of end-to-end fact-checking systems, requiring document retrieval and evidence sentence extraction to corroborate a veracity relation prediction (supports, refutes, not enough info). An example is given in Figure 1. Since the claims in FEVER 1.0 were manually written using information from Wikipedia, the dataset may lack linguistic challenges that occur in verifying naturally occurring check-worthy claims, such as temporal reasoning or lexical generalization/specification. Thorne and Vlachos (2019) designed a second shared task (FEVER 2.0) for participants to create adversarial claims ("attacks") to break state-of-the-art systems and then develop systems to resolve those attacks.

We present a **novel dataset of adversarial examples** for fact extraction and verification in three challenging categories: 1) multiple propositions (claims that require multi-hop document or sentence retrieval); 2) temporal reasoning (date comparisons, ordering of events); and 3) named entity ambiguity and lexical variation (Section 4). We show that **state-of-the-art systems are vulnerable** to adversarial attacks from this dataset (Section 6). In addition, we take steps toward addressing these vulnerabilities, presenting a system for end-to-end fact-checking that brings **two novel contri-**

*Work completed in part at Amazon

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 8593–8606
July 5 - 10, 2020.

butions using pointer networks: 1) a document ranking model; and 2) a joint model for evidence sentence selection and veracity relation prediction framed as a sequence labeling task (Section 5). Our new system achieves **state-of-the-art results for FEVER** and we present an **evaluation of our models** including ablation studies (Section 6). Data and code will be released to the community.[1]

2 Related Work

Approaches for predicting the veracity of naturally-occurring claims have focused on statements fact-checked by journalists or organizations such as PolitiFact.org (Vlachos and Riedel, 2014; Alhindi et al., 2018), news articles (Pomerleau and Rao, 2017), or answers in community forums (Mihaylova et al., 2018, 2019). However, those datasets are not suited for end-to-end fact-checking as they provide sources and evidence while FEVER (Thorne et al., 2018) requires retrieval.

Initial work on FEVER focused on a pipeline approach of retrieving documents, selecting sentences, and then using an entailment module (Malon, 2018; Hanselowski et al., 2018; Tokala et al., 2019); the winning entry for the FEVER 1.0 shared task (Nie et al., 2019a) used three homogeneous neural models. Other work has jointly learned either evidence extraction and question answering (Nishida et al., 2019) or sentence selection and relation prediction (Yin and Roth, 2018; Hidey and Diab, 2018); unlike these approaches, we use the same sequential evidence prediction architecture for both document and sentence selection, jointly predicting a sequence of labels in the latter step. More recently, Zhou et al. (2019) proposed a graph-based framework for multi-hop retrieval, whereas we model evidence sequentially.

Language-based adversarial attacks have often involved transformations of the input such as phrase insertion to distract question answering systems (Jia and Liang, 2017) or to force a model to always make the same prediction (Wallace et al., 2019). Other research has resulted in adversarial methods for paraphrasing with universal replacement rules (Ribeiro et al., 2018) or lexical substitution (Alzantot et al., 2018; Ren et al., 2019). While our strategies include insertion and replacement, we focus specifically on challenges in fact-checking. The task of natural language inference (Bowman et al., 2015; Williams et al., 2018) provides similar challenges: examples for numerical reasoning and lexical inference have been shown to be difficult (Glockner et al., 2018; Nie et al., 2019b) and improved models on these types are likely to be useful for fact-checking. Finally, (Thorne and Vlachos, 2019) provided a baseline for the FEVER 2.0 shared task with entailment-based perturbations. Other participants generated adversarial claims using implicative phrases such as "not clear" (Kim and Allan, 2019) or GPT-2 (Niewinski et al., 2019). In comparison, we present a diverse set of attacks motivated by realistic, challenging categories and further develop models to address those attacks.

3 Problem Formulation and Datasets

We address the end-to-end fact-checking problem in the context of FEVER (Thorne et al., 2018), a task where a system is required to verify a claim by providing evidence from Wikipedia. To be successful, a system needs to predict both the correct veracity relation– supported (S), refuted (R), or not enough information (NEI)– and the correct set of evidence sentences (not applicable for NEI). The **FEVER 1.0** dataset (Thorne et al., 2018) was created by extracting sentences from popular Wikipedia pages and mutating them with paraphrases or other edit operations to create a claim. Then, each claim was labeled and paired with evidence or the empty set for NEI. Overall, there are 185,445 claims, of which 90,367 are S, 40,107 are R, and 45,971 are NEI. Thorne and Vlachos (2019) introduced an adversarial set up for the **FEVER 2.0** shared task – participants submitted claims to break existing systems and a system designed to withstand such attacks. The organizers provided a baseline of 1000 adversarial examples with negation and entailment-preserving/-altering transformations and this set was combined with examples from participants to form the FEVER 2.0 dataset. Table 1 shows the partition of FEVER 1.0 and 2.0 data (hereafter FV1/FV2-train/dev/test).

Dataset	Train	Dev.	Blind Test
FEVER 1.0	145,449	19,998	19,998
FEVER 2.0	–	1,174	1,180

Table 1: FEVER Dataset Statistics

4 Adversarial Dataset for Fact-checking

While the FEVER dataset is a valuable resource, our goal is to evaluate complex adversarial claims

[1] https://github.com/chridey/fever2-columbia

which resemble check-worthy claims found in news articles, speeches, debates, and online discussions. We thus propose three types of attacks based on analysis of FV1 or prior literature: those using multiple propositions, requiring temporal and numerical reasoning, and involving lexical variation.

