The Role of Context in Detecting Previously Fact-Checked Claims

Anonymous ACL submission

Abstract

Recent years have seen the proliferation of disinformation and fake news online. Traditional proposals to mitigate these problems are manual and automatic fact-checking. Recently, another approach has emerged: checking whether the input claim has previously been fact-checked, which can be done automatically, and thus fast, while also offering credibility and explainability, thanks to the human fact-checking and explanations in the associated fact-checking article. Here we focus on claims made in a political debate, where context really matters. We study the impact of modeling the context of the claim: both on the source side, i.e., in the debate, as well as on the target side, i.e., in the fact-checking explanation document. We do this by modeling the local context, the global context, as well as by means of co-reference resolution, and multi-hop reasoning over the sentences of the document describing the fact-checked claim. The experimental results show that each of these represents a valuable information source, but that modeling the source-side context is more important, and can yield 10+ points of absolute improvement over a state-of-the-art model.

1 Introduction

The fight against the spread of dis/mis-information in social media has become an urgent social and political issue. Social media have been widely used not only for social good but also to mislead entire communities. Many fact-checking organizations, such as FactCheck.org, Snopes, PolitiFact, and FullFact, along with many others, and also along with some broader international initiatives such as the *Credibility Coalition* and *Eufactcheck*, have emerged in the past few years to address the issue (Stencel, 2019).

At the same time, there have been efforts to develop automatic systems to detect and to flag such content (Vo and Lee, 2018; Shu et al., 2017; Thorne and Vlachos, 2018; Li et al., 2016; Lazer et al., 2018; Vosoughi et al., 2018). Such efforts include the development of datasets (Hassan et al., 2015; Augenstein et al., 2019), systems, and evaluation campaigns (Barrón-Cedeño et al., 2020).

An important issue with automatic systems is that journalists and fact-checkers often question their credibility for reasons such as (perceived) insufficient accuracy given the state of present technology, but also due to the lack of explanation about how the system has made its decision. At the same time, manual fact-checking is time-consuming as it requires to go through several manual steps Vlachos and Riedel (2014) .

As both manual and automatic systems have their limitations, there have been also proposals of human-in-the-loop settings, aiming to bring the best of both worlds. In order to enable such an approach, one question that arises is how to facilitate fact-checkers and journalists with automated systems. An immediate interesting problem is to know whether a given input claim has been previously fact-checked by a reputable fact-checking organization. This would give them a credible reference and could save them significant amount of time and resources, as manually fact-checking a single non-trivial claim may take from 1-2 days to 1-2 weeks. Looking from a different perspective, at the time of COVID-19, we see the same false claims and conspiracy theories coming over and over again (e.g., about garlic water as a cure, about holding your breath for 10 seconds as a way to test for COVID-19, etc.). That is why fact-checking makes sense: to debunk such *frequent* claims. The problem is that next time they come in a slightly different form (although having the same meaning), it is important to be able to recognize them quickly and possibly to post a reply in social media with a link to a fact-checking article. If we consider a scenario in which a politician is being interviewed or is taking part in a debate, a quick response would

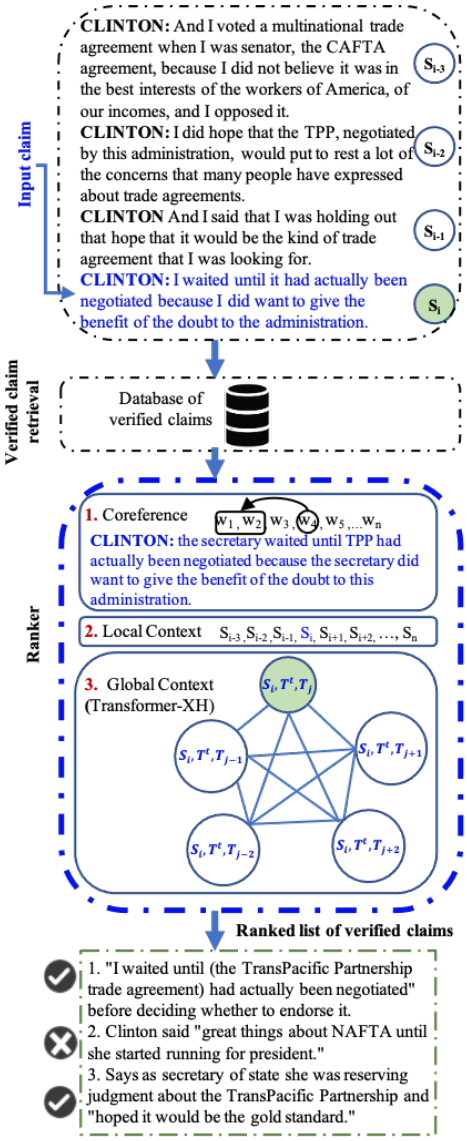

Figure 1: A pipeline of retrieving and ranking previously fact-checked claims. S_i is the claim (source), T^t is the title of the target, T_j is a sentence from the target.

make it possible to put him/her on the spot.

