“Who said it, and Why?”
Provenance for Natural Language Claims

Yi Zhang **Zachary G. Ives** **Dan Roth**
Department of Computer and Information Science
University of Pennsylvania
{yizhang5, zives, danroth}@cis.upenn.edu

Abstract

In an era where generating content and publishing it is so easy, we are bombarded with information and are exposed to all kinds of claims, some of which do not always rank high on the truth scale. This paper suggests that the key to a longer-term, holistic, and systematic approach to navigating this information pollution is capturing the *provenance* of claims. To do that, we develop a formal definition of *provenance graph* for a given natural language claim, aiming to understand where the claim may come from and how it has evolved. To construct the graph, we model *provenance inference*, formulated mainly as an information extraction task and addressed via a textual entailment model. We evaluate our approach using two benchmark datasets, showing initial success in capturing the notion of provenance and its effectiveness on the application of claim verification.

1 Introduction

Never before have humans been able to generate and disseminate content so easily, leading to a contamination of information supply with irrelevant, redundant, unsolicited, and often low-value information (Orman, 1984). While significant attention has been devoted recently to identifying false claims, the age of “information pollution” we live in calls for the development of additional important insights. At the heart of these insights is the need to determine the *provenance* of claims — who first made a given claim, and how an original claim developed and changed over time (and potentially across contributors).

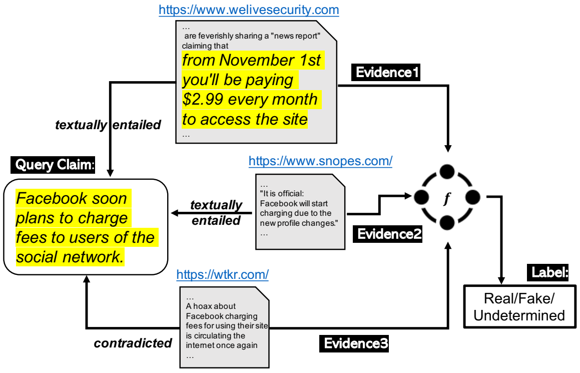
https://www.welivesecurity.com
...
are feverishly sharing a "news report" claiming that
from November 1st you'll be paying $2.99 every month to access the site
...
Evidence1
textually entailed
Query Claim:
Facebook soon plans to charge fees to users of the social network.
https://www.snopes.com/
...
"It is official: Facebook will start charging due to the new profile changes."
...
textually entailed
Evidence2
f
Label:
Real/Fake/ Undetermined
https://wtkr.com/
...
A hoax about Facebook charging fees for using their site is circulating the internet once again
...
contradicted
Evidence3

Figure 1: A typical claim verification pipeline of our running example. It starts with searching for evidence, and feed it to a textual entailment model f to decide the claim’s veracity.

Consider the following claim: *“Facebook soon plans to charge fees to users of the social network.”*[1] As shown in Figure 1, a typical modern claim verification pipeline starts with searching for existing evidence for the given *query claim*, and then leverages textual entailment models to determine the veracity of the claim relative to the evidence (Thorne et al., 2018). However, sites such as `snopes.com` and other fact-checking websites will not only provide their conclusion about the veracity of the claim relative to the evidence, but would also seek additional information that explains why people may think it fake. For example, Snopes details how the claim originated from `nationalreport.net`. The original version of the claim is related to the query claim, as well as other relevant claims, but carries a different meaning. It says: *“Facebook could cease to exist, if they don’t do something about their rising costs”*. Subsequently, the inaccurate claim, *triggered* by the original one, has been repeated by other websites and retweeted on social media, as shown in Figure 2, possibly increasing the level of credibility some readers assign to it.

The origins and causal derivations of data, as described above, are explicitly modeled in the context of databases (Cheney et al., 2009) and scientific workflow systems (Davidson and Freire, 2008), where they are termed “data provenance.” We argue

[1] `https://www.snopes.com/fact-check/facebook-implementing-user-fees/`

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 4416–4426
July 5 - 10, 2020. ©2020 Association for Computational Linguistics

that modeling and understanding the provenance of a claim made in natural language is also very important since, beyond attribution, it helps people understand the background and the context in which a claim was generated, how different aspects of the claim are combined, and how a claim has been changed over time by different agents. At the same time, provenance provides us with an explanation for *why* people think a claim is real or fake, by looking at its history. Even if all one wants is to determine a stance relative to a claim, this may involve considering more than just its current incarnation, but rather its evolution over time and all of the sources that contributed to this evolution. Similarly, one may want to consider *who* influenced a claim, or who influences a specific author of multiple claims, and this can be accomplished by considering the origin and evolution of these claims. Figure 2 shows that our notion of provenance can not only provide us with evidence but also with the structure of- and relationships among supporting evidence and claims.

