A Novel Unsupervised Approach for Precise Temporal Slot Filling from Incomplete and Noisy Temporal Contexts

Xueying Wang
University of Notre Dame
Notre Dame, IN
xwang41@nd.edu

Haiqiao Zhang
University of Notre Dame
Notre Dame, IN
hzhang5@nd.edu

Qi Li
University of Notre Dame
Notre Dame, IN
qli8@nd.edu

Yiyu Shi
University of Notre Dame
Notre Dame, IN
yshi4@nd.edu

Meng Jiang
University of Notre Dame
Notre Dame, IN
mjiang2@nd.edu

ABSTRACT

The task of temporal slot filling (TSF) is to extract the values (or called facts) of specific attributes for a given entity from text data and find the time points when the values were valid. It is challenging to find precise time points with incomplete and noisy temporal contexts in the text. In this work, we propose an unsupervised approach of two modules that mutually enhance each other: one is a reliability estimator on fact extractors conditionally to the temporal contexts; the other is a fact trustworthiness estimator based on the extractor's reliability. The iterative learning process reduces the noise of the extractions. Experiments demonstrate that our approach, with the novel design, can accurately and efficiently extract precise temporal facts from newspaper corpora.

KEYWORDS

Temporal slot filling, Incompleteness, Information extraction

ACM Reference Format:
Xueying Wang, Haiqiao Zhang, Qi Li, Yiyu Shi, and Meng Jiang. 2019. A Novel Unsupervised Approach for Precise Temporal Slot Filling from Incomplete and Noisy Temporal Contexts. In *Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3308558.3313435

1 INTRODUCTION

Can AI read ten million news articles in two minutes and then fill in the three slots below:

(“vicente_fox”, per:is_president_of, “□”, [□ , □]) ?
(entity, attribute, value, [beginTime, endTime])

The first slot □ is the value of a specific attribute (e.g., country's president) for an entity (e.g., the person “vicente_fox”). Here the value should be a country's name. The second and third slots are the *beginning* and *ending* time points of the attribute value being valid. We name this task “precise temporal slot filling” (PTSF). PTSF techniques will facilitate the automation of knowledge base construction and question answering.

Existing TSF task presents the time as a single slot and expects to extract time expressions directly from sentences to fill the slot. For example, given a sentence in the newspaper:

“...Vicente Fox served as the President of Mexico from 2000 to 2006,...”

TSF looks for “mexico” to fill the value slot and the time expression “from 2000 to 2006” to fill the time slot. Decomposing this time expression into two time values is not hard in this case, in order to solve PTSF with the TSF result. However, we observe from a data set of ten million news articles that among all the sentences that contain time expressions, fewer than 0.1% contain at least two time points like “from 2000 to 2006:” most of the expressions are shorter like “in 2002,” which indicates a time point of the fact but cannot precisely complete the *beginTime* and *endTime* slots.

We examined two information extraction methodologies: open-domain IE (OpenIE) and pattern-based IE (PatternIE). We find that the OpenIE approaches [1, 2, 7, 10, 25] generate *low precision on the value slots* and *lower-than-0.1 recall on the time slots*. First, OpenIE relies on predicates to extract relations between a subject and an object, however, the attribute may not be a predicate. Given the above sentence example, OpenIE generates (“vicente_fox”, “serve_as”, “president_of_mexico”) instead of finding “mexico” as the value of attribute “is_president_of”. Second, OpenIE could not find the precise time points if there was very few such long time expressions as “from 2000 to 2006” in the sentences that contain the fact. On the other hand, PatternIE methods [16, 17, 22] use textual patterns to find the attribute values, but very few methods associate the values with time information. A pattern-based temporal anchoring method [23] assumes that both types of temporal contexts, time expression and document post time, are accurate to be the time of the attribute value being valid. However, this assumption is too strong to find precise temporal slots from the noisy contexts. For example, given the sentences:

“*In 1979, the [former U.S. President Jimmy Carter] deregulated the American beer industry...*” (posted on August 5, 2010), “*[Donald Trump, now President of United States,] published his first book in 1987...*” (posted on June 3, 2017),

the post time in the first sentence “2010” is not in the presidential term of President Carter (1977−1981), and the time expression in

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313435

Table 1: Pattern's reliability scores for country's presidency.

Textual Pattern p	$r^{(post)}(p)$	$r^{(tag)}(p)$
\$Country president \$Person	0.91	0.53
\$Country 's president \$Person	0.86	0.84
president \$Person of \$Country	0.84	0.70
former \$Country president \$Person	-1	0.85
\$Country 's former president \$Person	-0.81	0.83
\$Person , now president of \$Country ,	0.95	-0.89
current \$Country president \$Person	0.93	-0.59
\$Country 's current president , \$Person ,	0.81	0
new \$Country president , \$Person ,	0.57	-0.25
\$Country 's newly elected president , \$Person ,	0.20	-0.64
current \$Country prime minister \$Person	-1	-1
\$Country premier \$Person	-0.82	-0.86
\$Country foreign minister \$Person	-1	-1
\$Country golfer \$Person	-1	-1

the second sentence "1987" is not in the term of President Trump, either. We observe that different textual patterns may have different *reliability* on extracting the time points, conditionally to the temporal contexts (including time expression and post time):

- Pattern [former \$Country president \$Person] is reliable for the time expression "1979", not the post time "2010."
- Pattern [\$Person, now president of \$Country,] is reliable for the post time "2018", not the time expression "1987."