However, the problem in such a real-time scenario is that, unlike written text, interviews, debates and speeches are more spontaneous, and claims are often not clearly formulated in a single sentence. This is illustrated in Figure 1, where we can see a fragment from a Democratic debate for the 2016 US Presidential election, where Hillary Clinton said: "*I waited until it had actually been negotiated because I did want to give the benefit of the doubt to the administration.*" Understanding this claim requires pronominal co-reference resolution (e.g., what does *it* refer to, is it *CAFTA* or is it *TPP*, as both are mentioned in the previous sentences), more general co-reference (e.g., that the administration being discusses is the *Obama* administration), as well as a general understanding of the conversation so far, and possibly general world knowledge about US politics at the time of the debate (e.g., that Hillary Clinton was Secretary of State when TPP was being discussed).

Moreover, previous work has shown that it is beneficial to try to match the input claim not only against the canonical verified claim that fact-checkers worked with, but against the entire article that they wrote explaining why the claim was judged to be true/false (Shaar et al., 2020; Vo and Lee, 2020). This is because, in the fact-checking article, the claim is likely to be mentioned in different forms, and also a lot of background information and related terms would be mentioned, which can facilitate matching, and thus recall. This means that we need to exploit global contextual information contained within whole fact-checking articles or at least previous and following context of the claim (i.e., local context). Similarly, for the FEVER fact-checking task against Wikipedia, it has been shown that multi-hop reasoning (Transformer-XH) over the sentences of the target article can help (Zhao et al., 2019), an observation that was further confirmed in the context of fact-checking political claims (Ostrowski et al., 2020). Transformer-XH uses a novel attention mechanism that naturally "hops" across the connected text sequences in addition to attending over tokens within each sequence. As claims and their reasonings are manifested across documents, this hop-based attention mechanism constructs global contextualized representation to provide better joint multi-evidence reasoning. We rely on Transformer-XH to extract and use global contextual information.

Based on the above considerations, we propose a framework that focuses on modeling the co-reference, local context (features from neighboring sentences, see Section 4.2.2) and global context (features using Transformer-XH, see Section 4.2.3), both on the source and on the target side, while also using multi-hop reasoning over the target side.

Our contributions can be summarized as follows:

- We perform careful manual analysis to understand what makes detecting previously fact-checked claims a hard problem, and we categorize the claims by type. We release these annotations to enable further research.
- Unlike previous work, we focus on modeling the context both on the source side and on the target

side, both local and global, using co-reference resolution and reasoning with Transformer-XH, which yields sizable improvements over state-of-the-art models of over 10 MAP points absolute.
- We propose a realistic and challenging, time-sensitive and document-aware, data split compared to previous work, which we also release.

2 Related Work

Check-Worthiness Estimation Notable work in this direction includes context-aware approaches to detect check-worthy claims in political debates (Gencheva et al., 2017), using various patterns to find factual claims (Ennals et al., 2010), multi-task learning (Vasileva et al., 2019b), and a variety of other approaches used by the participants of the CLEF CheckThat! labs' shared tasks on checkworthiness (Nakov et al., 2018; Elsayed et al., 2019b,a; Vasileva et al., 2019a).

Previously Fact-Checked Claims While there is a surge in research to develop systems for automatic fact-checking, such systems suffer from credibility issues, e.g., in the eyes of journalists, and manual efforts are still the norm. Thus, it is important to reduce such manual effort by detecting when a claim has already been fact-checked. Work in this direction includes (Shaar et al., 2020) and (Vo and Lee, 2020): the former developed a dataset for the task and proposed a ranking model, while the latter proposed a neural ranking model using textual and visual modalities.

A recent work by Sheng et al. (2021) highlights the importance of lexical, semantic, and pattern-based information and proposes a re-ranker based on memory-enhanced transformers for matching (MTM) to detect and rank previously fact-checked claims.

Semantic Matching and Ranking Here we focus on the textual problem formulation of the task, as defined in the work of Shaar et al. (2020): given an input claim, we want to detect potentially matching previously fact-checked claims and to rank them accordingly. A related research area is semantic matching and ranking, as matching some *Input-Claim–VerClaim* pairs might require BERT-based sentence embeddings, natural language inference, and coreference resolution. An example of such a difficult pair is shown in Table 1, line 607. Recent relevant work in this direction uses neural approaches. Nie et al. (2019) proposed a semantic matching method that combines document retrieval, sentence selection, and claim verification neural models to extract facts and to verify them. Thorne et al. (2018) proposed a very simple model, where pieces of evidence are concatenated together and then fed into a Natural Language Inference (NLI) model. Yoneda et al. (2018) used a four-stage approach that combines document and sentence retrieval with NLI. Hanselowski et al. (2018) introduced Enhanced Sequential Inference Model (BiLSTM based) (Chen et al., 2016) methods to rank candidate facts and to classify a claim based on the selected facts. Several studies used model combination (i.e., document retrieval, sentence retrieval, and NLI for classifying the retrieved sentences) with joint learning (Yoneda et al., 2018; Hidey and Diab, 2018; Luken et al., 2018).

