Knowledge and Information Systems
https://doi.org/10.1007/s10115-020-01493-w

REGULAR PAPER

Precise temporal slot filling via truth finding with data-driven commonsense

Xueying Wang[1] · **Meng Jiang**[1]

Received: 25 February 2019 / Accepted: 6 July 2020

Abstract
The task of temporal slot filling (TSF) is to extract values of specific attributes for a given entity, called "facts", as well as temporal tags of the facts, from text data. While existing work denoted the temporal tags as single time slots, in this paper, we introduce and study the task of Precise TSF (PTSF), that is to fill two precise temporal slots including the beginning and ending time points. Based on our observation from a news corpus, most of the facts should have the two points, however, fewer than 0.1% of them have time expressions in the documents. On the other hand, the documents' post time, though often available, is not as precise as the time expressions of being the time a fact was valid. Therefore, directly decomposing the time expressions or using an arbitrary post-time period cannot provide accurate results for PTSF. The challenge of PTSF lies in finding precise time tags in noisy and incomplete temporal contexts in the text. To address the challenge, we propose an unsupervised approach based on the philosophy of truth finding. The approach has two modules that mutually enhance each other: One is a reliability estimator of fact extractors conditionally on the temporal contexts; the other is a fact trustworthiness estimator based on the extractor's reliability. Commonsense knowledge (e.g., one country has only one president at a specific time) was automatically generated from data and used for inferring false claims based on trustworthy facts. For the purpose of evaluation, we manually collect hundreds of temporal facts from Wikipedia as ground truth, including country's presidential terms and sport team's player career history. Experiments on a large news dataset demonstrate the accuracy and efficiency of our proposed algorithm.

Keywords Temporal slot · Slot filling · Truth finding · Information extraction

1 Introduction

Temporal slot filling (TSF) is one of the most important and challenging tasks in discovering knowledge from text data and building information systems. An example is to find which

✉ Meng Jiang
mjiang2@nd.edu

[1] Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556, USA

Published online: 16 July 2020

country a president belongs to as well as his/her *presidential term*,[1] in the form of a tuple such as (Mexico, Vicente Fox, [2000, 2006]), from a collection of news articles [28]. Without loss of generality, the TSF task can be formulated as below:

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

The value of the first slot □ is a country's name. It is the value of a specific attribute (e.g., country's president) for an entity (e.g., the person "vicente_fox"). 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.

In traditional TSF approaches, the temporal field only contains one slot and is filled by a direct extraction of time expression, for example, "from 2000 to 2006" in the following sentence:

"…Vicente Fox served as the President of Mexico from 2000 to 2006,…"
(entity, attribute, value, time expression)

In this case, it is not hard to decompose the time expression into PTSF result. However, based on our observation from a news dataset of ten million news articles, there are fewer than 0.1% sentences that contain at least two time points as "from 2000 to 2006". Most temporal expressions are stated shorter, such as "in 2002," which only indicates a single time point without providing enough information for *beginTime* and *endTime* slots.

A comprehensive set of precise temporal facts has been collected manually from Wikipedia as the ground truth, which includes (a) countries all over the world, names of presidents, and presidential terms (since the year of 1789) and (b) sports teams (e.g., those in the NBA, NFL, MLB, and NHL), names of players, and career history. Based on these facts, we implemented and evaluated two unsupervised information extraction (IE) methods: open-domain IE (OpenIE) and pattern-based IE (PatternIE). The results of OpenIE methods [1,2,8,11,32] showed very low precision on the value slots and lower-than-0.1 recall on the time slots. Using the above sentence as an example, OpenIE methods would generate ("vicente_fox", "serve_as", "president_of_mexico") instead of finding "mexico" as the value of attribute "is_president_of". OpenIE methods focused on extracting the relations between the subject and the object in a sentence, and it could not find precise time points due to the limited number of long time expressions as "from 2000 to 2006". On the other hand, PatternIE methods [17,18,27] focused on finding attribute values, and very little work included time information with the associated values. One pattern-based temporal anchoring method [29] assumed both time expression and document post-time, are accurate to be the time of the attribute value being valid. This assumption took a risk of tolerating too much noisy contexts from the data, and could not provide a precise temporal slot at the end. Especially for daily news, a past event could be mentioned in the current discussion. An example is given as follows:

"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),

[1] If a president has multiple terms of office, multiple tuples of the same country, the same president name and different valid time periods are expected.

In the first sentence, the post-time "2010" is not in the presidential term of President Carter (1977–1981). Meanwhile, "1987" is not in the term of President Trump for the second sentence. We question and observe the differences among different textual patterns on extracting time points. The *reliability* of them can be quite diverse 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."