In this work, we propose a novel unsupervised approach based on PatternIE to infer the precise temporal slots from *incomplete* and *noisy* temporal contexts. For an attribute, PatternIE discovers a large set of textual patterns and uses them to extract EVT-tuples from text data: $p \rightarrow \{(e, v, t)\}$, where p is a textual pattern, e denotes the entity, v denotes the attribute value, and t is a time point from either time expression or post time. The goal is to infer temporal fact tuples $\{(e, v, [t_b, t_e])\}$, where t_b/t_e is beginTime/endTime, from millions of EVT-tuples (e.g., 5.3M for country's president). Our idea is to jointly estimate the pattern's reliability and the tuple's trustworthiness: if a set of tuples (including the time point) are more trustworthy, the pattern that extracted these tuples is more likely to be reliable; and, if a pattern is more reliable, its extractions are more likely to be true. Effectiveness of this iterative truth-discovery methodology has been demonstrated for finding the book's true author names from information on book-selling websites [18, 34].

Unfortunately, the truth-discovery algorithms cannot be directly applied to the proposed problem because they often hold the single-truth assumption (e.g., a book has only one true author list) to find conflicting values and thus capture unreliable sources [9, 19, 31, 34]. However, in the task of fact extraction, an entity may have multiple values at multiple time points. Then how can we know that the tuple ("u.s.", "jimmy_carter", 2010) is wrong (w.r.t. country's president) when United States had multiple presidents? Our common sense can help: if we had a trustworthy tuple ("u.s.", "barack_obama", 2010) and we knew that one country is likely to have *only one* president in one year, we would have a chance to find the conflicts.

Based on the above observation, we propose to use *the World's invariants* (i.e., quantitative common senses) including time-irrelevant and time-relevant constraints to find the conflicts and the truth. In the following section, we show that with just one seed pattern, we can generate and validate (1) time-irrelevant hypotheses:

- H1: one country is likely to have *multiple* presidents in the history;
- H2: one president is likely to serve *only one* country;

and (2) time-relevant hypothesis:

- H3: in one year, one country is likely to have *only one* president.

H2 and H3 can be used as constraints to find conflicts between tuples while H1 cannot. For example, suppose the fact ("barack_obama", "u.s.", [2009, 2017]) is true. So ("barack_obama", "china", 2014), which was extracted from the following sentence:

"*[President Barack Obama visited China] and attended the APEC summit...*" (posted on November 11, 2014),

is false because of H2 and ("jimmy_carter", "u.s.", 2010) is false because of H3. Then we know that (1) the pattern [president \$Person visited \$Country] is unreliable to extract a fact of country's president and (2) [former \$Country president \$Person] is unreliable to claim the fact's time point as the post time.

Based on the above ideas, estimating pattern reliability and finding conflicts with the World's invariants, we propose a Truth Finding-driven framework using the World's INvariants, called TFWIN, to extract precise temporal facts from text corpus. First, it uses PatternIE to structure the corpus into textual patterns and (e, v, t)-tuples. Second, it uses hypothesis testing to derive time-irrelevant and time-relevant constraints. Third, it iteratively evaluates pattern's reliability (upon two different temporal context types), estimates time point's trustworthiness, updates beginTime/endTime slots, and detects false tuples using the constraints. Two important properties of this algorithms are: (1) the time complexity is *quasi-linear* to the corpus size; and (2) it requires *NO* expensive annotations or heavy parameter tuning.

Table 1 presents the reliability scores given to a few pattern examples by our algorithm, where $r^{(post)}(p)$ and $r^{(tag)}(p)$ denote the reliability of pattern p on claiming the post time and temporal tag (i.e., time expression) as a valid time point, respectively. The score ranges from -1 to 1: -1 means that the extractions by the pattern are very likely to be false w.r.t. the specific attribute; 1 means the extractions are likely to be true; 0 means the extractions have very little correlation with the attribute. We observed that a textual pattern could be positive or negative on both, or positive on one and negative on the other, or vice versa. The pattern reliability match our intuition and effectively estimates the truth of value, beginTime, and endTime slots.

We summarize our main contributions as follows.

- We study the problem of precise temporal slot filling and point out the limitations of existing OpenIE and PatternIE.
- We propose the ideas of estimating pattern reliability and detecting conflicts with the World's invariants to handle incompleteness and noise of temporal contexts in text data.
- We propose a novel unsupervised framework (TFWIN) to find precise temporal facts from massive general corpora with no requirement of human annotations.
- Experiments demonstrated the effectiveness. AUC and F1 were improved by 25+% over the state-of-the-art.

The rest of this paper is organized as follows. Section 2 provides data preprocessing and problem definition. Section 3 presents the overview and details of the proposed framework. Experimental results can be found in Section 4. Section 5 surveys the literature of related works. Section 6 concludes the paper.

2 PRELIMINARIES

We first introduce the techniques to turn the text data into "pattern-to-(entity, value, time)-tuples." Then we define the problem.

2.1 Structuring Text into "Pattern-Tuple"

Pattern-based methods are the most popular for information extraction in an unsupervised way from massive text corpora. The idea is that the textual patterns become frequent when entity names in the patterns are replaced by symbols $E (entity) or $V (value) [12, 13, 33] or their types like $Person or $Country [16, 22]. The type-level textual patterns can generate a large set of concrete (entity, value)-tuples from sentences. Then we will introduce how to have the "pattern-(entity, value, *time*) tuple" structures in detail.

2.1.1 *Entity recognition and typing.*

We use the entity recognition and typing techniques to jointly recognize entity names and their *fine-grained* types simultaneously. For example, country names such as "United States", "Mexico", and "Burkina Faso" are recognized and typed as "$Location.Country" (simplified as "$Country").