Context Modeling for Factuality Fact-checking is a complex problem. It requires retrieving pieces of evidence, which are often scattered in the document in different contexts. Once they are retrieved, they can be used to verify the claim. The evidence with contextual information can play a great role for fact verification and retrieval. Previous work has shown that the relation between the target statement and a context in the document (e.g., debate), the interaction between speakers, and the reaction of the moderator and the public can significantly help to find check-worthy claims (Gencheva et al., 2017). Liu et al. (2020) proposed a graph-based approach, a Kernel Graph Attention Network, to use evidence as context for fact verification. Similarly, Zhou et al. (2019) used a fully connected evidence graph with multi-evidence information for fact verification.

Since Transformer-based models have shown great success in many downstream NLP tasks, Zhong et al. (2020) used different pre-trained Transformer models and a graph-based approach (i.e., graph convolutional network and graph attention network) for fact verification. Zhao et al. (2019) introduced extra hop attention to incorporate contextual information, while maintaining the Transformer capabilities. The extra hop attention enables it to learn a global representation of the different pieces of evidence and to jointly reason over the evidence graph. It is a promising approach that uses contextual information as a graph representation and Transformer capabilities in the same model. One of the limitations is the need for human-labeled evidence in relation to the input claims in existing fact-verification datasets. The study by Ostrowski

et al. (2020) addressed this limitation by developing a dataset of annotated pieces of evidence associated with input claims and explored multihop attention mechanism, proposed in (Zhao et al., 2019), to make prtediction on the factuality of a claim.

Unlike the above work, here we target a different task: detecting previously fact-checked claims as opposed to performing fact-checking per se. Moreover, while the above work was limited to the target, we also model the source context (which turns out to be much more important).

3 Dataset

We focus on the task of detecting previously fact-checked claims, using the task formulation and also the data from (Shaar et al., 2020). They had two datasets: one on matching tweets against Snopes claims, and another one on matching claims in the context of a political debate to PolitiFact claims. Here, we focus on the latter,[1] and we perform a close analysis of the claims and what makes them easy/hard to match.

The dataset was collected from the US political fact-checking organization PolitiFact. After a US political debate, speech, or interview, fact-checking journalists would select few claims made in the event and would verify them either from scratch or by linking them to a previously fact-checked claim. Each previously fact-checked claim has an associated article stating its truthfulness along with a justification. The dataset has two parts: (*i*) verified claims {normalized *VerClaim*, article *title*, and article *text*}, (*ii*) transcripts of the political events (e.g., debates). They annotated the data by linking sentences from the transcript (*InputClaim*) to one or more verified claim (out of 16,636 claims).

To further analyze the dataset, we looked at the *InputClaim–VerClaim* pairs, and we manually categorized them into one of the following categories:

1. ***clean*** **:** A *clean* pair is a self-contained *InputClaim* with a *VerClaim* that directly verifies it (see line 255 in Table 1 for an example).
2. ***clean-hard*****:** A *clean-hard* pair is a self-contained *InputClaim* with a *VerClaim* that indirectly verifies it (see line 688 in Table 1).
3. ***part-of*****:** A *part-of*'s pair *InputClaim* is not self-contained and requires the addition of other sentences from the transcript to fully form a single claim.
4. ***context-dep*****:** A *context-dep* pair is similar to *clean* and *clean-hard*; however, the *InputClaim* is not self-contained and needs coreference.

These categories include all types of pairs we have seen. Moreover, since the dataset is constructed from speeches, debates, and interviews, the structure of the *InputClaim–VerClaim* pairs differs. For example, in debates, we see more *part-of* examples, as there are multiple questions–answers claims and back-and-forth arguments splitting the claims into multiple sentences.

The annotations were performed by three annotators who are experts in fact-checking (and co-authors of this paper), using the above definitions for the categories. We consolidated their annotations using majority voting, and they had a consolidation discussion for cases with no majority. The Fleiss Kappa inter-annotator agreement was 0.5, which corresponds to moderate agreement, which is reasonable for such a complex annotation task. Note that our agreement is much higher than for related tasks (Roitero et al., 2020): Krippendorff's α in [0.066; 0.131].

Table 1 shows examples of *InputClaim–VerClaim* pairs that demonstrate the above four categories. From the table, it is clear that due to the presence of cases like line 607 and 695–699, the task goes beyond simple textual similarity and natural language inference. Recognizing the *context-dep* pairs requires understanding the *InputClaim*'s local context, and recognizing the *clean-hard* pairs requires analysis of the overall global context of the *VerClaim*. While annotating the data into the four categories described in this section, we found out that a few *InputClaim-VerClaim* pairs in (Shaar et al., 2020) were false matches (which happened, as they did the matching automatically, without manually double-checking every single example) and we removed them. Thus, the reported number of pairs here is slightly lower, but it is also more accurate than in their work.