For the **multi-propositional** type, Graves (2018) notes that professional fact-checking organizations need to synthesise evidence from multiple sources; automated systems struggle with claims such as *"Lesotho is the smallest country in Africa."* In FV1-dev, 83.18% of S and R claims require only a single piece of evidence and 89% require only a single Wikipedia page. Furthermore, our previous work on FEVER 1.0 found that our model can fully retrieve 86% of evidence sentences from Wikipedia when only a single sentence is required, but the number drops to 17% when 2 sentences are required and 3% when 3 or more sentences are required (Hidey and Diab, 2018).

For the second type, check-worthy claims are often numerical (Francis, 2016) and **temporal reasoning** is especially challenging (Mirza and Tonelli, 2016). Rashkin et al. (2017) and Jiang and Wilson (2018) showed that numbers and comparatives are indicative of truthful statements in news, but the presence of a date alone does not indicate its veracity. In FV1-dev, only 17.81% of the claims contain dates and 0.22% contain time information.[2] To understand how current systems perform on these types of claims, we evaluated three state-of-the-art systems from FEVER 1.0 (Hanselowski et al., 2018; Yoneda et al., 2018; Nie et al., 2019a), and examined the predictions where the systems disagreed. We found that in characterizing these predictions according to the named entities present in the claims, the most frequent types were numerical and temporal (such as percent, money, quantity, and date).

Finally, adversarial attacks for **lexical variation**, where words may be inserted or replaced or changed with some other edit operation, have been shown to be effective for similar tasks such as natural language inference (Nie et al., 2019b) and question answering (Jia and Liang, 2017), so we include these types of attacks as well. For the fact-checking task, models must match words and entities across claim and evidence to make a veracity prediction. As claims often contain ambiguous entities (Thorne and Vlachos, 2018) or lexical features indicative of credibility (Nakashole and Mitchell, 2014), we desire models resilient to minor changes in entities (Hanselowski et al., 2018) and words (Alzantot et al., 2018).

We thus create an adversarial dataset of 1000 examples, with 417 multi-propositional, 313 temporal and 270 lexically variational. Representative examples are provided in Appendix A.

Multiple Propositions Check-worthy claims often consist of multiple propositions (Graves, 2018). In the FEVER task, checking these claims may require retrieving evidence sequentially after resolving entities and events, understanding discourse connectives, and evaluating each proposition.

Consider the claim *"Janet Leigh was from New York and was an author."* The Wikipedia page **[Janet Leigh]** contains evidence that she was an author, but makes no mention of New York. We generate new claims of the CONJUNCTION type *automatically* by mining claims from FV1-dev and extracting entities from the subject position. We then combine two claims by replacing the subject in one sentence with a discourse connective such as "and." The new label is S if both original claims are S, R if at least one claim is R, and NEI otherwise.

While CONJUNCTION claims provide a way to evaluate multiple propositions about a single entity, these claims only require evidence from a single page; hence we create new examples requiring reasoning over multiple pages. To create MULTI-HOP examples, we select claims from FV1-dev whose evidence obtained from a single page P contains at least one other entity having a valid page Q. We then modify the claim by appending information about the entity which can be verified from Q. For example, given the claim *"The Nice Guys is a 2016 action comedy film."* we make a multi-hop claim by obtaining the page **[Shane Black]** (the director) and appending the phrase *"directed by a Danish screenwriter known for the film Lethal Weapon."*

While multi-hop retrieval provides a way to evaluate the S and R cases, composition of multiple propositions may also be necessary for NEI, as the relation of the claim and evidence may be changed by more general/specific phrases. We thus add ADDITIONAL UNVERIFIABLE PROPOSITIONS that change the gold label to NEI. We selected claims from FV1-dev and added propositions which have no evidence in Wikipedia (e.g. for the claim *"Duff McKagan is an American citizen,"* we can add the reduced relative clause *"born in Seattle"*).

[2] As determined by NER using Spacy: https://spacy.io

Temporal Reasoning Many check-worthy claims contain dates or time periods and to verify them requires models that handle temporal reasoning (Thorne and Vlachos, 2017).

In order to evaluate the ability of current systems to handle temporal reasoning we modify claims from FV1-dev. More specifically, using claims with the phrase "in <date>" we *automatically* generate seven modified claims using simple DATE MANIPULATION heuristics: arithmetic (e.g., *"in 2001"* → *"4 years before 2005"*), range (*"in 2001"* → *"before 2008"*), and verbalization (*"in 2001"* → *"in the first decade of the 21st century"*).

We also create examples requiring MULTI-HOP TEMPORAL REASONING, where the system must evaluate an event in relation to another. Consider the S claim *"The first governor of the Indiana Territory lived long enough to see it become a state."* A system must resolve entity references (Indiana Territory and its first governor, William Henry Harrison) and compare dates of events (the admittance of Indiana in 1816 and death of Harrison in 1841). While multi-hop retrieval may resolve references, the model must understand the meaning of *"lived long enough to see"* and evaluate the comparative statement. To create claims of this type, we mine Wikipedia by selecting a page X and extracting sentences with the pattern "is/was/named the A of Y" (e.g. A is *"first governor"*) where Y links to another page. Then we manually create temporal claims by examining dates on X and Y and describing the relation between the entities and events.

Named Entity Ambiguity and Lexical Variation As fact-checking systems are sensitive to lexical choice (Nakashole and Mitchell, 2014; Rashkin et al., 2017), we consider how variations in entities and words may affect veracity relation prediction.

ENTITY DISAMBIGUATION has been shown to be important for retrieving the correct page for an entity among multiple candidates (Hanselowski et al., 2018). To create examples that contain ambiguous entities we selected claims from FV1-dev where at least one Wikipedia disambiguation page was returned by the Wikipedia python API.[3] We then created a new claim using one of the documents returned from the disambiguation list. For example the claim *"Patrick Stewart is someone who does acting for a living."* returns a disambiguation page, which in turn gives a list of pages such as **[Patrick Stewart]** and **[Patrick Maxwell Stewart]**.