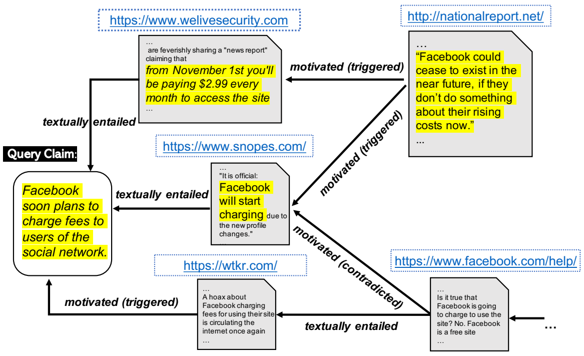
https://www.welivesecurity.com
...
are feverishly sharing a "news report" claiming that
from November 1st you'll be paying $2.99 every month to access the site
...
http://nationalreport.net/
...
"Facebook could cease to exist in the near future, if they don't do something about their rising costs now."
...
motivated (triggered)
textually entailed
Query Claim:
Facebook soon plans to charge fees to users of the social network.
https://www.snopes.com/
...
"It is official: Facebook will start charging due to the new profile changes."
...
textually entailed
motivated (triggered)
motivated (contradicted)
https://wtkr.com/
...
A hoax about Facebook charging fees for using their site is circulating the internet once again
...
https://www.facebook.com/help/
...
Is it true that Facebook is going to charge to use the site? No. Facebook is a free site
...
motivated (triggered)
textually entailed
...

Figure 2: A provenance subgraph of our running example. The nodes represent information sources, each with a statement that *influenced* the target claim. The edges are in the direction of influence, and labels indicate the relations between the corresponding statements.

In this paper, we propose and develop a computational framework for *claim provenance graphs*, which provide information and supporting evidence about where a claim is believed to have originated and how it has been disseminated. Our challenge is to *infer and reconstruct* this graph using available evidence. A claim provenance graph consists of two components:

1. As nodes: the sources that may have made the query claim and earlier versions of it, or those influencing the eventual query claim;

2. As labeled edges: the relationships between the claims made by sources.

Like provenance graphs in other fields (including the W3C PROV specification (Belhajjame et al., 2013)), a claim provenance graph tracks the data, operations, and parties responsible for a claim. Unlike most prior provenance graphs, claim provenance is often inferred, uncertain, and comprised of approximate relationships (e.g., "textually entailed"), as indicated in Figure 2.

However, inferring the provenance graph of a claim is a difficult task. In our current implementation of this notion, given a natural language claim in a document, we search for the claim on the web, restricting our focus to content published prior to the document (eliminating many sources that could not have influenced the document). A match to the claim search may itself make a statement about the claim, or it may in turn report a statement relevant to the claim made by other sources. If a source mentioned in the article is describing the claim, one of the sub-tasks is to identify the correct source(s). Therefore, we view obtaining the nodes of the provenance graph as an information extraction (IE) problem. However, in contrast to a typical IE approach that uses *annotated data* (Hendrickx et al., 2009), Wikipedia or other large scale knowledge bases (Auer et al., 2007), identifying sources of a statement in an article is an IE task which is very hard to annotate. The reason is that both the statement and its sources can be described *implicitly* in the given text, and this may require additional reasoning or coreference resolution. In this work, we tackle this IE problem as a textual entailment (TE) problem, and propose a solution that leverages off-the-shelf semantic role labeling tools to generate candidates for source identification. Following that, we wikifiy extracted source mentions, which further allows us to link nodes in the provenance graph and label them. As an application, we propose models that can use the provenance graph to improve the estimation of claims' veracity.

The key contributions of this paper are (1) it is the first work to study and formally define the notion of a *provenance graph* for a natural language claim; (2) it proposes a TE model to automatically extract provenance information, regardless of whether the relevant statement and the source are described explicitly or implicitly in the text; this is then used to construct a graph and label its edges; (3) it develops techniques that exploit the provenance graph to improve claim verification. We provide initial experimental support for our novel formulation by studying the effectiveness of extracting sources and the benefit of leveraging provenance graph when doing claim verification.

It is important to note that we have not solved the claim provenance problem. We introduce it and explain its importance, provide an initial formulation and an implementation. We argue that, already at this point, our initial formulation and the results it supports provide a significant contribution. We point to a range of future work directions that we discuss at the end of this paper.

2 Provenance of Claims

Given a target claim and a large corpus, we want to infer the *provenance graph* of the claim from the given corpus. This graph will represent previously-made statements with their sources, which, with high probability, ultimately led to the target claim. In this section, we first define the claim provenance graph, and present the problems one must solve to infer it. Note that to distinguish between the query claim and the claims in its previous versions, we use *statements* to refer those previously-made claims by other sources.

2.1 Definition

Let $S_D(q)$ be the set of sources making statements about claim q in corpus D, and $t_s(q)$, an individual statement made by $s \in S_D(q)$.

Definition 1 (Claim Provenance Graph) *Let $G_D(q) = (V, E, L)$ denote the provenance graph of q given D. Here $G_D(q)$ is a labeled directed acyclic graph; $V = \{\langle s, t_s(q)\rangle \cup q | s \in S_D(q)\}$ is a set of nodes. $\forall \langle s, t_s(q)\rangle \in V$, s is the source making statement $t_s(q)$ that is related to the derivation of q. E represents a set of labeled directed edges, and denote $v_i = \langle s_i, t_{s_i}(q)\rangle$, $v_j = \langle s_j, t_{s_j}(q)\rangle$, such that $\forall (v_i, v_j, l) \in E$, $v_i, v_j \in V$, the presence of an edge (v_i, v_j) indicates that $t_{s_i}(q)$* influences the creation *of $t_{s_j}(q)$ via relation $l \in L$. Note that q is the sink node of $G_D(q)$, whose outdegree equals to* 0.