Table 2 gives statistics about the distribution the four categories of claims in the dataset. We can see that *clean* and *clean-hard* are the most frequent categories, while *part-of* is the least frequent one.

We also investigated previous work and observed that they dealt with each *InputClaim* independently, i.e., at the sentence level. That means two claims from the same debate can end up being in the training set and test set. This is problematic because if

[1] github.com/sshaar/That-is-a-Known-Lie

Line No.	Type		Input Claim	Verified Claim
255	*clean*	D. Trump:	*Hillary Clinton wanted the wall.*	Says Hillary Clinton "wanted the wall."
695	*part-of*	C. Wallas:	*And since then, as we all know, nine women have come forward and have said that you either groped them or kissed them without their consent.*	The stories from women saying he groped or forced himself on them "largely have been debunked."
⋮				
699	*part-of*	D. Trump:	*Well, first of all, those stories have been largely debunked.*	The stories from women saying he groped or forced himself on them "largely have been debunked."
688	*clean-hard*	D. Trump:	*She gave us ISIS as sure as you are sitting there.*	Hillary Clinton invented ISIS with her stupid policies. She is responsible for ISIS.
605		D. Trump:	*Now she wants to sign TransPacific Partnership.*	
⋮				
607	*context-dep*	D. Trump:	*She lied when she said she didn't call it the gold standard in one of the debates.*	Says Hillary Clinton called the TransPacific Partnership "the gold standard. You called it the gold standard of trade deals. You said its the finest deal youve ever seen."

Table 1: Fragment from the 3rd US Presidential debate in 2016 showing the *verified claims* chosen by PolitiFact and the fine-grained category of the pair. Most input sentences have no *verified claim*, e.g., see line 605.

	PolitiFact	
InputClaim–VerClaim pairs	695	
– *clean*	291	42%
– *clean-hard*	210	30%
– *part-of*	68	10%
– *context-dep*	126	18%
Total # of verified claims (to match against)	16,636	

Table 2: **Statistics about the dataset:** shown are the total number of *InputClaim–VerClaim* pairs and the total number of *VerClaims* to match an *InputClaim* against in the entire dataset.

Split	**MAP**
Debate-Level – Chrono	**0.429**
Debate-Level – Semi-chrono	0.539
Debate-Level – Random	0.590
Sentence-Level – Random (Shaar et al., 2020)	0.602

Table 3: MAP scores of the reranker models when using four different splits representing different scenarios. We use *Debate-Level – Chrono* for our experiments.

we have pairs that are categorized as *part-of*, we could end up splitting them and putting them in different sets, i.e., train and test.

Moreover, splitting the dataset in this manner has another implication: the discussed topics in the input claim can fall into both training and test sets.

To avoid such issues, we can split the data in different settings that reflects various scenarios:

- *Debate-Level Chrono*: We split the data chronologically. We use the first 50 debates for training and the last 20 for testing. Specifically, we have 554 pairs for training, and 141 pairs for testing. This is a more realistic scenario, where we would only have access to earlier debates, and we can use them to make decisions about claims made in future debates. The complexity of this setting is also reflected in the MAP score as shown in Table 3. We see that this score is lower than the best model in the previous work (last row). This is because this setting is complex as we use a model trained on debates and speeches from 2012-2018, and we test on debates from 2019. Across those different time frames, different politicians discuss different topics.
- *Debate-Level Semi-Chrono*: We split the data per year, e.g., for year 2018, we divide the transcripts into train and test with 80/20 splits, and then we train and evaluate using the same reranking model. In Table 3, we can see an improvement with this setting compared to the *Debate Level Chrono* setting. This might be because the same politicians discuss same/similar issues throughout the same year.
- *Debate-Level Random*: We randomly choose 80% of the debates for training and the remaining ones for testing. This is a comparatively easier setting as the data is randomly distributed in training and testing. This is also reflected in the results in Table 3. The reason could be that politi-

cians repeat themselves a lot, especially in two consecutive political events, and the random split can lead to having two similar debates/speeches in two splits.

- *Sentence Level Random*: This is the setting used in (Shaar et al., 2020), where *sentences* from the debates are randomly divided into train and test set with 80% and 20% proportion, respectively. This is the most unrealistic split.

In the rest of the experiments, we choose to use the more realistic setup *Debate Level Chrono*, which means that our baseline MAP score (which is in fact the state-of-the-art from previous work) goes down from 0.602 to 0.429.

4 Experimental Setup

4.1 Baseline

From our analysis of the dataset (described in Section 3), we conclude that (*i*) we need to resolve the references in the *InputClaim*, (*ii*) to capture the local context of the *InputClaim*, and (*iii*) to encapsulate the global context of the *VerClaim*.