Fortunately, research on "truth finding/discovery" has emerged in the field of data mining. We will review and discuss the truth finding methods in Sect. 2.2 [7,10,19,24,40,41,45,46]. They point us to a new idea of solving the problem. The fundamental idea of truth finding is resolving conflicts among multi-source information by estimating source reliability with a hypothesis that (un-)reliable sources provide (un-)trustworthy information. Effectiveness of this iterative truth discovery methodology has been demonstrated for finding the book's true author names from information on book-selling Web sites [20,44].

In this work, we introduce truth finding method to slot filling tasks and propose a novel unsupervised approach based on PatternIE. We first use PatternIE to discover a large set of textual patterns and use 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). Then we 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.

However, it wound not work if we directly applied the truth-finding algorithms to the PTSF task because those frameworks were based on the single-truth assumption, i.e., "one-fact-per-object" constraint [10,44], to define conflicts. This assumption is valid for finding true author list for a book (because a book has only one author list) but cannot be held for an entity that may have multiple values at multiple time points. For example, USA has over forty presidents in the history. And, based on our commonsense knowledge: one country is likely to have *only one* president in 1 year. If we can *learn* this kind of commonsense from data itself, we would have a chance to find conflicts, say, what is true and what is false.

We propose to generate commonsense knowledge from data using statistical methods and use the data-driven commonsense (or called *the World's invariants*) including time-irrelevant and time-relevant constraints to find the conflicts and the truth. Details are presented in Sect 4.2. Based on the above two 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 does *not* require expensive annotations or heavy parameter tuning.

Experiments on *country's president* and *sports team's player* demonstrate that TFWIN can fill the precise temporal slots with a higher accuracy than the state-of-the-art. We observe that many other types of attributes also have "invariants". For example, the number of a person's parents is likely to be two; at one time, a person is likely to have zero or one spouse in most of the countries in the world. There were no available datasets on these attributes that contain beginTime and endTime information as ground truth. Collecting more ground truth—temporal facts of more attributes such as *country's prime minister*, *city's mayor*, and *state's governor* are on our agenda; however, it will take a lot of human efforts while not providing new types of the World's invariants beyond the two attributes we study in this work. We leave them as future work.

We summarize our main contributions as follows.

- We study a challenging 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 in quasi-linear time with no requirement of human annotations.
- Experiments demonstrated the effectiveness and efficiency. AUC and F1 were improved by 25+% over the state of the art.

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

2 Related work

In this section, we review two fields related to our work, temporal fact extraction and truth discovery. The first field, temporal fact extraction, is a popular task in the fields of Spoken Language Understanding (SLU) and Natural Language Processing (NLP). The second field, truth discovery, is a branch of database and data mining research, which was not yet extended to text data or the problem of temporal fact extraction.

2.1 Temporal fact extraction

The task is defined as extracting (entity, attribute name, attribute value)-tuples along with their time conditions from text corpora [4,16,33]. 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 [5,9,26,31], and the other is based on learning neural networks with human annotations [6,25,34]. These models usually work on individual sentences/paragraphs [2,8,11,36], nd suffer from high complexity and unavailability of training data [15]. 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 (entity–attribute) patterns [13,14], S–O–V (subject–object–value) patterns [42], dependency parsing patterns (by PATTY [27]), and meta patterns (by METAPAD [17]), have been proposed to turn text data into structures in an unsupervised way. Specifically, Google's Biperpedia generated the E–A patterns (e.g., "A of E" and "E 's A") from users' fact-seeking queries by replacing entity with "E" and noun-phrase attribute with "A". ReNoun generated

the S–A–O patterns (e.g., "S's A is O" and "O, A of S,") from human-annotated corpus on a predefined subset of the attribute names. PATTY used parsing structures to generate relational patterns with semantic types. In terms of PatternIE, TruePIE [18] was proposed based on mutual enhancement between reliable patterns and reliable tuples. 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.