Edge Label Set We use L to categorize how a current statement may be derived by a previous one. Typically, it includes (1) *identical*, when a source quotes a statement from another source; (2) *paraphrased*, when a source describes the same statement with different words; (3) *textually entailed*, when the previous statement can support the current one; (4) *motivated*, when the previous statement potentially influences the appearance of the current one. Practically, we further consider there are two sub-types of *'motivated'*. One is *triggered*, in our running example, the appearance of the claim is very likely due to other related claims, such as *"Facebook should charge users."*, the other one is *contradicted*, when the derived statement has an opposite opinion.

Therefore, the problem we are to solve is given the query claim q and the corpus D, we want to automatically construct its provenance graph $G_D(q)$.

2.2 Problem Overview

To construct the provenance graph, it is obvious that we need to (1) obtain the sources that describe the statements about the claim, i.e., $S_D(q)$; (2) infer the relationship between the sources and the statements, i.e., determine the labeled edges of the provenance graph. To accomplish those two goals, we divide our problem into three subproblems.

Problem 1: Claim Search Detecting the sources requires locating the statements about the claim in the corpus. Therefore, searching for related (and contradictory) sentences to the given claim is a critical aspect. However, it is difficult to locate all statements accurately, since a claim can be spread in many different ways. Moreover, we do not know, when one source proposes a statement, if the statement was a hypothesis supported by the claim, was another claim associated, or it was just simply consistent with the claim. In our running example, the claim of interest can be paraphrased as *"Using Facebook will cost money"*, or can be described as *"Facebook would be implementing a tiered membership system."*, which entails the claim.

Problem 2: Source Extraction *Claim Search* returns a list of articles with sentences related to the given claim, and the next step is to identify who authored those sentences. We assume there are two cases. One is that the *writer of the article* makes a statement about the claim; the other is that *some other source mentioned in the article* describes the claim[2]. For example, one of the articles returned by the *"New York Post"* has a paragraph:

"...First, Facebook should charge users a nominal $5-a-month fee. You can give seniors a discount

[2] We leave for future work a richer model that might also allow for a source to make a claim after being *indirectly influenced* by another uncited source.

so you do not lose them. " In this example, it is clear to the reader that the author of the paragraph is making a statement about the given claim.

Consider another example: *"... In September 2014 the fake news site National Report published a fictitious article positing that Facebook would begin charging users $2.99 per month starting 1 November 2014..."* In this paragraph, the writer is making a statement about how the *"National Report"* asserts the given claim.

In this work, we consider *source extraction* as an information extraction task. Given a *statement* c and the *context* around c, denoted as $T(c)$, from the article returned by *claim search* — we are to determine if there exist sources mentioned in the context which are describing the statement, and if so, to identify the correct sources.

Problem 3: Provenance Graph Construction *Source Extraction* provides us with a multitude of sources mentioned in the articles that are describing the claims. In the previous examples, the sources are *"National Report"* and *"nypost.com"* respectively. However, source extraction only provides a two-layer directed graph, i.e., the writer/url of the article is directed from the sources mentioned in the text. To further complete the provenance graph, we then need to identify the same sources from the sources extracted. For example, the same statement made by *"New York Post"* and *"NY Post"* obtained from the text should link to the same statement made by *"nypost.com"*. After connecting the subgraphs, we then need to determine the relationship between the statements about the claim on the edge, which we view as a classification problem.

3 Inferring the Provenance Graph

To infer the provenance graph for the given claim, we need to solve the three problems outlined in Section 2. Here, we propose a pipelined solution, and elaborate them one by one.

3.1 Searching for the Context

As we described in Section 2, accurately locating the previous statements about the claim is a very challenging problem. Therefore, instead of directly searching for a possible previous statement, we search for related context, where the source are describing a statement related to the claim.

Specifically, we rank sentences in the given corpus, by computing the cosine similarity to the given claim with their ELMo (Peters et al., 2018) representations. Then, we choose sentences that are most similar and fetch their context in a window size w, which means we consider w sentences before and after the returned sentence together as the context, from which we will extract the sources. Note that a returned sentence is denoted as c , and its context is denoted as $T(c)$.

3.2 Extraction as Textual Entailment

Given a sentence c within its context $T(c)$ returned by claim search for q, we need to identify the sources in $T(c)$ that are talking about a statement related to q. This is actually an IE task. Typically, IE is a sequential tagging problem: it needs to learn linguistic patterns from annotated data using syntactic and semantic features, which can express the targeted semantic relations. Most of the solutions in the literature (Surdeanu et al., 2012; Schmitz et al., 2012; Chan and Roth, 2011; Li and Ji, 2014) focus on extracting relationships between two named entities or two nominals. However, in our problem, the relationship of interest is between a nominal/an entity and a statement. The statement can be written either explicitly or implicitly in the given context, and what we only know is that the statement is about q. Therefore, annotation is hard, and existing IE solutions can not be used in this case. Furthermore, the source and the statement may appear across sentences rather than within a single sentence, therefore, coreference resolution may be necessary. For example, *"The website Hoax Slayer said the message dates back to 2012 and has recently resurfaced ... it also noted Facebook has no plans to start charging users for normal access..."* requires a cross-sentence relation extraction (Peng et al., 2017).