2.1.2 *Textual pattern mining.*

We use the textual pattern mining method MetaPAD [16] to discover "meta patterns" as information extractors. The meta pattern is defined in [16] as below.

Definition 2.1 (Meta Pattern). A meta pattern refers to a frequent textual pattern of entity types (e.g., $Country, $Person), words, and possibly punctuation marks, which serves as an integral semantic unit in certain context.

2.1.3 *EVT-tuples and precise temporal fact tuples.*

For a specific attribute (e.g., country's president), the meta patterns of the corresponding entity type (e.g., $Country) and value type (e.g., $Person) can generate a set of (entity, value)-pairs. To discover temporal facts, we attach two types of time signals to the tuples: One is the "post time" which is the time of the document being posted, and the other is "time expression" or called "temporal tag" which is the nearest temporal tags (within a 20-word window) to the entity mention. We use a popular tagging tool [29] to extract the temporal tags. Now we can define the EVT-tuples as below.

Definition 2.2 (EVT-tuple). For a specific attribute a that refines the entity type as $c_e(a)$ and the value type as $c_v(a)$, an EVT-tuple refers to an (e, v, t)-tuple, where the type of e is $c_e(a)$, the type of v is $c_v(a)$, (e, v)-pair is extracted by a pattern p, and t is the timestamp attached to the pair.

Given the text data, we use the above techniques to preprocess the data and find millions of textual patterns, EVT-tuples, and their associations. We look for precise temporal fact tuples:

Definition 2.3 (Precise temporal fact tuple). For a specific attribute a, a temporal fact tuple refers to an $(e, v, [t_b, t_e])$-tuple, where for any time $t \in [t_b, t_e]$, v is a valid attribute value of e's attribute a. The beginTime t_b and endTime t_e must be precisely specified as time values (e.g., a concrete year, month, or date) instead of text-based time expressions (e.g., "from ... to ...", "since ...").

Table 2: Symbols and their descriptions.

Symbol	Description
$\mathcal{D}^{(*)}$ $* \in \{``post", ``tag"\}$	"Pattern-tuple" extraction list in which time signal comes from $*$
$\mathcal{P}$	The set of textual patterns
(e, v, t)	EVT-tuple (entity, value, time)
$c^{(*)}(p, (e, v, t))$	The count of times p extracts (e, v, t)
$(e, v, [t_b, t_e])$	Precise temporal fact tuple
$\mathcal{F}$	The list of true temporal facts
$r^{(*)}(p)$	The reliability score of pattern p
$w((e, v, t))$	The trustworthiness score of EVT-tuple
$\mathcal{H}$	A set of hypotheses to define conflicts
$f((e, v, t), \mathcal{F}, \mathcal{H}) \in$ $\{``T", ``F", ``U"\}$	Flag of checking (e, v, t) with $\mathcal{F}$ on $\mathcal{H}$: True, False, Undetermined
$\mathcal{P}^{(*)}_{(e,v,t)}$	Pattern set that extracts (e, v, t) from $\mathcal{D}^{(*)}$
$\mathcal{D}^{(*)}_{p}$	EVT-tuple set extracted by p from $\mathcal{D}^{(*)}$

2.2 Problem Definition

Table 2 describes the symbols we use in this paper. With the above concepts defined in Section 2.1, we define the problem of precise temporal fact extraction on the "pattern-tuple" structured data.

Problem (Precise Temporal Fact Extraction)**. Given** two "pattern-tuple" structured extraction lists $\mathcal{D}^{(post)}$ and $\mathcal{D}^{(tag)}$, that can be represented as $\mathcal{D}^{(*)} = \left\{ \left(p, (e, v, t), c \right) \right\}$ in which the time t comes from two kinds of signals, i.e., "post" for the time of document being posted and "tag" for the nearest temporal tag, and where c can be concretely written, for example, $c^{(tag)}(p, (e, v, t))$ is the count of times that textual pattern p extracts the (e, v, t)-tuple along the temporal tag t, **(1) estimate** the reliability of each textual pattern that is described as a function $r^{(*)}(p) : p \in \mathcal{P} \rightarrow [-1, 1]$, where $\mathcal{P}$ is the set of textual patterns, **(2) infer** the trustworthiness of each EVT-tuple that is described as a function: $w(e, v, t) : (e, v, t) \rightarrow [-1, 1]$, and **(3) find** the list of true temporal fact tuple $\mathcal{F} = \left\{ \left(e, v, [t_b, t_e] \right) \right\}$.

3 THE PROPOSED APPROACH

3.1 Overview

Figure 1 presents the illustration of the proposed TFWIN framework using the attribute *country's president* as an example.

The unsupervised approach is initialized by one seed pattern (assuming that it has high reliability) and iteratively does two-step learning: step (a) is to estimate the tuple trustworthiness based on pattern reliability and to update the two precise time slots of temporal facts with trustworthy time points; step (b) is to find true EVT-tuples (if satisfies the precise temporal facts) and false EVT-tuples (if conflicts with the World's invariants), and then to estimate the pattern reliability based on tuple trustworthiness. It will converge when all the EVT-tuples were separated into two parts: one can be located into the precise temporal facts, and the other violate at least one precise temporal fact holding the constraints.