For the baseline, we use the same setup as in the state-of-the-art model of Shaar et al. (2020). We use the claim as a query against the full text of the documents using BM25 (a hard-to-beat model from information retrieval). We then train a reranker on the top-100 results returned by BM25 using rankSVM (Herbrich et al., 1999) with an RBF kernel. The reranker uses nine similarity measures that compare the *InputClaim* to the *VerClaim*, as well as the respective reciprocal ranks. In particular, we compute the BM25 score for *InputClaim* vs. *VerClaim*, *title*, *text*, *VerClaim+title+text*. We also compute the cosine using sentence-BERT embeddings for *InputClaim* vs. *VerClaim*, *title*, and the top-4 sentences from *text*. Using these scores, we create a vector representation of the *InputClaim–VerClaim* pair with dimensionality $\mathbb{R}^{18}$. We then scale the vectors of all *InputClaim–VerClaim* pairs in $[-1; 1]$ and we train a rankSVM with the default parameters ($KernelDegree = 3$, $\gamma = 1/num\_features$, $\epsilon = 0.001$).

4.2 Proposed Models

As shown in Figure 1, our model uses co-reference resolution on the source and on the target side, the local context (i.e., neighboring sentences as context), and the global context (Transformer-XH) as discussed below. It is still a pairwise reranker, but with a richer context representation.

4.2.1 Co-reference Resolution

We manually inspected the training transcripts and the associated verified claims, and we realized that there were many co-reference dependencies. Thus, resolving them can help to obtain more representative textual and contextual similarity scores. As for the verified claims, we noticed that not all *VerClaim* were self-contained, and that some understanding of the context was needed[2] from the article's *text* that explains the verdict provided by the PolitiFact journalists. Therefore, our hypothesis is that resolving such co-references should improve the downstream matching scores. For the same reason, we also performed co-reference resolution on the PolitiFact articles when they were used to compute the BM25 scores.

We explored different co-reference models such as **NeuralCoref**, [3] **e2e-coref** [4] and **SpanBERT** [5]. We found that **NeuralCoref** model performed best on the transcripts, while **e2e-coref** was best on the *VerClaim*s. Hence, in the rest of the experiments, we show results using **NeuralCoref** for the source side, and **e2e-coref** for the target side.

We resolved the co-reference in the *InputClaim* by performing co-reference resolution on the entire input transcript (as was suggested in the literature); we will refer to this approach as ***src-coref***. As for the verified claims, we aimed to resolve the co-references in both the *VerClaim* and the *text* of the PolitiFact articles. We also aimed to ensure that the dependencies from the *text* can be used for the *VerClaim*. Therefore, we concatenated both the text and *VerClaim* (in the same order), and we applied the co-reference model on the concatenated text. We choose this order of concatenation because the published *text* reserves the last paragraph to rephrase the *VerClaim* and to provide a summary of the justification; hence, there is a higher probability to resolve the co-references correctly.

4.2.2 Local Context

Resolving co-references allows us to obtain the correct objects and names the *InputClaim* is referring to. However, by analyzing the dataset, we noticed that different *VerClaim*s, although having similar structure, could be talking about different things, depending on the article text and the surrounding context. Therefore, it is important to understand

[2] For example, who is speaking or what is being discussed.
[3] github.com/huggingface/neuralcoref
[4] github.com/kentonl/e2e-coref
[5] github.com/facebookresearch/SpanBERT

the context of an *InputClaim*. We achieve this by doing a feature-level concatenation of the neighboring sentences in the transcript, i.e., we take the 18 features ($\mathbb{R}^{18}$, as discussed in Section 4.1) for the neighboring sentences, and we concatenate them to the similarity score for the *InputClaim*. We then use that as a feature vector for the reranker. For example, if we take three sentences before the *InputClaim* and one sentence after, then, we denote this as ***FC***(3, 1).

Let S_i be our *InputClaim*, which is the i'th sentence in the transcript. We compute the similarity measures and the reciprocal rank (as described in Section 4.1) to obtain the vector representation $S_{i,v}$ for S_i. With $k = 3$ previous and $l = 1$ following neighbouring sentences our final feature vector is

$$FC(k = 3, l = 1) = S_{i-3,v} + S_{i-2,v} + S_{i-1,v} + S_{i,v} + S_{i+1,v} \quad (1)$$

where $+$ represents concatenation. After the concatenation, the resulting dimension of the feature vector is $18 \times (3 + 1 + 1) = 90$ for ***FC(3, 1)***.