Rather than tackling the problem as a sequential tagging problem, we model it as a textual entailment (TE) problem (Dagan et al., 2013). Similar to QA-SRL (He et al., 2015), *TE-IE* task formulation has the advantages of (1) easier annotation (2) being able to capture implicit statements and implicit sources which requires coreference resolution.

TE Modeling We use the dataset (Choi et al., 2005) that contains a set of annotated articles. For each article, it annotates "who" has an opinion on "what". Formally, given a corpus D, for each article $d \in D$, our training data comes in the form of pairs $\{(q_i^d, S_i^d)\}_{i=1}^N$, where we view q_i^d as a claim, and $\forall s \in S_i^d$ is the source of q_i^d mentioned in d.

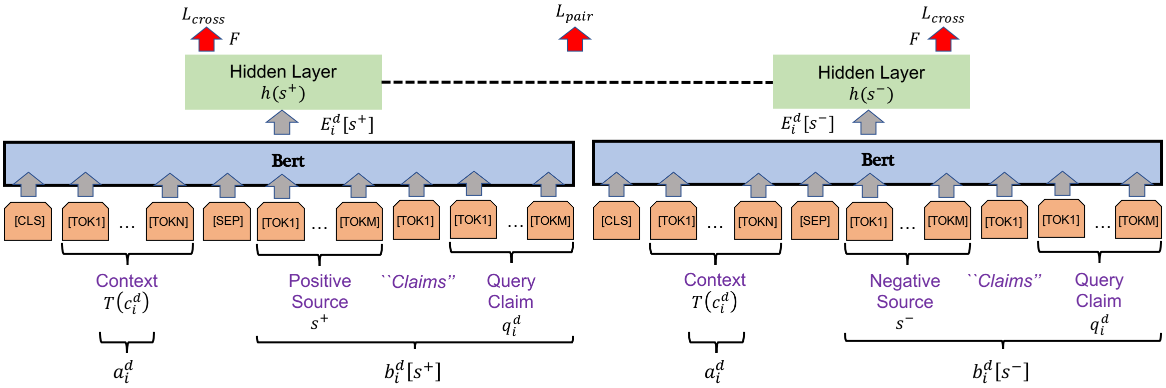

Figure 3: Our TE extraction model: we transform the source extraction problem to be a textual entailment task. In our fine-tuned model based on BERT, our objective function considers (1) binary prediction correctness for sources by cross-entropy loss; (2) difference between positive and negative examples by margin ranking loss.

We search for related sentences and their context for each q_i^d, and denote the returned set of context as $\{T(c_i^d)\}$. Therefore, given q_i^d, a related sentence c_i^d with its $T(c_i^d)$, our problem is to identify s from $T(c_i^d)$, if $s \in S_i^d$.

As we have described, it is hard to directly use existing sequential tagging techniques to solve this problem. Instead, we model it as a TE task. Assume we are given a candidate list of sources, which is a list of spans in text $T(c_i^d)$, denoted as $sc(c_i^d)$ (we will describe how to generate the candidate list later). Then, if we view the context $T(c_i^d)$ as a premise, and generate a sentence following the pattern that the source s "claims"/ "says" the claim q_i^d, where $s \in sc(c_i^d)$ as the hypothesis, we transform the tagging problem to a TE problem. If the premise denoted as a_i^d can entail the hypothesis denoted as $b_i^d[s]$, it means that $s \in S_i^d$, otherwise $s \notin S_i^d$ which means s does not say anything about c_i^d. For each candidate $s \in sc(c_i^d)$, we have a binary classification problem: learn a function F that can decide if a_i^d can entail $b_i^d[s]$.

However, given the query claim q_i^d, a related sentence c_i^d, with its context $T(c_i^d)$ and the candidate list $sc(c_i^d)$, the binary decisions mentioned above are not made independently over the candidate sources. Besides fitting a label that is either entailment or not, the representation of the correct claims should be different from incorrect ones, so that we can have a better chance to learn the discriminative features. We reflect this idea by including a margin ranking loss within our model.

Specifically, we design our model on top of a pre-trained language model for general purpose (BERT) (Devlin et al., 2018), so that we can have a representation of sentences that can capture both semantic and syntactic information. We concatenate a_i^d, $b_i^d[s]$ with separation tokens of BERT to the pre-trained model as shown in Figure 3, and represent the output as $E_i^d[s]$. Then, we add another hidden layer, and feed its result through a final classifier F to do binary prediction, where F is a feed forward network followed by a linear layer

$$\hat{y} = F\big(h(s)\big) \tag{1}$$

where $h(s) = \tanh\left(W_1 E_i^d[s] + b_1\right)$, and $\hat{y} \in \mathrm{R}^C$ represents the predicted scores for each class, and consequently the predicted class is given by $\hat{y} = argmax_i \hat{y}_i$. Here $C = \{0, 1\}$ and W_1, b_1 are learned parameters.