Finally, as previous work has shown that neural models are vulnerable to LEXICAL SUBSTITUTION (Alzantot et al., 2018), we apply their genetic algorithm approach to replace words via counter-fitted embeddings. We make a claim adversarial to a model fine-tuned on claims and gold evidence by replacing synonyms, hypernyms, or hyponyms, e.g. *created* → *established*, *leader* → *chief*. We manually remove ungrammatical claims or incorrect relations.

5 Methods

Verifying check-worthy claims such as those in Section 4 requires a system to 1) make sequential decisions to handle multiple propositions, 2) support temporal reasoning, and 3) handle ambiguity and complex lexical relations. To address the first requirement we make use of a pointer network (Vinyals et al., 2015) in two novel ways: i) to rerank candidate documents and ii) to jointly predict a sequence of evidence sentences and veracity relations in order to compose evidence (Figure 3). To address the second we add a post-processing step for simple temporal reasoning. To address the third we use rich, contextualized representations. Specifically, we fine-tune BERT (Devlin et al., 2019) as this model has shown excellent performance on related tasks and was pre-trained on Wikipedia.

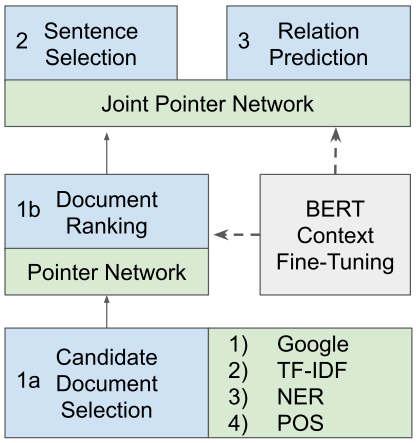

Figure 2: Our FEVER pipeline: 1) Retrieving Wikipedia pages by selecting an initial candidate set (1a) and ranking the top D (1b); 2) Identifying the top N sentences; 3) Predicting supports, refutes, or not enough info. Dashed arrows indicate fine-tuning steps.

Our full pipeline is presented in Figure 2. We first identify an initial **candidate set of documents**

[3] https://pypi.org/project/wikipedia/

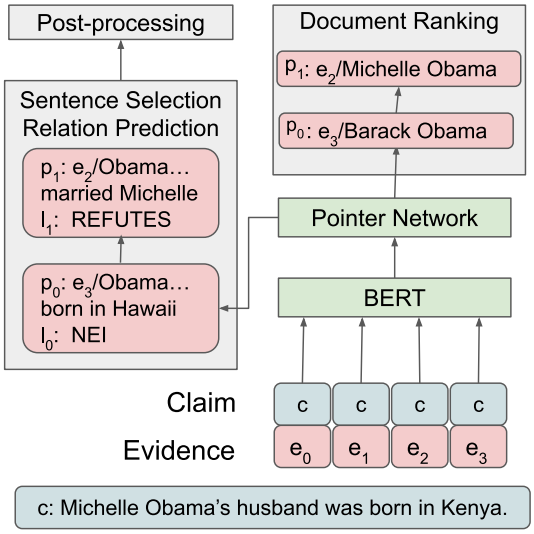

Figure 3: Pointer network architecture. Claim and evidence (page title or sentence) are embedded with BERT and evidence is sequentially predicted (for sentence selection the relation sequence is jointly predicted).

(1a) by combining the top M pages from a TF-IDF search using DrQA (Chen et al., 2017) with pages from the approach of Chakrabarty et al. (2018), which provides results from Google search and predicted named entities and noun phrases. Then, we perform **document ranking** by selecting the top $D < M$ pages with a pointer network (1b). Next, an N-long sequence of evidence sentences (2) and veracity relation labels (3) are **predicted jointly by another pointer network**.

Prior to training, we fine-tune BERT for document and sentence ranking on claim/title and claim/sentence pairs, respectively. Each claim and evidence pair in the FEVER 1.0 dataset has both the title of the Wikipedia article and at least one sentence associated with the evidence, so we can train on each of these pairs directly. For the claim *"Michelle Obama's husband was born in Kenya"*, shown in Figure 3, we obtain representations by pairing this claim with evidence sentences such as *"Obama was born in Hawaii"* and article titles such as **[Barack Obama]**.

The core component of our approach is the pointer network, as seen in Figure 3. Unlike our previous work (Hidey and Diab, 2018), we use the pointer network to re-rank candidate documents and jointly predict a sequence of evidence sentences and relations. Given a candidate set of evidence (as either document titles or sentences) and a respective fine-tuned BERT model, we extract features for every claim c and evidence e_p pair by summing the $[CLS]$ embedding for the top 4 layers (as recommended by Devlin et al. (2019)):

$$m_p = BERT(c, e_p) \tag{1}$$

Next, to select the top k evidence, we use a pointer network over the evidence for claim c to extract evidence recurrently by computing the extraction probability $P(p_t|p_0 \cdots p_{t-1})$ for evidence e_p at time $t < k$. At time t, we update the hidden state z_t of the pointer network decoder. Then we compute the weighted average h_t^q of the entire evidence set using q hops over the evidence (Vinyals et al., 2016; Sukhbaatar et al., 2015):[4]

$$\begin{aligned} \alpha_t^o &= \text{softmax}(v_h^T \tanh(W_g m_p + W_a h_t^{o-1})) \\ h_t^o &= \sum_j \alpha_t^o W_g m_j \end{aligned} \tag{2}$$

We concatenate m_p and h_t^q and use a multi-layer perceptron (MLP) to predict p_t. The loss is then:

$$\mathcal{L}(\theta_{ptr}) = -1/k \sum_{t=0}^{k-1} \log P_{\theta_{ptr}}(p_t|p_{0:t-1}) \tag{3}$$

We train on gold evidence and perform inference with beam search for both document ranking (Section 5.1) and joint sentence selection and relation prediction (Section 5.2).