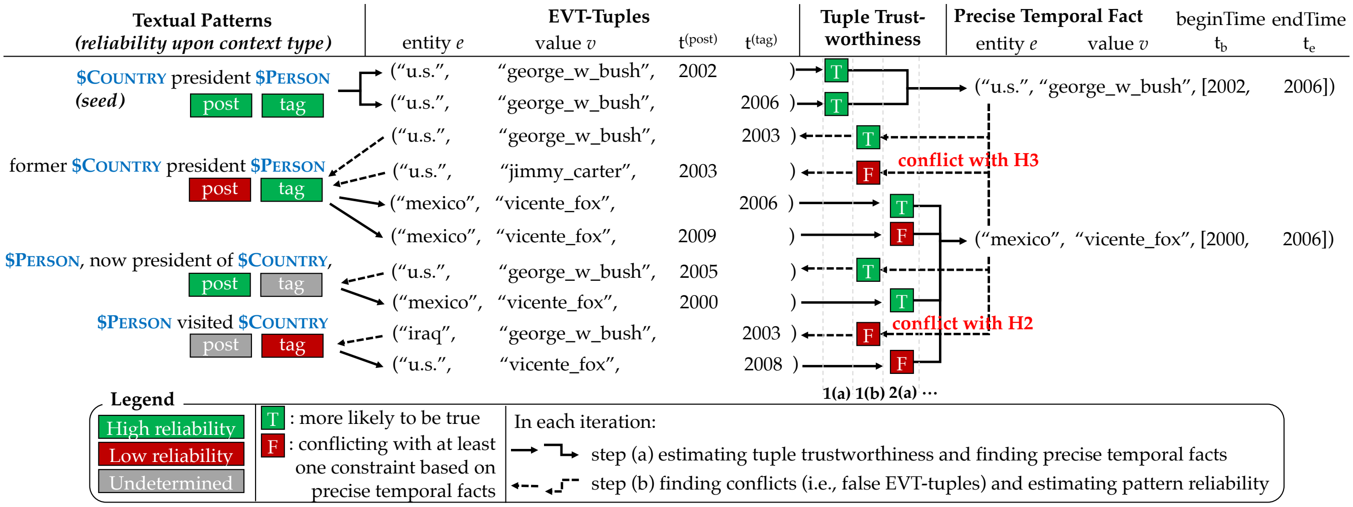
Textual Patterns
(reliability upon context type)
EVT-Tuples
entity e
value v
t(post)
t(tag)
Tuple Trust-worthiness
Precise Temporal Fact
entity e
value v
beginTime tb
endTime te
$COUNTRY president $PERSON
(seed)
post
tag
former $COUNTRY president $PERSON
post
tag
$PERSON, now president of $COUNTRY,
post
tag
$PERSON visited $COUNTRY
post
tag
("u.s.", "george_w_bush", 2002)
("u.s.", "george_w_bush", 2006)
("u.s.", "george_w_bush", 2003)
("u.s.", "jimmy_carter", 2003)
("mexico", "vicente_fox", 2006)
("mexico", "vicente_fox", 2009)
("u.s.", "george_w_bush", 2005)
("mexico", "vicente_fox", 2000)
("iraq", "george_w_bush", 2003)
("u.s.", "vicente_fox", 2008)
T
F
conflict with H3
conflict with H2
("u.s.", "george_w_bush", [2002, 2006])
("mexico", "vicente_fox", [2000, 2006])
1(a) 1(b) 2(a) ···
Legend
High reliability
Low reliability
Undetermined
T : more likely to be true
F : conflicting with at least one constraint based on precise temporal facts
In each iteration:
step (a) estimating tuple trustworthiness and finding precise temporal facts
step (b) finding conflicts (i.e., false EVT-tuples) and estimating pattern reliability

Figure 1: Our proposed TFWIN framework iteratively estimates reliability of textual patterns as information sources, infers trustworthiness of temporal facts, and resolves conflicts defined by the World's invariants (e.g., H2 and H3).

3.2 The Iterative Learning Algorithm

Generally, the algorithm is an iterative method. It starts with very light supervisory information, that is, *one* highly reliable pattern. Usually we use the pattern [$TYPEOF$e$ *a* $TYPEOF$v$] as the seed pattern for attribute *a*. This pattern is not necessarily the most frequent one though most of the time it is. Since only one pattern is needed as the seed pattern, it will not take a lot of effort to find one. For example, [$COUNTRY president $PERSON] is a reliable seed pattern for the attribute *country's president*.

We use the frequent EVT-tuples extracted by the seed pattern to generate seed temporal fact tuples till *a conflict occurs*. Then we generate negative labels (i.e., false EVT-tuples) based on the constraints. With the positive and negative tuples, we iteratively estimate the reliability of textual patterns and infer the trustworthiness of undetermined tuples. We use tuples of good trustworthy scores, from the highest to low non-negatives, to update the positive labels (i.e., temporal fact tuples) till *a conflict occurs* or *the tuple's support is below a threshold* α and re-generate the negative labels. When it comes to convergence, the algorithm returns the final set of temporal facts.

3.2.1 Conflicts and negative label generation. Here we define a function of checking if an (e, v, t)-tuple is conflicted with existing true temporal facts $\mathcal{F}$ based on the set of hypotheses $\mathcal{H}$:

$$f((e,v,t),\mathcal{F},\mathcal{H}) = \begin{cases} \text{“}T\text{”}, & \text{if a fact tuple in } \mathcal{F} \text{ includes } (e,v,t); \\ \text{“}F\text{”}, & \text{if } (e,v,t) \text{ conflicts with } \mathcal{F} \text{ on} \\ & \text{any hypothesis } H \in \mathcal{H}; \\ \text{“}U\text{”}, & \text{else;} \end{cases}$$

where "T" denotes for True, "F" denotes for False, and "U" denotes for Undetermined. More formally, $f((e,v,t),\mathcal{F},\mathcal{H}) =$ "T" for $\forall \mathcal{H}$, if the statement