Then, we use cross-entropy loss as a part of our optimization goals.

$$L_{cross} = \frac{1}{N} \sum_{i=1}^{N} \sum_{c} {y_c}^i \log \frac{\exp(\hat{y_c}^i)}{\sum_{c'} \exp(\hat{y_{c'}}^i)} \tag{2}$$

where y_c^i is an indicator that if y^i's label is c.

At the same time, if s_j is a positive example, which means $s_j \in S_i^d$, we randomly sample for s_j a negative example denoted as $s_j^- \in sc(c_i^d)$ and $s_j^- \notin S_i^d$. In this case, we are to maximize the difference between $h(s_j)$ and $h(s_j^-)$, and we reflect it by adding a margin ranking loss as follows:

$$L_{pair}^+ = \frac{1}{N^+} \sum_{j=1}^{N^+} \max\left(0, 1 - (h(s_j) - h(s_j^-))\right) \tag{3}$$

Similarly, we can also sample a positive example s_j^+ for a negative source s_j to get:

$$L_{pair}^- = \frac{1}{N^-} \sum_{j=1}^{N^-} \max\left(0, 1 + (h(s_j) - h(s_j^+))\right) \tag{4}$$

where N^+, N^- are the numbers of positive and negative examples in the annotated data.

For training, we use a loss function L combining both cross-entropy loss for binary prediction and the margin ranking loss to maximize the difference between positive and negative examples to fine-tune the language model. That is:

$$L = \lambda L_{cross} + (1 - \lambda) L_{pair} \quad (5)$$

where $L_{pair} = L^{+}_{pair} + L^{-}_{pair}$, and λ is the parameter to trade off different objectives.

Candidate Generation. The next question is how to generate source candidate list $sc(c^d_i)$ for c^d_i given $T(c^d_i)$. Here, we leverage an off-the-shelf semantic role labeling (SRL) tool (He et al., 2018) that can parse the sentences $T(c^d_i)$ to tell us "who did what to whom" in the appropriate sentences. We then take all "who", i.e., the span of the text with tag ARG0 detected as a candidate source of c^d_i. Even though only the "who" followed by a verb such as "say" or "claim" can be the source theoretically, we included all of them as candidates, and leave the identification made by our TE model.

Note that here we only use SRL to generate candidate sources. Considering (1) the noisy relationship produced by SRL parser, (2) the cross-sentence relationship between the source and the claim, and (3) the fact that a claim can be paraphrased with multiple sentences, we do not determine the sources based on the matching between the claim and the span of text with tag ARG1. We will also show the comparison in our evaluation.

Augmenting Training Data Besides the sources and claims provided by the annotated data, there are still many sentences in the document with a pattern that "who" says or claims "what", which is useful to train the model. To get those examples, we use the off-the-shelf SRL tool to parse all of the sentences in the document, and then compute the similarity between the verb in the parsed sentence and the verb attached with the sources annotated in the text. If the average similarity is higher than a threshold, we include the ARG0 and ARG1 in the parsed sentence as a positive example of the source and the claim. In terms of creating the corresponding negative examples, we randomly replace either ARG0 or ARG1 with other sources or claims. Then we use those created examples to incrementally fine-tune our TE extraction model, which can lead to a better performance.

3.3 Constructing the Graph

After extracting the provenance information, the last process is to construct the provenance graph. The first step thereof, is to link the same sources detected in the text with the same statement. Since the sources can be a url or a mention of an entity, we do wikification (Ratinov et al., 2011; Cheng and Roth, 2013) for the extracted sources. Specifically, to wikify a source mention, we first adapt a redirect-based wikification method (RedW) (Shnayderman et al., 2019), which is efficient and context free. Besides Wikipedia redirects, we also include the value of the attribute *website* as a candidate mention of the entity if it exists, for example *nytimes.com* for *The New York Times*. Then we compute the text similarity between the source mention and the other mentions that have already been linked, and eventually map the source mention to the entity in Wikipedia with a similarity score higher than a threshold. Our similarity score is a linear combination of lexical similarity (Do et al.) between the source mention to (1) candidate mentions produced by RedW and (2) mentions linked. To determine the same statement, we allow an approximate match by computing the cosine similarity with their ELMo representations.

The second step is to decide the relationship between the statements. In this work, we include the relations, i.e., identical, paraphrased, textually entailed and contradicted. Determining if the two statements are identical is straightforward, and we collect parallel sentences (Ganitkevitch et al., 2013; Thorne et al., 2018) to fine-tune classifiers (Devlin et al., 2018) to determine other relations.

4 Application: Claim Verification

We take claim verification as an example to demonstrate the importance of claim provenance graph. Concretely, we elaborate how we can use the graph to improve the estimation of claim veracity.