5.1 Document Ranking

In order to obtain representations as input to the pointer network for document ranking, we leverage the fact that Wikipedia articles all have a title (e.g. **[Barack Obama]**), and fine-tune BERT on title and claim pairs, in lieu of examining the entire document text (which due to its length is not suitable for BERT). Because the title often overlaps lexically with the claim (e.g. **[Michelle Obama]**), we can train the model to locate the title in the claim. Furthermore, the words in the title co-occur with words in the article (e.g. *Barack* and *Michelle*), which the pre-trained BERT language model may be attuned to. We thus fine-tune a classifier on a dataset created from title and claim pairs (where positive examples are titles of gold evidence pages and negative are randomly sampled from our candidate set), obtaining 90.0% accuracy. Given the fine-tuned model, we extract features using Equation 1 where e_p is a title, and use Equation 3 to learn to predict a sequence of titles as in Figure 3.

[4]Initially, $h^{t,0}$ is set to z_t. v_h, W_g, and W_a are learned.

5.2 Joint Sentence Selection and Relation Prediction

The sentence selection and relation prediction tasks are closely linked, as predicting the correct evidence is necessary for predicting S or R and the representation should reflect the interaction between a claim and an evidence set. Conversely, if a claim and an evidence set are unrelated, the model should predict NEI. We thus jointly model this interaction by sharing the parameters of the pointer network - the hidden state of the decoder is used for both tasks and the models differ only by a final MLP.

Sentence Selection Similar to our document selection fine-tuning approach, we fine-tune a classifier on claim and evidence sentence pairs to obtain BERT embeddings. However, instead of training a binary classifier for the presence of valid evidence we train directly on veracity relation prediction, which is better suited for the end task. We create a dataset by pairing each claim with its set of gold evidence sentences. As gold evidence is not available for NEI relations, we sample sentences from our candidate documents to maintain a balanced dataset. We then fine-tune a BERT classifier on relation prediction, obtaining 93% accuracy. Given the fine-tuned model, we extract features using Equation 1 where e_p is a sentence, and use Equation 3 to learn to predict a sequence of sentences.

Relation Prediction In order to closely link relation prediction with evidence prediction, we reframe the task as a sequence labeling task. In other words, rather than make a single prediction given all evidence sentences, we make one prediction at every timestep during decoding to model the relation between the claim and *all evidence retrieved to that point*. This approach provides three benefits: it allows the model to better handle noise (when an incorrect evidence sentence is predicted), to handle multi-hop inference (to model the occurrence of switching from NEI to S/R), and to effectively provide more training data (for $k = 5$ timesteps we have 5 times as many relation labels). For the claim in Figure 3, the initial label sequence is NEI and R because the first evidence sentence by itself (the fact that Barack Obama was born in Hawaii) would not refute the claim. Furthermore for $k = 5$, the remaining sequence would be R, R, R, as additional evidence (guaranteed to be non-contradictory in FEVER) would not change the prediction. On the other hand, given a claim that requires only a single piece of evidence, such as that in Figure 1, the sequence would be R, R, R, R, R if the correct evidence sentence was selected at the first timestep, NEI, R, R, R, R if the correct evidence sentence was selected at the second timestep, and so forth.

We augment the evidence sentence selection described previously to use the hidden state of the pointer network after q hops (Equation 2) and an MLP to also predict a label at that time step, closely linking evidence and label prediction:

$$P(l_t) = \text{softmax}(W_{l2}\text{tanh}(W_{l1}h_t^o)) \quad (4)$$

As with evidence prediction (Equation 3), when the gold label sequence is available, the loss term is:

$$\mathcal{L}(\theta_{rel\_seq}) = -1/k \sum_{t=0}^{k-1} \log P_{\theta_{rel\_seq}}(l_t) \quad (5)$$

When training, at the current timestep we use both the gold evidence, i.e. "teacher forcing" (Williams and Zipser, 1989), and the model prediction from the previous step, so that we have training data for NEI. Combining Equations 3 and 5, our loss is:

$$\mathcal{L}(\theta) = \lambda\mathcal{L}(\theta_{ptr}) + \mathcal{L}(\theta_{rel\_seq}) \quad (6)$$

Finally, to predict a relation at inference, we ensemble the sequence of predicted labels by averaging the probabilities over every time step.[5]

Post-processing for Simple Temporal Reasoning As neural models are unreliable for handling numerical statements, we introduce a rule-based step to extract and reason about dates. We use the Open Information Extraction system of Stanovsky et al. (2018) to extract tuples. For example, given the claim *"The Latvian Soviet Socialist Republic was a republic of the Soviet Union 3 years after 2009,"* the system would identify **ARG0** as preceding the verb *was* and **ARG1** following. After identifying tuples in claims and predicted sentences, we discard those lacking dates (e.g. **ARG0**). Given more than one candidate sentence, we select the one ranked higher by the pointer network. Once we have both the claim and evidence date-tuple we apply one of three rules to resolve the relation prediction based on the corresponding temporal phrase. We either evaluate whether the evidence

[5]The subset of timesteps was determined empirically: while at the final timestep the model is likely to have seen the correct evidence it also contains more noise; in future work we will experiment with alternatives.

date is between two dates in the claim (e.g. *between/during/in*), we add/subtract x years from the date in the claim and compare to the evidence date (e.g. *x years/days before/after*), or compare the claim date directly to the evidence date (e.g. *before/after/in*). For the date expression *"3 years after 2009,"* we compare the year *2012* to the date in the retrieved evidence (*1991*, the year the USSR dissolved) and label the claim as R.