4.2.3 Global Context

The similarity scores leveraging the local context are obtained from the textual content of the *InputClaim* and the *VerClaim* (*i*) using BM25, (*ii*) cosine similarity between the Sentence-BERT embeddings of *InputClaim* vs. the top-4 sentences of the *VerClaim*. This might miss relevant information further away from the *InputClaim* in the input document and further away from *VerClaim* in the document accompanying the *VerClaim*. We refer to such scattered information as **global context**. To capture it, we adapt a graph-based Transformer, Transformer-XH (Zhao et al., 2019). In particular, we use a Transformer-XH model pretrained on the FEVER (Fact Extraction and VERification) dataset, which is trained to predict whether a given input claim is supported/refuted by a set of target sentences (from Wikipedia), represented as a graph, or there is no enough information. We used the model that is publicly made available by (Zhao et al., 2019). For a given *InputClaim*, we generate a graph for each of the top-100 *VerClaim*s retrieved from the BM25 algorithm using the normalized claim, the *title* and the top-3 sentences from the *text* as nodes. Using the ***Transformer-XH*** model on the graph, we obtain three additional scores that correspond to the posterior probability that *VerClaim* supports or refutes the *InputClaim*, or there is no enough information.

4.3 Hyper-Parameter Values

For the baseline, we use the best values of the hyper-parameters as found in (Shaar et al., 2020). For our context-aware models, we select the values of the hyper-parameters by splitting the training dataset into train-train (debates from 2012-2017) and train-dev (debates from 2018), then training on train-train, and testing on train-dev.

4.4 Evaluation Measures

As we have a ranking task, we use mean average precision (MAP). It is a suitable score as some *InputClaim*s have more than one *VerClaim* paired to them. This is why we opted for not using mean reciprocal rank (MRR), which would only pay attention to the rank of the highest-ranked match.

5 Results

5.1 Source-Side Experiments

For the source side experiments, we used co-reference resolution on transcripts and variations of the local context by varying k and l in Eq. 1.

When we inspected the transcripts, we found that co-references tend to be resolved by a few sentences before the *InputClaim*; therefore, we tried ***FC***(1, 1), ***FC***(3, 1), ***FC***(3, 3), and ***FC***(5, 1). We obtained the best results (on cross-validation) using ***FC***(3, 1), which we use in this study. As shown in Table 4, local context (Line 2) has improved over the baseline (Line 1) by 8 MAP points absolute.

We then experiment using co-reference resolution with the **NeuralCoref** model. Compared to the baseline, we have a sizable improvement using co-reference resolution as shown in line 3, in Table 4. Specifically, in *part-of* and *context-dep*, because those pairs have many co-references that confuses the *InputClaim*. After combining both methods, i.e., *src-coref* and *FC(3,1)* (Line 4), we achieved the highest MAP score of 0.532.

As expected, we always see an increase in the performance for the *clean* category as the resolved *InputClaim* can match the article text better.

5.2 Target-Side Experiments

For the target side experiments, we investigate the co-references in the *VerClaim* and their documents and modeling the global context with *(Transformer-XH)*. Compared to the baseline, we see a sizable improvement (from 0.365 to 0.441) in *clean-hard* as shown in line 5 in Table 4.

Line No.	Model	Overall	*clean*	*clean-hard*	*part-of*	*context-dep*
1	Baseline	0.429	0.661	0.365	0.161	0.375
	Source-Side Experiments: Co-reference Resolution, Local Context					
2	***FC***(3, 1)	0.513	0.690	**0.485**	0.305	0.448
3	src-coref	0.479	0.667	0.408	0.286	0.429
4	src-coref + ***FC***(3, 1)	**0.532**	0.695	0.452	**0.385**	**0.485**
	Target-Side Experiments: Co-reference Resolution, Global Context					
5	*Transformer-XH*	0.468	0.680	0.441	0.226	0.384
6	tgt-coref	0.443	0.673	0.422	0.182	0.339
7	tgt-coref + *Transformer-XH*	0.458	0.702	0.444	0.161	0.357
	Source+Target-Side Experiments: Co-reference Resolution, Local Context, Global Context					
8	src-coref + tgt-coref	0.487	0.672	0.440	0.291	0.411
9	All	0.517	**0.749**	0.389	0.321	0.464

Table 4: MAP Scores of the reranker models on the test set using the *Debate Level – Chrono*.

This is expected as the pair does not have much semantic similarity, and we need to build our own understanding of the *text* of the *VerClaim* in order to capture the contextual similarity in the pair. We also experiment with co-reference resolution on the *VerClaim* and the *text* of the *VerClaim* and also see some improvement. Combining *tgt-coref* and *(Transformer-XH)* (line 7) improved the performance over *tgt-coref* alone, but it under-performs *(Transformer-XH)* alone. The combination outperforms other target-side experiments on *clean* type.

5.3 Source-Side & Target-Side Experiments

Eventually, we tried to combine modeling the source and the target side. Line 8 in Table 4 shows a result when we use both source and target co-reference resolution. We can see that this yields better overall MAP score of 0.487, compared to using source-side (MAP of 0.479; line 3) or target-side only (MAP of 0.443; line 6). Moreover, co-reference resolution on both the source and target improves *clean-hard* and *part-of* pairs (compared to using co-reference on one side only) as they require better local and global context, respectively.

We further tried putting it all together, and the result is shown in line 9.[6] While this yielded better results for *clean*, it was slightly worse compared to the source-side context modeling combination, in line 4. This is probably due to source-side context models being generally stronger than target-side ones (compare lines 2–3 to lines 5–6).