4.1 Claim Evidence Graph

Claim provenance graph is to help us understand where the claim may come from and how it may be disseminated over time. The nodes of the graph represent the sources with the statements they made, and the edges represent the relations between the statements. However, when doing claim verification, we also care about the direct relation between the statement made by the source and the given claim. Therefore, we derive a *claim evidence graph* from the *claim provenance graph* based on which we do claim verification. Specifically, we keep the nodes and edges in the claim provenance graph, and add another label on each edge with one of *support, contradiction* and *neutral*. The new label on the edge represents the opinion of the source to the given claim, whose generation can be viewed as a regular textual entailment problem.

4.2 Boosting Claim Verification

Given a claim, the most straightforward way to do claim verification is voting by the opinion of different sources. Without the graph, typically we can first search for related articles for the given claim, then collect their opinions and vote. Since each article has its own opinion, we can determine the veracity of the claim by the majority vote of the opinions by those articles. However, an article can include multiple different statements about the same claim with different opinions, and multiple articles can refer to the same statement about the claim from a common source. Therefore, the majority vote by opinions in article level is not good enough, since it suffers from (1) opinions which are too coarse-grained and (2) overcounting the opinions from the same source, which is also known as *collusion* or *dependency of sources* problem in truth finding (Pochampally et al., 2014).

Luckily, with the claim evidence graph, we can collect the opinions *in statement level*, and vote the veracity by sources that are more *independent* with each other. Specifically, given an evidence graph of a claim, we start with the *sink node* and do breadth first search to find all *source nodes* whose indegree are 0, and leverage those sources to vote by their opinions to get an estimation of the claim veracity. To distinguish between sources and *source nodes* of the claim evidence graph, we call sources, a.k.a all nodes of the graph *all-sources*, and independent sources, a.k.a all source nodes of the graph, *prov-sources*. In this case, we can leverage prov-sources that are not dependent with each other to vote. This strategy can also be used to choose sources that will be fed to other source-aware fact finding models (Pasternack and Roth, 2013).

5 Experimental Evaluation

We evaluate (1) the solutions to infer the provenance graph, and (2) the effectiveness of the claim evidence graph on claim verification, which is adapted from the inferred provenance graph. For each goal, we first elaborate the experimental settings, and then describe the results and analysis [3].

5.1 Claim Search and Source Extraction

To evaluate the methods inferring the provenance, we focus on the performance of claim search and source extraction by looking at if the method can extract the sources accurately and exhaustively.

DataSet In this experiment, we use MPQA 2.0[4] (Choi et al., 2005) as the corpus to train and test our models. The dataset consists of 535 documents that have been manually annotated with opinion related information including sources. For example, given a piece of text *"... According to Malan, the dollarization process is irreversible ... "*, *"Pedro Malan"* is annotated that it has an opinion on *"the dollarization process is irreversible"*. Note that a single claim can be annotated with multiple sources including the writer of the text, and each source except the writer is a span of text in the given text. MPQA dataset is originally developed for identifying sources for the given opinion, and the opinion sometimes can be a noun phrase or an entity, while in our problem we are to extract sources for claims. Therefore, we only leave the opinions which are sentences as the query claim, and perform 10-fold cross validation to evaluate the performance of our models and the baselines.

To evaluate the performance, we compute precision, recall and F1 score with overlap match, which means we consider the returned source correct, if it overlaps with at least half of the words of the corresponding annotated source.

Models and Baselines We view source extraction as an IE problem and tackle it by TE models. According to Section 3.2, we evaluate the performance of our model with different versions. The first one is the vanilla TE model, which is fine-tuning BERT to determine if the source makes the claim given the context, i.e., if a_i^d entails $b_i^d[s]$, denoted as TE-V. The second one is the pairwise TE model, which is fine-tuning BERT with two objectives as described in Section 3.2, denoted as TE-P. The third one is the pairwise TE model with the incremental training data provided by an off-the-shelf SRL tool (He et al., 2018), denoted as TE-D.

In terms of the baselines, we compare our models with (1) the sequential tagging solution, which is fine-tuning BERT to predict if the token in the

[3] Our code is available at https://cogcomp.seas.upenn.edu/page/publication_view/901.

[4] http://mpqa.cs.pitt.edu/

text is part of the source, denoted as SEQ; (2) TE model with semantic role labeling, which is to predict if the ARG1 labeled by the SRL is a paraphrase of the query claim, denoted as TE-S.

	PRECISION		RECALL		F1	
	AVG	STD	AVG	STD	AVG	STD
SEQ	0.4906	0.056	0.3373	0.064	0.3998	0.060
TE-S	0.6918	0.053	0.6124	0.048	0.6459	0.022
TE-V	**0.7282**	0.038	0.7103	0.064	0.7165	0.033
TE-P	0.7249	0.024	0.7877	0.065	0.7538	0.038
TE-D	0.7240	0.028	**0.8125**	0.048	**0.7645**	0.024

Table 1: The performance of different models on source extraction. In this table, we report the average precision, recall and F1 score of the 10-fold cross validation on MPQA with their corresponding standard deviations.