6 Experiments and Results

We evaluate our dataset and system as part of the FEVER 2.0 shared task in order to validate the vulnerabilities introduced by our adversarial claims (Section 4) and the solutions proposed by our system (Section 5). We train our system on FV1-train and evaluate on FV1/FV2-dev/test (Section 3). We report *accuracy* (percentage of correct labels) and *recall* (whether the gold evidence is contained in selected evidence at $k = 5$). We also report the *FEVER score*, the percentage of correct evidence sentences (for S and R) that also have correct labels, and *potency*, the inverse FEVER score (subtracted from one) for evaluating adversarial claims.

Our Baseline-RL: For baseline experiments, to compare different loss functions, we use the approach of Chakrabarty et al. (2018) for document selection and ranking, the reinforcement learning (RL) method of Chen and Bansal (2018) for sentence selection, and BERT (Devlin et al., 2019) for relation prediction. The RL approach using a pointer network is detailed by Chen and Bansal (2018) for extractive summarization, with the only difference that we use our fine-tuned BERT on claim/gold sentence pairs to represent each evidence sentence in the pointer network (as with our full system) and use the FEVER score as a reward. The reward is obtained by selecting sentences with the pointer network and then predicting the relation using an MLP (updated during training) and the concatenation of all claim/predicted sentence representations with their maximum/minimum pooling.

Hyper-parameters and settings for all experiments are detailed in Appendix B.

6.1 Adversarial Dataset Evaluation

We present the performance of our adversarial claims, obtained by submitting to the shared task server. We compare our claims to other participants in the FEVER 2.0 shared task (Table 2) and divided by attack type (Table 3). *Potency* was macro-averaged across different fact-checking systems (Thorne and Vlachos, 2019), correctness of labels was verified by shared task annotators, and adjusted potency was calculated by the organizers as the potency of correct examples. Compared to other participants (Table 2), we presented a larger set of claims (501 in dev and 499 in test). We rank second in adjusted potency, but we provided a more diverse set than those created by the organizers or other participants. The organizers (Thorne and Vlachos, 2019) created adversarial claims using simple pattern-matching and replacement, e.g. quantifiers and negation. Niewinski et al. (2019) trained a GPT-2-based model on the FEVER data and manually filtered disfluent claims. Kim and Allan (2019) considered a variety of approaches, the majority of which required understanding area comparisons between different regions or understanding implications (e.g. that "not clear" implies NEI). While GPT-2 is effective, our approach is controllable and targeted at real-world challenges. Finally, Table 3 shows that when we select our top 200 most effective examples (multi-hop reasoning and multi-hop temporal reasoning) and compare to the approaches of Niewinski et al. (2019) and Kim and Allan (2019) (who both provided less than 204 examples total) our potency is much higher. In particular, multi-hop reasoning has a potency of 88% for SUPPORT relations and 93% for REFUTES relations and multi-hop temporal reasoning obtains 98% for SUPPORT and REFUTES relations.

Team	#	Pot.	Corr.	Adj.
Organizer Baseline	498	60.34	82.33	49.68
Kim and Allan (2019)	102	79.66	64.71	51.54
Ours	**501**	**68.51**	**81.44**	**55.79**
Niewinski et al. (2019)	79	79.97	84.81	66.83

Table 2: The evaluation of our claims relative to other participants. **#:** Examples in blind test **Pot:** Potency score **Corr.:** Percent grammatical and coherent with correct label and evidence **Adj.:** Adjusted potency

6.2 Evaluation against State-of-the-art

In Tables 4 and 5 we compare Our System (Section 5) to recent work from teams that submitted to the shared task server for FEVER 1.0 and 2.0, respectively, including the results of Our Baseline-RL system in Table 5. Our dual pointer network approach obtains state-of-the-art results on the FEVER 1.0 blind test set (Table 4) on all measures even over systems designed specifically for evidence retrieval (Nishida et al., 2019; Zhou et al.,

Attack	M/A	#S/P	#R/P	#NEI/P
Conjunct.	A	-/-	54/55%	75/63%
Multi-hop	M	100/88%	88/93%	99/50%
Add. Unver.	M	-/-	-/-	50/50%
Date Man.	A	49/59%	129/80%	80/46%
Mul. Temp.	M	46/98%	5/98%	4/29%
Entity Dis.	M	46/50%	-/-	-/-
Lexical Sub.	A*	92/70%	57/70%	25/38%

Table 3: **Attack:** Type of attack as described in Section 4. **M/A:** Whether claims are generated manually (M), automatically (A), or verified manually (A*) **#S:** Support examples **#R:** Refute examples **#NEI** Not enough info examples **P:** Potency on Shared Task systems

2019), largely due to a notable improvement in recall (more than 3 points over the next system (Hanselowski et al., 2018)). We also find improvements in accuracy over the remaining pipeline systems, suggesting that joint learning helps. Compared to Our Baseline-RL, Our System has 1.8 point improvement in FEVER score on FV1-test with 4 points on FV2-test. Notably, our system finishes second (with a score of 36.61) on the FEVER 2.0 shared task test set, even though our claims were designed to be challenging for our model. The model of Malon (2018) performs especially well; they use a transformer-based architecture without pre-training but focus only on single-hop claims.