$$\exists (e, v, [t_b, t_e]) \in \mathcal{F}, t_b \le t \le t_e \tag{1}$$

is true. $f((e,v,t),\mathcal{F},\{H_{1e\text{-}to\text{-}1v}\}) =$ "F", if the statement

$$\exists (e, v', [t_b, t_e]) \in \mathcal{F}, v' \ne v \tag{2}$$

is true. $f((e,v,t),\mathcal{F},\{H_{1v\text{-}to\text{-}1e}\}) =$ "F", if the statement

$$\exists (e', v, [t_b, t_e]) \in \mathcal{F}, e' \ne e \tag{3}$$

is true. $f((e,v,t),\mathcal{F},\{H_{(1t)1e\text{-}to\text{-}1v}\}) =$ "F", if the statement

$$\exists (e, v', [t_b, t_e]) \in \mathcal{F}, t_b \le t \le t_e \textit{ and } v' \ne v \tag{4}$$

is true. $f((e,v,t),\mathcal{F},\{H_{(1t)1v\text{-}to\text{-}1e}\}) =$ "F", if the statement

$$\exists (e', v, [t_b, t_e]) \in \mathcal{F}, t_b \le t \le t_e \textit{ and } e' \ne e \tag{5}$$

is true. At the i-th iteration, given $\mathcal{F}^{(i)}$ and $\mathcal{H}$, we assign *polarized* trustworthiness score to tuples as below:

$$w((e,v,t)) = \begin{cases} 1, & \text{if } f((e,v,t),\mathcal{F},\mathcal{H}) = \text{“}T\text{”}; \\ 0, & \text{if } f((e,v,t),\mathcal{F},\mathcal{H}) = \text{“}U\text{”}; \\ -1, & \text{if } f((e,v,t),\mathcal{F},\mathcal{H}) = \text{“}F\text{”}. \end{cases} \tag{6}$$

Then we have positive/negative labels to estimate pattern reliability.

3.2.2 Pattern reliability estimation. The reliability score of pattern p can be estimated from the trustworthiness of its EVT-tuples:

$$r^{(*)}(p) = \frac{\sum_{(e,v,t)\in\mathcal{D}_p^{(*)}} c^{(*)}(p,(e,v,t))\, w((e,v,t))}{\sum_{(e,v,t)\in\mathcal{D}_p^{(*)}} c^{(*)}(p,(e,v,t))\, |w((e,v,t))|} \in [-1, 1], \tag{7}$$

where $*$ is for temporal context type (or called time signal source, either "post" for post time or "tag" for temporal tag) and $c^{(*)}(p,(e,v,t))$ is the count of times that the (e,v,t)-tuples are extracted by p. The idea is that a pattern is more (un-)reliable, if its EVT-tuples are more (un-)trustworthy. Note that the pattern reliabilities are separately modeled based on different time signal sources. In the experiments, we will compare the performances of our algorithm's settings: (1) between source-aware and source-unaware modeling and (2) between considering and not considering counts.

3.2.3 Tuple trustworthiness inference. When the pattern reliabilities are estimated, we evaluate the trustworthiness of the *undetermined* tuples as below:

$$w((e,v,t)) = \frac{\sum_{*} \sum_{p \in \mathcal{P}^{(*)}_{(e,v,t)}} c^{(*)}(p,(e,v,t)) r^{(*)}(p)}{\sum_{*} \sum_{p \in \mathcal{P}^{(*)}_{(e,v,t)}} c^{(*)}(p,(e,v,t))} \in [-1,1], \tag{8}$$

where we integrate pattern reliability's contributions from both time signal sources. If an EVT-tuple is often extracted by (un-)reliable patterns, it is more (un-)trustworthy. In the experiments, we will investigate the effectiveness of considering counts c.

4 EXPERIMENTS

Here we conduct experiments to answer the following quesitons:

Q1. (Effectiveness) Is TFWIN effective in mining temporal facts from text data? Do both time sources (post time and temporal tag) help? Does the truth finding module improve the mining process? Does the World's Invariants derived from significance tests help?

Q2. (Interpretability) Do textual patterns have different reliabilities? For one pattern, are its reliabilities the same or different upon different time sources?

4.1 Experimental Settings

Text data description. The dataset has 9,876,086 news articles (4 billion words) published from 1994–2010.

Structured data size and ground truth. We focus on finding precise temporal facts on *country's president.* For *country's president*, we have 53,298 textual patterns of \$Country and \$Person, 116,631 unique EVT-tuples, and as many as **5,335,344** tuple extractions, where the timestamps are refined to the *year* level. We collected the ground truth from Google and Wikipedia, which contains 365 $(e, v, [t_s, t_e])$-tuples of 130 countries and can then be split into 3,175 (e, v, t)-tuples.

Evaluation methods. We evaluate the performance of our method and all baselines on mining the 3,175 true (e, v, t)-tuples using standard Information Retrieval metrics: *precision*, *recall*, *F1 score*, and *AUC* (Area Under the Curve). Precision is the the fraction of true (e, v, t)-tuples among the (e, v, t)-tuples split from $(e, v, [t_s, t_e])$-tuples. Recall is the fraction of true (e, v, t)-tuples among the (e, v, t)-tuples split from the ground truth. F1 score is the harmonic mean of precision and recall. For all of the metrics, the higher scores indicate that the method has better performance.

Baseline methods and parameter settings. There was no existing work that introduces the idea of truth discovery methodology to the problem of temporal fact extraction from unstructured data. Our work proposes to structure the data with textual patterns and use the World's invariants for truth finding. We compare our method with existing iterative-based (or called propagation-based) truth finding algorithms as well as its multiple variants when given the structures. As shown in Table 3, the methods are as follows.