Results We report the source extraction results of different methods in Table 1. As shown in the table, modeling source extraction as a TE problem can achieve a better performance than modeling the problem as a sequential tagging task, since both precision and recall of SEQ are lower than the ones of TE-S, which obtained the lowest precision and recall among all of the TE methods. We think the reason is that doing sequential tagging well may need to capture the syntactic relationship in the sentences, while only annotating the source is not enough to make the model understand it.

Comparing TE-S with other TE based models, we can observe that leveraging off-the-shelf SRL to produce candidate sources is helpful. However, determining the sources based on the entailment relationship between ARG1 and the claim will introduce noise, and the quality and the deficiency of the SRL then becomes a bottleneck. Thus, TE-V is better than TE-S. Furthermore, as we argued in Section 3.2, incorporating margin ranking loss into the objective function can help learn the discriminate feature better, which is reflected by the better performance achieved by TE-P compared to the performance of TE-V. We can also observe that incremental training can further improve the performance, as TE-D achieves the best F1 score.

5.2 Claim Verification

In this experiment, we evaluate if the provenance graph can help claim verification methods by its derived claim evidence graph.

DataSet We crawl all 495 fact check questions listed on `www.factcheck.org/askfactcheck/` as the set of query claims, and annotate true or false for each claim based on its conclusion shown on the webpage. Note that we remove the fact check questions without a consolidate conclusion or asking why or what questions about the claim. We also crawl the short answer section, which is a summarized sentence to support the conclusion of the fact check question, listed on the webpage. We use the sentence as the premise, the claim as the hypothesis, and the annotated label as the label, to fine-tune a textual entailment model (Devlin et al., 2018) that can help us determine the label of the edge in the claim evidence graph.

Models and Baselines For each claim, we search it by *google search* [5], and obtain the articles from the top-10 links[6] as the corpus to extract the sources and construct the provenance graph.

Given the provenance graph, we transform it to a claim evidence graph using our fine-tuned model. Then, we implement two methods for claim verification: majority vote, and Simple LCA (Pasternack and Roth, 2013). Note that Simple LCA is iteratively estimating the trustworthiness of the sources and the veracity of the claims. As described in Section 4, we feed the two methods with prov-sources obtained from the claim evidence graph, denoted as Prov-Src. For comparisons, we (1) feed the top-10 links directly as sources into majority vote and Simple LCA respectively; this baseline is denoted as Doc; (2) feed all-sources of the claim evidence graph into majority vote and Simple LCA, denoted as All-Src. Note that All-Src only leverages the nodes of the provenance graph, while Prov-Src leverages both the nodes and the structure of the provenance graph.

To compare the performance, we compute the accuracy of the estimation of the claim veracity.

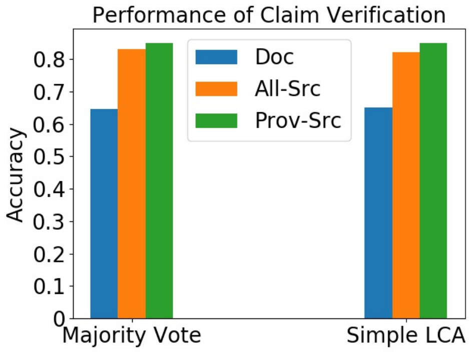

Figure 4: The performance of claim verification with majority vote and Simple LCA. For each method, we evaluate the performance with sources based on articles, all nodes and source nodes of claim evidence graph respectively.

[5] `https://pypi.org/project/google/`

[6] remove the link from `www.factcheck.org`

Results In Figure 4, we report the accuracy of both algorithms, majority vote and Simple LCA, with three groups of sources. Our results show that for both majority vote and Simple LCA, leveraging the claim evidence graph leads to a better performance when compared with using articles as sources. It demonstrates that using articles as sources is too coarse-grained for claim verification, and thus it is very likely to be biased. The evidence graph provides the models with evidence from more sources (All-Src) and sources that are more likely to be independent (Prov-Src), thus improves the performance.

6 Related Work

To the best of our knowledge, our work is the first to formally define and propose a framework to infer the *provenance graph* of given claims made in natural language. One line of the related work includes identifying sources of opinions in opinion analysis (Choi et al., 2005) and quote attribution (Muzny et al., 2017; Pavllo et al., 2018), which is related to one of the components we use to infer the provenance graph. Earlier work performs information extraction via sequential tagging in a given text and collects paired sources and opinions or quotes and speakers. We do not detect all quotes or opinions stated in the text, but rather detect the sources generating statements related to the given claim, whether it is described implicitly or explicitly in the text. Furthermore, we also construct a graph that depicts the history of how a claim has disseminated over time, a task that was not addressed in earlier work.

Another line of related work includes fact-checking (Thorne et al., 2018; Thorne and Vlachos, 2018; Zhang et al., 2019) and claim verification (Popat et al., 2017, 2018). However, those works focus only on capturing discriminative linguistic features of misinformation, while we argue that determining the provenance of claims is essential for addressing the root of the problem, understanding claims and sources.

7 Conclusion and Future Work

We introduce a formal definition and a computational framework for the provenance of a natural language claim given a corpus. We argue that this notion of provenance is essential if we are to understand how claims evolve over time, and what sources contributed to earlier versions of the claims. We provide initial results exhibiting that our framework can be used successfully to infer the provenance graph and, that it can be applied to boost the performance of claim verification.