We can conclude that modeling the context on the source side is much more important than on the target side. This is expected for political debates, which are conversational in nature. In contrast, the target side is a well-written journalistic article, where sentences are much more self-contained. Thus, features from the source side (i.e., from the debate) are more useful as can be seen in Table 4.

Comparison to Previous Work As mentioned above, our baseline is a reimplementation of the best system of Shaar et al. (2020), and our context modeling adds additional components on top of it. Note, however, that our results are not directly comparable to their work, as we use a more realistic and also a much harder setup, where the data is split by entire debates and also chronologically, i.e., training on the data from 2012 to 2018 and testing on 2019 (while they split all debates into sentences and randomly distribute them to training/testing).

6 Conclusion and Future Work

We have presented our work on the problem of detecting previously fact-checked claims in political debates. In particular, we studied the impact of modeling the context of the claim: both on the source side, i.e., in the debate, as well as on the target side, i.e., in the fact-checking explanation document. We did this by modeling the local context, the global context, as well as by means of co-reference resolution, and reasoning over the target text using Transformer-XH. The experimental results have shown that each of these represents a valuable information source, however, modeling the source-side context is more important, and can yield 10+ points of absolute improvement.

In future work, we plan to experiment with other language models, and also to apply our approach to other domains and languages, and tasks.

[6]Note that in this result we did not use target-side co-reference, as adding it yielded somewhat worse results. It seems to interact badly with Transformer-XH, which can also be seen by comparing lines 5 and 7.

Ethics and Broader Impact

Biases We note that there might be some biases in the data we use, as well as in some judgments for claim matching. These biases, in turn, will likely be exacerbated by the unsupervised models trained on them. This is beyond our control, as the potential biases in pre-trained large-scale transformers such as BERT and RoBERTa, which we use in our experiments.

Intended Use and Misuse Potential Our models can make it possible to put politicians on the spot in real time, e.g., during an interview or a political debate, by providing journalists with tools to do trustable fact-checking in real time. They can also save a lot of time to fact-checkers for unnecessary double-checking something that was already fact-checked. However, these models could also be misused by malicious actors. We, therefore, ask researchers to exercise caution.

Environmental Impact We would also like to warn that the use of large-scale Transformers requires a lot of computations and the use of GPUs/TPUs for training, which contributes to global warming (Strubell et al., 2019). This is a bit less of an issue in our case, as we do not train such models from scratch; rather, we fine-tune them on relatively small datasets. Moreover, running on a CPU for inference, once the model is fine-tuned, is perfectly feasible, and CPUs contribute much less to global warming.

References

Isabelle Augenstein, Christina Lioma, Dongsheng Wang, Lucas Chaves Lima, Casper Hansen, Christian Hansen, and Jakob Grue Simonsen. 2019. MultiFC: A real-world multi-domain dataset for evidence-based fact checking of claims. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4685–4697, Hong Kong, China. Association for Computational Linguistics.

Alberto Barrón-Cedeño, Tamer Elsayed, Preslav Nakov, Giovanni Da San Martino, Maram Hasanain, Reem Suwaileh, Fatima Haouari, Nikolay Babulkov, Bayan Hamdan, Alex Nikolov, Shaden Shaar, and Zien Sheikh Ali. 2020. Experimental ir meets multilinguality, multimodality, and interaction proceedings of the eleventh international conference of the clef association (clef 2020). In *Overview of CheckThat! 2020: Automatic Identification and Verification of Claims in Social Media*, LNCS (12260). Springer.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2016. Enhanced lstm for natural language inference. *arXiv preprint arXiv:1609.06038*.

Tamer Elsayed, Preslav Nakov, Alberto Barrón-Cedeño, Maram Hasanain, Reem Suwaileh, Giovanni Da San Martino, and Pepa Atanasova. 2019a. CheckThat! at CLEF 2019: Automatic identification and verification of claims. In *Advances in Information Retrieval*, ECIR '19, pages 309–315. Springer International Publishing.

Tamer Elsayed, Preslav Nakov, Alberto Barrón-Cedeño, Maram Hasanain, Reem Suwaileh, Giovanni Da San Martino, and Pepa Atanasova. 2019b. Overview of the CLEF-2019 CheckThat!: Automatic identification and verification of claims. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, LNCS, pages 301–321. Springer.

Rob Ennals, Dan Byler, John Mark Agosta, and Barbara Rosario. 2010. What is disputed on the web? In *Proceedings of the 4th Workshop on Information Credibility*, WICOW '10, pages 67–74, New York, NY, USA.

Pepa Gencheva, Preslav Nakov, Lluís Màrquez, Alberto Barrón-Cedeño, and Ivan Koychev. 2017. A context-aware approach for detecting worth-checking claims in political debates. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 267–276.

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. *arXiv preprint arXiv:1809.01479*.