System	Acc.	Rec.	FEVER
Hanselowski et al. (2018)	65.46	85.19	61.58
Nishida et al. (2019)	69.30	76.30	61.80
Yoneda et al. (2018)	67.62	82.84	62.52
Nie et al. (2019a)	68.16	71.51	64.21
Tokala et al. (2019)	69.98	77.28	66.72
Zhou et al. (2019)	71.60	-	67.10
Our System	**72.47**	**88.39**	**68.80**

Table 4: Comparison with state of the art on FV1-test

Team	FV1-test	FV2-test
Hanselowski et al. (2018)	61.58	25.35
Nie et al. (2019a)	64.21	30.47
Our Baseline-RL	**67.08**	**32.92**
Stammbach and Neumann (2019)	68.46	35.82
Yoneda et al. (2018)	62.52	35.83
Our System	**68.80**	36.61
Malon (2018)	57.36	**37.31**

Table 5: Comparison of FEVER score to other shared-task systems (ordered by FV2-test FEVER score)

6.3 System Component Ablation

To better understand the improved performance of our system, we present two ablation studies in Tables 6 and 7 on FV1 and FV2 dev, respectively.[6]

Table 6 presents the effect of using different objective functions for sentence selection and relation prediction, compared to joint sentence selection and relation prediction in our full model. We compare Our System to Our Baseline-RL system as well as another baseline (*Ptr*). The *Ptr* system is the same as Our Baseline-RL, except the pointer network and MLP are not jointly trained with RL but independently using gold evidence and predicted evidence and relations, respectively. Finally, the Oracle upper bound presents the maximum possible recall after our document ranking stage, compared to 94.4% for Chakrabarty et al. (2018), and relation accuracy (given the MLP trained on 5 sentences guaranteed to contain gold evidence). We find that by incorporating the relation sequence loss, we improve the evidence recall significantly relative to the oracle upper-bound, reducing the relative error by 50% while also obtaining improvements on relation prediction, even over a strong RL baseline. Overall, the best model is able to retrieve 95.9% of the possible gold sentences after the document selection stage, suggesting that further improvements are more likely to come from document selection.

Model	Acc.	Rec.	FEVER
Oracle	84.2	94.7	–
Ptr	74.6	86.1	68.6
Our Baseline-RL	74.6	87.5	69.2
Our System	76.74	90.84	73.17

Table 6: Ablation experiments on FV1-dev

Table 7 evaluates the impact of the document pointer network and rule-based date handling on FV2-dev, as the impact of multi-hop reasoning and temporal relations is less visible on FV1-dev. We again compare Our Baseline-RL system to Our System and find an even larger 7.16 point improvement in FEVER score. We find that ablating the date post-processing (*-dateProc*) and both the date post-processing and document ranking components (*-dateProc,-docRank*) reduces the FEVER score by 1.45 and 3.5 points, respectively, with the latter largely resulting from a 5 point decrease in recall.

6.4 Ablation for Attack Types

While Table 3 presents the macro-average of all systems by attack type, we compare the performance of Our Baseline-RL and Our System in Table 8.

[6]Our system is significantly better on all metrics ($p < 0.001$ by the approximate randomization test).

System	Acc.	Rec.	FEVER
Our System	48.13	63.28	43.36
-dateProc	45.14	63.28	41.91
-dateProc,-docRank	44.29	58.32	39.86
Our Baseline-RL	44.04	57.56	36.2

Table 7: Ablation experiments on FV2-dev

Attack Type		Acc.	Rec.	FEVER
Conjunction	B	16.95	92.0	16.95
	S	**40.68****	92.0	**40.68****
Multi-hop	B	55.81*	29.07	19.77
	S	33.72	**45.35***	17.44
Add. Unver.	B	48.0	–	48.0
	S	**80.0****	–	**80.0****
Date Manip.	B	30.99	79.59	27.46
	S	**53.52*****	79.59	**42.25****
Multi-hop Temp.	B	3.33	10.34	0.0
	S	3.33	**13.79**	0.0
Entity Disamb.	B	70.83	62.5	58.33
	S	**79.17**	**79.17***	**70.83**
Lexical Sub.	B	33.33	65.71	25.0
	S	29.76	**75.71***	**26.19**

Table 8: Attack results for our FV2-dev claims. **B:** Our Baseline-RL, **S:** Our System. *: $p < 0.05$ **: $p < 0.01$ ***: $p < 0.001$ by approximate randomization test

Our System improves on evidence recall for multi-hop claims (indicating that a multi-hop document retrieval step may help) and those with ambiguous entities or words (using a model to re-rank may remove false matches with high lexical similarity). For example, the claim *"Honeymoon is a major-label record by Elizabeth Woolridge Grant."* requires multi-hop reasoning over entities. Our System correctly retrieves the pages **[Lana Del Rey]** and **[Honeymoon (Lana Del Rey album)]**, but Our Baseline-RL is misled by the incorrect page **[Honeymoon]**. However, while recall increases on multi-hop claims compared to the baseline, accuracy decreases, suggesting the model may be learning a bias of the claim or label distribution instead of relations between claims and evidence.

We also obtain large improvements on date manipulation examples (here a rule-based approach is better than our neural one); in contrast, multi-hop temporal reasoning leaves room for improvement. For instance, for the claim *"The MVP of the 1976 Canada Cup tournament was born before the tournament was first held,"* our full system correctly retrieves **[Bobby Orr]** and **[1976 Canada Cup]** (unlike the RL baseline). However, a further inference step is needed beyond our current capabilities – reasoning that Orr's birth year (1948) is before the first year of the tournament (1976).

Finally, we enhance performance on multi-propositions as conjunctions or additional unverifiable information (indicating that relation sequence prediction helps). Claims (non-verifiable phrase in brackets) such as *"Taran Killam is a [stage] actor."* and *"Home for the Holidays stars an actress [born in Georgia]."* are incorrectly predicted by the baseline even though correct evidence is retrieved.

7 Conclusion

We showed weaknesses in approaches to fact-checking via novel adversarial claims. We took steps towards realistic fact-checking with targeted improvements to multi-hop reasoning (by a document pointer network and a pointer network for sequential joint sentence selection and relation prediction), simple temporal reasoning (by rule-based date handling), and ambiguity and variation (by fine-tuned contextualized representations).