The framework introduces a range of important questions both from the inference and the application perspectives. For example, inferring the current version of the provenance graph depends on the ability to identify authors. This could be difficult when the authors are not mentioned in the text, which might require a deeper understanding of sources' writing style and positions.

From the application perspective, it is clear that the graph contains more information than we have exploited so far. For example, the edge labels, indicating the evolution operators of a claim should also be useful. In particular, this will support a more informed study of influence of specific sources and of trustworthiness, and possibly other aspects of information spread.

Acknowledgement

The authors would like to thank Nitish Gupta and the anonymous reviewers for insightful comments and suggestions. This work was supported in part by a Focused Award from Google and by IARPA Contract No. 2019-19051600006 under the BETTER Program. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

References

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer.

Khalid Belhajjame, Reza BFar, James Cheney, Sam Coppens, Stephen Cresswell, Yolanda Gil, Paul Groth, Graham Klyne, Timothy Lebo, Jim McCusker, et al. 2013. Prov-dm: The prov data model. *W3C Recommendation. http://www. w3. org/TR/prov-dm.*

Yee Seng Chan and Dan Roth. 2011. Exploiting syntactico-semantic structures for relation extraction. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, Portland, Oregon.

James Cheney, Laura Chiticariu, Wang-Chiew Tan, et al. 2009. Provenance in databases: Why, how, and where. *Foundations and Trends® in Databases*, 1(4):379–474.

Xiao Cheng and Dan Roth. 2013. Relational inference for wikification. In *Proceedings of the 2013 Conference on EMNLP*, pages 1787–1796.

Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. 2005. Identifying sources of opinions with conditional random fields and extraction patterns. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 355–362. Association for Computational Linguistics.

Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzoto. 2013. Recognizing textual entailment: Models and applications.

Susan B Davidson and Juliana Freire. 2008. Provenance and scientific workflows: challenges and opportunities. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1345–1350. ACM.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Quang Do, Dan Roth, Mark Sammons, Yuancheng Tu, and V Vydiswaran. Robust, light-weight approaches to compute lexical similarity.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764.

Luheng He, Kenton Lee, Omer Levy, and Luke Zettlemoyer. 2018. Jointly predicting predicates and arguments in neural semantic role labeling. *arXiv preprint arXiv:1805.04787*.

Luheng He, Mike Lewis, and Luke Zettlemoyer. 2015. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 643–653.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2009. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pages 94–99. Association for Computational Linguistics.

Qi Li and Heng Ji. 2014. Incremental joint extraction of entity mentions and relations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 402–412, Baltimore, Maryland. Association for Computational Linguistics.

Grace Muzny, Michael Fang, Angel Chang, and Dan Jurafsky. 2017. A two-stage sieve approach for quote attribution. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 460–470.

Levent Orman. 1984. Fighting information pollution with decision support systems. *Journal of management information systems*, 1(2):64–71.

Jeff Pasternack and Dan Roth. 2013. Latent credibility analysis. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1009–1020. ACM.

Dario Pavllo, Tiziano Piccardi, and Robert West. 2018. Quootstrap: Scalable unsupervised extraction of quotation-speaker pairs from large news corpora via bootstrapping. In *Twelfth International AAAI Conference on Web and Social Media*.

Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. 2017. Cross-sentence n-ary relation extraction with graph lstms. *TACL*, 5:101–115.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Ravali Pochampally, Anish Das Sarma, Xin Luna Dong, Alexandra Meliou, and Divesh Srivastava. 2014. Fusing data with correlations. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 433–444. ACM.

Kashyap Popat, Subhabrata Mukherjee, Jannik Strötgen, and Gerhard Weikum. 2017. Where the truth lies: Explaining the credibility of emerging claims on the web and social media. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 1003–1012. International World Wide Web Conferences Steering Committee.

Kashyap Popat, Subhabrata Mukherjee, Jannik Strötgen, and Gerhard Weikum. 2018. Credeye: A credibility lens for analyzing and explaining misinformation. In *Companion of the The Web Conference 2018 on The Web Conference 2018*, pages 155–158. International World Wide Web Conferences Steering Committee.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and global algorithms for disambiguation to wikipedia. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Michael Schmitz, Robert Bart, Stephen Soderland, Oren Etzioni, et al. 2012. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534. Association for Computational Linguistics.

Ilya Shnayderman, Liat Ein-Dor, Yosi Mass, Alon Halfon, Benjamin Sznajder, Artem Spector, Yoav Katz, Dafna Sheinwald, Ranit Aharonov, and Noam Slonim. 2019. Fast end-to-end wikification. *arXiv preprint arXiv:1908.06785*.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 455–465. Association for Computational Linguistics.

James Thorne and Andreas Vlachos. 2018. Automated fact checking: Task formulations, methods and future directions. *arXiv preprint arXiv:1806.07687*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*.

Yi Zhang, Zachary Ives, and Dan Roth. 2019. Evidence-based trustworthiness. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 413–423.