(1) MajVote-t [11]: it uses the weighted majority voting strategy and returns the most frequent (e, v, t)-tuples;
(2) MajVote-$[t_s, t_e]$: it composites frequent (e, v, t)-tuples into $(e, v, [t_s, t_e])$-tuples, where $t_s = \min t$ and $t_e = \max t$;
(3) TruthFinder-$H_{1e-to-1v}$ [34]: we modify TruthFinder by taking its hypothesis as $H_{1e-to-1v}$, because it assumes "one book only has one author list". Then the patterns and facts of attribute are regarded as the *websites* and *books' author list*, respectively.
(4) TruePIE [17]: it assumes hypothesis $H_{1v-to-1e}$ without considering time-relevant invariants which are very important for temporal fact extraction;
(5) TFWIN and its variants: all TFWIN methods use the valid time-irrelevant hypothesis $H_{1v-to-1e}$ and the valid time-relevant hypothesis $H_{(1t)1e-to-1v}$ that were derived from hypothesis tests. The four variants discuss whether the count of pattern-tuple extraction $c(p, (e, v, t))$ matters in evaluating the pattern reliability $r(p)$ (Eq.(7)) and tuple trustworthiness $w((e, v, t))$ (Eq.(8)).

4.2 Experimental Results

In this section, we present results on both effectiveness and interpretability of the proposed approach.

4.2.1 Effectiveness. Table 3 presents the AUC and F1 of all the methods on finding *country's president* in the "pattern-tuple" structures from text data.

Overall performance. The best baseline method MajVote-$[t_s, t_e]$ gives an AUC of 0.4958 when integrating post time and temporal tag for tuple majority voting, and gives an F1 of 0.6049 on post-time-only tuples. Our TFWIN, which conducts truth finding with two valid hypotheses and source-aware pattern reliability modeling, generates an AUC of 0.6146 (**+24.0%** over the baseline) and an F1 of 0.7572 (**+25.2%**), when $\alpha = 10$. The best of TFWIN ($\alpha = 7$) shows an AUC of 0.6216 (**+25.4%**) and an F1 of 0.7654 (**+26.5%**). Details of different experimental settings will be discussed as follows:

TFWIN vs majority voting. Table 3 shows that MajVote-$[t_s, t_e]$ performs significantly better than MajVote-t (+48.6% AUC; +37.3% F1) because the attribute *president* has $[t_s, t_e]$-period property.

TFWIN vs TruthFinder and TFWIN: comparing with existing iterative-based truth finding methods. TruthFinder [34] assumes one object has only one true fact; however, it doesn't make sense to assume $H_{1e-to-1v}$ (one country has only one president). It is no longer the book's authorlist case. So the AUC and F1 are very poor (< 0.01). We derive two World's invariants, a time-irrelevant constraint $H_{1v-to-1e}$ and a time-relevant constraint $H_{(1t)1e-to-1v}$. TruePIE [17] can be applied to time-irrelevant cases that follow $H_{1v-to-1e}$. The AUC is 0.06 and the F1 is 0.14. The performance is better than TruthFinder, because TruthFinder does not use this correct hypothesis. Our TFWIN uses both valid invariants as well as the time-relevant invariants, and therefore, it outperforms TruthFinder and TruePIE: the AUC becomes 0.62 (10.6× higher) and the F1 is 0.76 (5.4× higher). These demonstrate the power of using the valid World's invariants in true fact discovery.

4.2.2 Interpretability. Table 1 presents the reliability scores of some textual patterns upon the *post time* and *temporal tag* sources. Table 4 shows a subset of temporal facts TFWIN generates.

Pattern reliability modeling. From Table 1, we observe that (1) the top three frequent patterns on *country's president* have good reliability scores upon both time sources; (2) if a pattern has the words indicating past tense like "former", it has good reliability

Table 3: TFWIN wins against truth-discovery baselines on finding true *country's president* from unstructured data.

	Weight in pattern reliability estimation in Eq.(7)	Weight in tuple trustworthiness inference in Eq.(8)	Post time only		Temporal tag only		Post time + Temporal tag			
							Source unaware		Source aware	
			AUC	F1	AUC	F1	AUC	F1	AUC	F1
MAJVOTE-t [11]: return (e, v, t)-tuples			0.3022	0.4101	0.1356	0.2815	0.3336	0.4318	N/A	
MAJVOTE-$[t_s, t_e]$: return $(e, v, [t_s, t_e])$-tuples			0.4202	**0.6049**	0.1670	0.3458	**0.4958**	0.5927		
TRUTHFINDER[34]	$H_{1e-to-1v}$		AUC = 0.0006, F1 = 0.0012							
TRUEPIE[17]	$H_{1v-to-1e}$		AUC = 0.0587, F1 = 0.1430							
TFWIN	✗: 1	✗: 1	0.4411	0.6140	0.0818	0.1533	0.4403	0.6278	0.4614	0.6313
($H_{1v-to-1e}$,	✔: $c(p, (e, v, t))$	✗: 1	0.4440	0.6144	0.1094	0.1998	0.4209	0.6277	0.4680	0.6404
$H_{(1t)1e-to-1v}$,	✗: 1	✔: $c(p, (e, v, t))$	0.4713	0.6413	0.2335	0.3974	**0.5437**	**0.7242**	0.5822	0.7370
$\alpha = 10$)	✔: $c(p, (e, v, t))$	✔: $c(p, (e, v, t))$	**0.4764**	**0.6460**	0.2699	0.4340	0.4789	0.7065	**0.6146**	**0.7572**
TFWIN ($\alpha = 5$)	✔: $c(p, (e, v, t))$	✔: $c(p, (e, v, t))$	0.4737	0.6459	0.2979	0.4651	0.4405	0.6802	0.6101	0.7591
TFWIN ($\alpha = 7$)	✔: $c(p, (e, v, t))$	✔: $c(p, (e, v, t))$	0.4731	0.6448	0.2955	0.4670	0.4471	0.6829	**0.6216**	**0.7654**

Table 4: Temporal fact $(e, v, [t_s, t_e])$-tuples and the truth.