There are many unaddressed vulnerabilities that are relevant for fact-checking. The Facebook bAbI tasks (Weston et al., 2016) include other types of reasoning (e.g. positional or size-based). The DROP dataset (Dua et al., 2019) requires mathematical operations for question answering such as addition or counting. Propositions with causal relations (Hidey and McKeown, 2016), which are event-based rather than attribute-based as in FEVER, are also challenging. Finally, many verifiable claims are non-experiential (Park and Cardie, 2014), e.g. personal testimonies, which would require predicting whether a reported event was actually possible.

Finally, our system could be improved in many ways. Future work in multi-hop reasoning could represent the relation between consecutive pieces of evidence and future work in temporal reasoning could incorporate numerical operations with BERT (Andor et al., 2019). One limitation of our system is the pipeline nature, which may require addressing each type of attack individually as adversaries adjust their techniques. An end-to-end approach or a query reformulation step (re-writing claims to be similar to FEVER) might make the model more resilient as new attacks are introduced.

Acknowledgements

The authors thank Kathy McKeown, Chris Kedzie, Fei-Tzin Lee, and Emily Allaway for their helpful comments on the initial draft of this paper and the anonymous reviewers for insightful feedback.

References

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, Brussels, Belgium. Association for Computational Linguistics.

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.

Daniel Andor, Luheng He, Kenton Lee, and Emily Pitler. 2019. Giving BERT a calculator: Finding operations and arguments with reading comprehension. 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 5947–5952, Hong Kong, China. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642. Association for Computational Linguistics.

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, Brussels, Belgium. Association for Computational Linguistics.

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

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–686. 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, Minneapolis, Minnesota. Association for Computational Linguistics.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. 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 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.

Diane Francis. 2016. Fast furious fact check challenge.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking nli systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655.

Lucas Graves. 2018. Understanding the promise and limits of automated fact-checking. Technical report, Reuters Institute, University of Oxford.

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, Brussels, Belgium. Association for Computational Linguistics.

Naeemul Hassan, Fatma Arslan, Chengkai Li, and Mark Tremayne. 2017. Toward automated fact-checking: Detecting check-worthy factual claims by claimbuster. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, pages 1803–1812, New York, NY, USA. ACM.

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, Brussels, Belgium. Association for Computational Linguistics.

Christopher Hidey and Kathy McKeown. 2016. Identifying causal relations using parallel Wikipedia articles. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1424–1433, Berlin, Germany. Association for Computational Linguistics.

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

2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Shan Jiang and Christo Wilson. 2018. Linguistic signals under misinformation and fact-checking: Evidence from user comments on social media. *Proc. ACM Hum.-Comput. Interact.*, 2(CSCW):82:1–82:23.

Youngwoo Kim and James Allan. 2019. FEVER breaker's run of team NbAuzDrLqg. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, pages 99–104, Hong Kong, China. Association for Computational Linguistics.

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

Tsvetomila Mihaylova, Georgi Karadzhov, Pepa Atanasova, Ramy Baly, Mitra Mohtarami, and Preslav Nakov. 2019. Semeval-2019 task 8: Fact checking in community question answering forums. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 860–869.

Tsvetomila Mihaylova, Preslav Nakov, Lluís Màrquez, Alberto Barrón-Cedeño, Mitra Mohtarami, Georgi Karadzhov, and James R. Glass. 2018. Fact checking in community forums. *CoRR*, abs/1803.03178.

Paramita Mirza and Sara Tonelli. 2016. CATENA: causal and temporal relation extraction from natural language texts. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 64–75.

Ndapandula Nakashole and Tom M. Mitchell. 2014. Language-aware truth assessment of fact candidates. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1009–1019, Baltimore, Maryland. 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, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2019b. Adversarial nli: A new benchmark for natural language understanding. *arXiv preprint arXiv:1910.14599*.

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.

Kosuke Nishida, Kyosuke Nishida, Masaaki Nagata, Atsushi Otsuka, Itsumi Saito, Hisako Asano, and Junji Tomita. 2019. Answering while summarizing: Multi-task learning for multi-hop QA with evidence extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2335–2345, Florence, Italy. Association for Computational Linguistics.

Joonsuk Park and Claire Cardie. 2014. Identifying appropriate support for propositions in online user comments. In *Proceedings of the First Workshop on Argumentation Mining*, pages 29–38, Baltimore, Maryland. Association for Computational Linguistics.

Dean Pomerleau and Delip Rao. 2017. Fake news challenge.

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, Copenhagen, Denmark. Association for Computational Linguistics.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097, Florence, Italy. Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging nlp models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865.

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.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. Supervised open information extraction. 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 885–895.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2440–2448. Curran Associates, Inc.

James Thorne and Andreas Vlachos. 2017. An extensible framework for verification of numerical claims. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 37–40, Valencia, Spain. Association for Computational Linguistics.

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. Association for Computational Linguistics.

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

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819. Association for Computational Linguistics.

Santosh Tokala, G Vishal, Avirup Saha, and Niloy Ganguly. 2019. Attentivechecker: A bi-directional attention flow mechanism for fact verification. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2218–2222.

Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. 2016. Order matters: Sequence to sequence for sets. In *International Conference on Learning Representations (ICLR)*.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2692–2700. Curran Associates, Inc.

Andreas Vlachos and Sebastian Riedel. 2014. Fact checking: Task definition and dataset construction. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, pages 18–22. Association for Computational Linguistics.

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

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. 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 2153–2162, Hong Kong, China. Association for Computational Linguistics.

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.

Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. 2016. Towards ai-complete question answering: A set of prerequisite toy tasks. *CoRR*, abs/1502.05698.

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. Association for Computational Linguistics.

Ronald J. Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural Comput.*, 1(2):270–280.