Naeemul Hassan, Chengkai Li, and Mark Tremayne. 2015. Detecting check-worthy factual claims in presidential debates. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pages 1835–1838.

R. Herbrich, T. Graepel, and K. Obermayer. 1999. Support vector learning for ordinal regression. In *1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470)*, volume 1, pages 97–102 vol.1.

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.

David M.J. Lazer, Matthew A. Baum, Yochai Benkler, Adam J. Berinsky, Kelly M. Greenhill, Filippo

Menczer, Miriam J. Metzger, Brendan Nyhan, Gordon Pennycook, David Rothschild, Michael Schudson, Steven A. Sloman, Cass R. Sunstein, Emily A. Thorson, Duncan J. Watts, and Jonathan L. Zittrain. 2018. The science of fake news. *Science*, 359(6380):1094–1096.

Yaliang Li, Jing Gao, Chuishi Meng, Qi Li, Lu Su, Bo Zhao, Wei Fan, and Jiawei Han. 2016. A survey on truth discovery. *SIGKDD Explor. Newsl.*, 17(2):1–16.

Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2020. Fine-grained fact verification with kernel graph attention network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7342–7351, Online. Association for Computational Linguistics.

Jackson Luken, Nanjiang Jiang, and Marie-Catherine de Marneffe. 2018. Qed: A fact verification system for the fever shared task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 156–160.

Preslav Nakov, Alberto Barrón-Cedeño, Tamer Elsayed, Reem Suwaileh, Lluís Màrquez, Wajdi Zaghouani, Pepa Atanasova, Spas Kyuchukov, and Giovanni Da San Martino. 2018. Overview of the CLEF-2018 CheckThat! lab on automatic identification and verification of political claims. In *Proc. of the Ninth International Conference of the CLEF*, Lecture Notes in Computer Science, pages 372–387, Avignon, France. Springer.

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

Wojciech Ostrowski, Arnav Arora, Pepa Atanasova, and Isabelle Augenstein. 2020. Multi-hop fact checking of political claims. *arXiv preprint arXiv:2009.06401*.

Kevin Roitero, Michael Soprano, Shaoyang Fan, Damiano Spina, Stefano Mizzaro, and Gianluca Demartini. 2020. Can the crowd identify misinformation objectively? The effects of judgment scale and assessor's background. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, pages 439–448, Virtual Event, China. Association for Computing Machinery.

Shaden Shaar, Nikolay Babulkov, Giovanni Da San Martino, and Preslav Nakov. 2020. That is a known lie: Detecting previously fact-checked claims. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3607–3618, Online. Association for Computational Linguistics.

Qiang Sheng, Juan Cao, Xueyao Zhang, Xirong Li, and Lei Zhong. 2021. Article reranking by memory-enhanced key sentence matching for detecting previously fact-checked claims. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5468–5481, Online. Association for Computational Linguistics.

Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *SIGKDD Explor. Newsl.*, 19(1):22–36.

Mark Stencel. 2019. Number of fact-checking outlets surges to 188 in more than 60 countries. *Duke Reporters' LAB*, pages 12–17.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

James Thorne and Andreas Vlachos. 2018. Automated fact checking: Task formulations, methods and future directions. In *Proc. of the 27th COLING*, COLING '18, pages 3346–3359, Santa Fe, NM, USA.

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, New Orleans, Louisiana. Association for Computational Linguistics.

Slavena Vasileva, Pepa Atanasova, Lluís Màrquez, Alberto Barrón-Cedeño, and Preslav Nakov. 2019a. It takes nine to smell a rat: Neural multi-task learning for check-worthiness prediction. *arXiv preprint arXiv:1908.07912*.

Slavena Vasileva, Pepa Atanasova, Lluís Màrquez, Alberto Barrón-Cedeño, and Preslav Nakov. 2019b. It takes nine to smell a rat: Neural multi-task learning for check-worthiness prediction. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, RANLP '19, Varna, Bulgaria.

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, Baltimore, MD, USA. Association for Computational Linguistics.

Nguyen Vo and Kyumin Lee. 2018. The rise of guardians: Fact-checking url recommendation to combat fake news. In *The 41st International ACM*

SIGIR Conference on Research & Development in Information Retrieval, pages 275–284.

Nguyen Vo and Kyumin Lee. 2020. Where are the facts? searching for fact-checked information to alleviate the spread of fake news. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7717–7731, Online. 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.

Takuma Yoneda, Jeff Mitchell, Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Ucl machine reading group: Four factor framework for fact finding (hexaf). In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 97–102.

Chen Zhao, Chenyan Xiong, Corby Rosset, Xia Song, Paul Bennett, and Saurabh Tiwary. 2019. Transformer-XH: Multi-evidence reasoning with extra hop attention. In *International Conference on Learning Representations*.

Wanjun Zhong, Jingjing Xu, Duyu Tang, Zenan Xu, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2020. Reasoning over semantic-level graph for fact checking. *arXiv preprint 1909.03745*.

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. *arXiv preprint arXiv:1908.01843*.