Attribute value v	Start year t_s (truth $\hat{t}_s$)	End year t_e (truth $\hat{t}_e$)
entity e = "United States", attribute a = "president"		
F. D. Roosevelt	1933	1944
H. S. Truman	1945	**1953** (**1952**)
D. D. Einsenhower	**1954** (**1953**)	1960
J. F. Kennedy	1961	**1963** (**1962**)
L. B. Johnson	**1964** (**1963**)	1968
R. M. Nixon	1969	**1974** (**1973**)
G. R. Ford	**1975** (**1974**)	1976
J. E. Carter	1977	1980
R. W. Reagon	1981	1988
G. H.W. Bush	1989	**1993** (**1992**)
W. J. Clinton	**1994** (**1993**)	2000
G. W. Bush	2001	2008
B. H. Obama	2009	2016
entity e = "Burundi", attribute a = "president"		
C. Ntaryamira	1994	1994
S. Ntibantunganya	**1995** (**1994**)	1996

upon temporal tag but poor reliability upon post time, because the person is absolutely no longer the president at the post time but the temporal tag around the pattern may present the time when the person was on the stage; (3) if a pattern has the words like "now", "newly" or "current", it has good reliability upon post time but poor reliability upon temporal tag; (4) if a pattern has the words of other occupations such as "premier" and "golfer", the reliabilities are negative upon both sources. For (3), note that if a person is newly elected as a president in a year, he/she may still not be in office, so the reliability upon post time is only 0.20.

Results on temporal fact mining. Table 4 shows the names and presidential terms of United States presidents since the year 1933. We observe that the president names v are all correct, and the t_s and t_e of TFWIN's prediction sometimes has only one year difference from the ground truth. The prediction well preserves the valid invariants $H_{1v-to-1e}$ and $H_{(1t)1e-to-1v}$.

5 RELATED WORK

In this section, we review two relevant fields to our work, temporal fact extraction and truth discovery.

5.1 Temporal Fact Extraction

The task is defined as extracting (entity, attribute name, attribute value)-tuples along with their time conditions from text corpora [4, 15, 26, 27, 35]. The concept of fact is broader than the relation between two entities. There are two series of existing natural language processing models: one is based on dependency parsing [5, 8, 21, 24], and the other is based on learning neural networks with human annotations [6, 20, 28]. These models usually work on individual sentences/paragraphs [10, 30], and suffer from high complexity and unavailability of training data [14]. It is important to leverage the data amount and evaluate the trustworthiness of extracted information using the truth finding technology. Fortunately, textual patterns, such as E-A patterns [12], parsing patterns (by PATTY [22]), and meta patterns (by METAPAD [16]), have been proposed to turn text data into structures in an unsupervised way. However, it was not designed for the problem of temporal fact extraction: it did not consider the two types of temporal contexts. We infer precise temporal slots from post time and time expressions.

5.2 Truth Discovery

The era of big data draws the serious issue of "Veracity" on resolving conflicts among multi-source information [3]. Truth discovery, which integrates multi-source noisy information by estimating the reliability of each source, has emerged as a hot topic [19]. Several truth discovery methods have been proposed for various scenarios, and they have been successfully applied in diverse application domains. TRUTHFINDER proposed the source consistency assumption, iteratively estimated source reliabilities and identified truths [34]. CRH estimated the source reliability on heterogeneous data [18] The evolution of source reliability has been explored in [32]. We propose to apply truth discovery of estimating information extractor reliability to temporal fact extraction.

6 CONCLUSIONS

In this paper, we studied a challenging problem of precise temporal slot filling and point out the limitations of existing OpenIE and PatternIE. We proposed a novel unsupervised, pattern-based, truth finding-driven framework to find precise temporal facts from text data without human annotations. Experiments demonstrated the effectiveness and efficiency.

ACKNOWLEDGEMENTS

This project is supported in part by the National Science Foundation Industry/University Cooperative Research Center on Alternative and Sustainable Intelligent Computing (ASIC) at University of Notre Dame, under grant CNS-1822099.

REFERENCES

[1] Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Vol. 1. 344–354.

[2] Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the web.. In *IJCAI*, Vol. 7. 2670–2676.

[3] Laure Berti-Equille. 2015. Data veracity estimation with ensembling truth discovery methods. In *Big Data (Big Data), 2015 IEEE International Conference on*. IEEE, 2628–2636.

[4] Melisachew Wudage Chekol. 2017. Scaling probabilistic temporal query evaluation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 697–706.

[5] Aron Culotta and Jeffrey Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd annual meeting on association for computational linguistics*. Association for Computational Linguistics, 423.

[6] Dmitriy Dligach, Timothy Miller, Chen Lin, Steven Bethard, and Guergana Savova. 2017. Neural temporal relation extraction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, Vol. 2. 746–751.

[7] Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam Mausam. 2011. Open information extraction: The second generation.. In *IJCAI*, Vol. 11. 3–10.