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, Brussels, Belgium. Association for Computational Linguistics.

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, Brussels, Belgium. 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, Florence, Italy. Association for Computational Linguistics.

A Examples of Attack Types

Table 9 displays examples for each type of attack. The multi-propositional examples include attacks for CONJUNCTION, MULTI-HOP REASONING, and ADDITIONAL UNVERIFIABLE PROPOSITIONS. For temporal reasoning, we provide examples for DATE MANIPULATION and MULTI-HOP TEMPORAL REASONING. The lexical variation examples consist of ENTITY DISAMBIGUATION and LEXICAL SUBSTITUTION.

Attack Type	Example Claim	Label	Evidence
Conjunction	Blue Jasmine has Sally Hawkins acting in it **and** ~~Blue Jasmine~~ was filmed in San Francisco.	NEI	N/A
Multi-Hop Reasoning	Goosebumps was directed by ~~Rob Letterman~~ **the person who co-wrote Shark Tale**.	S	**[Goosebumps (film)]** It was directed by Rob Letterman, and written by Darren Lemke, based from a story by Scott Alexander and Larry Karaszewski. **[Rob Letterman]** Before Letterman's film subjects took him into outer space with Monsters vs. Aliens (2009), he was taken underwater, having co-directed and co-written Shark Tale.
Additional Unverifiable Propositions	Roswell is an American TV series **with 61 episodes**.	NEI	N/A
Date Manipulation	Artpop was Gaga's second consecutive number-one record in the United States ~~in 2009~~ **before 2010**.	R	**[Artpop]** Gaga began planning the project in 2011, shortly after the launch of her second studio album, Born This Way.
Multi-Hop Temporal Reasoning	Lisa Murkowski's father resigned from the Senate after serving as Senator.	S	**[Lisa Murkowski]** She is the daughter of former U.S. Senator and Governor of Alaska Frank Murkowski. Murkowski was appointed to the U.S. Senate by her father, Frank Murkowski, who resigned his seat in December 2002 to become the Governor of Alaska. **[Frank Murkowski]** He was a United States Senator from Alaska from 1981 until 2002 and the eighth Governor of Alaska from 2002 until 2006.
Entity Disambiguation	Kate Hudson is a left wing political activist	S	**[Kate Hudson (activist)]** Katharine Jane "Kate" Hudson (born 1958) is a British left wing political activist and academic who is the General Secretary of the Campaign for Nuclear Disarmament (CND) and National Secretary of Left Unity.
Lexical Substitution	The Last Song began ~~filming~~ **shooting** on Monday June 14th 2009.	R	**[The Last Song (film)]** Filming lasted from June 15 to August 18, 2009 with much of it occurring on the islands beach and pier.

Table 9: Examples of the seven sub-types of attacks. Claims edited with word substitution or insertion have their changes in bold. Deletions are marked in strikethrough. Wikipedia titles are represented in bold with square brackets. **S:** SUPPORTS **R:** REFUTES **NEI:** NOT ENOUGH INFORMATION

B Hyper-parameters and Experimental Settings

We select $M = 30$ Wikipedia articles using TF-IDF when combining with our other candidate document selection methods and select $D = 5$ after document ranking. We select $N = 5$ sentences during sentence selection, consistent with the shared task evaluation.

B.1 BERT Language Model Fine-Tuning

We use version 0.5.0 of the Huggingface library (https://github.com/huggingface/pytorch-pretrained-BERT) to fine-tune the "BERT-base" model using the default settings. We lowercase all tokens and use the default BERT tokenizer.

Document Ranking Our dataset of title and claim pairs (obtained from FV1-train) consists of 140,085 positive examples and 630,265 negative examples in training with approximately 10% set aside for validation (16,016 positive examples and 84,437 negative). As recommended by Devlin et al. (2019), we select hyper-parameters by grid search over 16 and 32 for batch size, 2e-5, 3e-5, and 5e-5 for learning rate, and 3 and 4 for the number of epochs.

Sentence Selection Our dataset of sentence and claim pairs (also obtained from FV1-train) consists of 54,431 S relations, 54,592 R relations, and 54,501 NEI relations in training, with approximately 10% set aside for validation (6,139 S relations, 5,984 R relations, and 6,050 NEI relations). We again select hyper-parameters consistent with the recommended best practice.

B.2 Pointer Network

We train both the document ranking and sentence selection pointer networks on FV1-train with the same hyper-parameters using Adagrad (Duchi et al., 2011) with a learning rate of 0.01, a batch size of 16, and a maximum of 140 epochs with early stopping on FV1-dev. The dimension of the pointer network LSTM hidden state is set to 200 with $q = 3$ hops over the memory. We use a beam width of 5 during inference. The MLP used to predict relations has a hidden layer dimensionality of 200 and we set $\lambda = 1$.

B.3 Reinforcement Learning

To make the sentence extractor an RL agent, we can formulate a Markov Decision Process (MDP): at each extraction step t, given a claim c, the agent observes the current state and samples an action from Equation 3 to extract a document sentence s, predict the relation label l and receive a reward $r(t+1) = \text{FEVER}(c, s, l)$. We train using REINFORCE, adapted with an Actor-Critic to minimize variance (detailed by Chen and Bansal (2018)). As RL often requires pre-training, we combine the pointer network loss from Equation 3 with the RL loss ($\mathcal{L}(\theta_{rl})$) and the relation prediction loss ($\mathcal{L}(\theta_{rel})$):

$$\mathcal{L}(\theta) = \lambda_1 \mathcal{L}(\theta_{ptr}) + \lambda_2 \mathcal{L}(\theta_{rl}) + \mathcal{L}(\theta_{rel}) \quad (7)$$

We set both $\lambda_1 = 1$ and $\lambda_2 = 1$.