[8] Katrin Fundel, Robert Küffner, and Ralf Zimmer. 2006. RelExâĂŤRelation extraction using dependency parse trees. *Bioinformatics* 23, 3 (2006), 365–371.

[9] Alban Galland, Serge Abiteboul, Amélie Marian, and Pierre Senellart. 2010. Corroborating information from disagreeing views. In *Proceedings of the third ACM international conference on Web search and data mining*. ACM, 131–140.

[10] Kiril Gashteovski, Rainer Gemulla, and Luciano Del Corro. 2017. Minie: minimizing facts in open information extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2630–2640.

[11] Sally A Goldman and Manfred K Warmuth. 1995. Learning binary relations using weighted majority voting. *Machine Learning* 20, 3 (1995), 245–271.

[12] Rahul Gupta, Alon Halevy, Xuezhi Wang, Steven Euijong Whang, and Fei Wu. 2014. Biperpedia: An ontology for search applications. *Proceedings of the VLDB Endowment* 7, 7 (2014), 505–516.

[13] Alon Halevy, Natalya Noy, Sunita Sarawagi, Steven Euijong Whang, and Xiao Yu. 2016. Discovering structure in the universe of attribute names. In *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 939–949.

[14] Julia Hirschberg and Christopher D Manning. 2015. Advances in natural language processing. *Science* 349, 6245 (2015), 261–266.

[15] Tuan-Anh Hoang-Vu, Huy T Vo, and Juliana Freire. 2016. A unified index for spatio-temporal keyword queries. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 135–144.

[16] Meng Jiang, Jingbo Shang, Taylor Cassidy, Xiang Ren, Lance M Kaplan, Timothy P Hanratty, and Jiawei Han. 2017. Metapad: Meta pattern discovery from massive text corpora. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 877–886.

[17] Qi Li, Meng Jiang, Xikun Zhang, Meng Qu, Timothy P Hanratty, Jing Gao, and Jiawei Han. 2018. Truepie: Discovering reliable patterns in pattern-based information extraction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 1675–1684.

[18] Qi Li, Yaliang Li, Jing Gao, Bo Zhao, Wei Fan, and Jiawei Han. 2014. Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. ACM, 1187–1198.

[19] Yaliang Li, Jing Gao, Chuishi Meng, Qi Li, Lu Su, Bo Zhao, Wei Fan, and Jiawei Han. 2016. A survey on truth discovery. *ACM Sigkdd Explorations Newsletter* 17, 2 (2016), 1–16.

[20] Chen Lin, Timothy Miller, Dmitriy Dligach, Steven Bethard, and Guergana Savova. 2017. Representations of time expressions for temporal relation extraction with convolutional neural networks. *BioNLP 2017* (2017), 322–327.

[21] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*. Association for Computational Linguistics, 1003–1011.

[22] Ndapandula Nakashole, Gerhard Weikum, and Fabian Suchanek. 2012. PATTY: a taxonomy of relational patterns with semantic types. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, 1135–1145.

[23] Nils Reimers, Nazanin Dehghani, and Iryna Gurevych. 2016. Temporal anchoring of events for the timebank corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 2195–2204.

[24] Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 74–84.

[25] Michael Schmitz, Robert Bart, Stephen Soderland, Oren Etzioni, and others. 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*. Association for Computational Linguistics, 523–534.

[26] Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R Voss, and Jiawei Han. 2018. Automated phrase mining from massive text corpora. *IEEE Transactions on Knowledge and Data Engineering* 30, 10 (2018), 1825–1837.

[27] Avirup Sil and Silviu-Petru Cucerzan. 2014. Towards Temporal Scoping of Relational Facts based on Wikipedia Data. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*. 109–118.

[28] Alejandro Sobrino, Cristina Puente, and José Ángel Olivas. 2017. Mining Temporal Causal Relations in Medical Texts. In *International Joint Conference SOCOâĂŹ17-CISISâĂŹ17-ICEUTEâĂŹ17 León, Spain, September 6–8, 2017, Proceeding*. Springer, 449–460.

[29] Jannik Strötgen and Michael Gertz. 2015. A baseline temporal tagger for all languages. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 541–547.

[30] David Tsurel, Dan Pelleg, Ido Guy, and Dafna Shahaf. 2017. Fun Facts: Automatic Trivia Fact Extraction from Wikipedia. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 345–354.

[31] Houping Xiao, Jing Gao, Qi Li, Fenglong Ma, Lu Su, Yunlong Feng, and Aidong Zhang. 2016. Towards confidence in the truth: A bootstrapping based truth discovery approach. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1935–1944.

[32] Houping Xiao, Yaliang Li, Jing Gao, Fei Wang, Liang Ge, Wei Fan, Long H Vu, and Deepak S Turaga. 2015. Believe it today or tomorrow? detecting untrustworthy information from dynamic multi-source data. In *Proceedings of the 2015 SIAM International Conference on Data Mining*. SIAM, 397–405.

[33] Mohamed Yahya, Steven Whang, Rahul Gupta, and Alon Halevy. 2014. Renoun: Fact extraction for nominal attributes. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 325–335.

[34] Xiaoxin Yin, Jiawei Han, and S Yu Philip. 2008. Truth discovery with multiple conflicting information providers on the web. *IEEE Transactions on Knowledge and Data Engineering* 20, 6 (2008), 796–808.

[35] Chao Zhang, Fangbo Tao, Xiusi Chen, Jiaming Shen, Meng Jiang, Brian Sadler, Michelle Vanni, and Jiawei Han. 2018. TaxoGen: Constructing Topical Concept Taxonomy by Adaptive Term Embedding and Clustering. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.