2.2 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 [23]. In this work, we aim at extracting precise temporal information including true entity's attribute value and true beginning and ending time points. Introducing truth discovery to the slot filling task is new and promising to solve the problem. Several truth discovery methods have been proposed for various scenarios, and they have been successfully applied in diverse application domains such as claim verification [37] and social sensing [39]. TruthFinder proposed the source consistency assumption, iteratively estimated source reliabilities and identified truths [44]. AccuSim and AccuCopy applied Bayesian analysis to capture the similarity of claimed values [7]. 2- Estimates and 3- Estimates adopted complementary voting by exploring the single truth assumption, that is "there is one and only one true value for each object" [10]. SSTF proposed semi-supervised truth discovery incorporating a small set of labeled truths [45]. LTM, a probabilistic graphical model, considered two types of errors, false positive and false negative [46]. CRH estimated the source reliability on heterogeneous data [20], and CATD derived the confidence interval for source reliability estimation [19]. The evolution of source reliability has been explored in [24,41]. Recently, Xiao et al. proposed a bootstrapping approach for efficient truth discovery [40]. Waguih et al. [38] evaluated the above methods on a set of benchmarks. Zhi et al. [47] and Yao et al. [43] adopted truth discovery to analyze numerical data and streaming data, respectively. A comprehensive survey on truth finding that was published in 2016 [23] has discussed some future directions of truth discovery, and the very first important problem is meeting unstructured data: "For most of the truth discovery approaches, they assume that the inputs are available as structured data. Nowadays, more and more applications are dealing with unstructured data such as text…The extracted inputs from unstructured data are much more noisy…" Here we propose to apply truth discovery of estimating information extractor reliability to temporal fact extraction. We address the issue of "multi-value truth" in this work. The issues of text source reliability such as deliberate fake news are out of scope [21,22].

3 Problem definition

We first introduce the techniques we use to turn the news text into "pattern-to-(entity, value, time)-tuples." Then we define the problem of precise temporal fact extraction, equivalent to precise temporal slot filling.

3.1 Preprocessing: 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) [13,14,42] or their types like $PERSON or $COUNTRY [17,27]. 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.

3.1.1 Entity recognition and typing

We use the COTYPE system [30] to jointly recognize entity names and their *fine-grained* types simultaneously. For example, country names such as "United States", "Mexico", "Russia", and "Burkina Faso" are recognized and typed as "$LOCATION.COUNTRY" (simplified as "$COUNTRY").

3.1.2 Textual pattern mining

We use the textual pattern mining method METAPAD [17] to discover "meta patterns" as information extractors. The meta-pattern is defined in [17] as below.

Definition 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.

The meta patterns that indicate the attribute name and the entity's attribute value, but not all of them are reliable. For example, the patterns [$COUNTRY president $PERSON] and [president $PERSON of $COUNTRY] can find a small but high-quality set of country's president names; the pattern [president $PERSON visited $COUNTRY] would extract wrong attribute value because a country's president never "visited" his/her home country. The reliability would be more doubtful when temporal contexts were introduced into the process of extraction. We will use all the meta patterns as extractors and estimate their reliability for finding true precise temporal facts.

3.1.3 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 "tag time" which is the nearest temporal tags (within a 20-word window) to the entity mention, if any. We use a popular tagging tool [35] to extract the temporal tags. Now we can define the EVT-tuples as below.

Definition 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.

Table 1 Symbols used throughout the paper and their descriptions

Symbol	Description
$\mathcal{D}^{(*)}$	"Pattern-tuple" extraction list
$* \in$ {"post", "tag"}	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 $\mathcal{D}^{(*)}$
p_{seed}	Seed pattern
α	Minimum tuple frequency

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

Definition 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 …").

3.2 Problem definition

Table 1 describes the symbols we use in this paper. With the above concepts defined in Sect. 3.1, we define the problem of precise temporal fact extraction on the "pattern-tuple" structured data.

Problem 1 *(Precise Temporal Fact Extraction)* ***Given*** *two "pattern-tuple" structured extraction lists* $\mathcal{D}^{(\text{post})}$ *and* $\mathcal{D}^{(\text{tag})}$*, represented as* $\mathcal{D}^{(*)} = \left\{\left(p, (e, v, t), c\right)\right\}$ *in which the time* t *can come 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,* $c^{(\text{tag})}(p, (e, v, t))$ *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* $\mathcal{F} = \left\{\left(e, v, [t_b, t_e]\right)\right\}$.

We assume the EVT-tuple whose trustworthiness $w(e, v, t)$ is perfect (exactly 1) is true. The true temporal fact tuples are derived from the fully trusted EVT-tuples. Here we assume that for each pair of e and v, there is only one valid time period $[t_b, t_e]$: $t_b = \min_{w(e,v,t)=1} t$ and $t_e = \max_{w(e,v,t)=1} t$. This work does not handle multiple valid time periods, e.g., multiple presidential terms of the same person. We expect future work to resolve this issue.

4 The proposed framework

In this section, we present the framework for precise temporal fact extraction. We introduce the overview and how to derive the invariants, following with the algorithm and complexity analysis.

4.1 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.

Convergence analysis: As we know the ground truth (i.e., the set of the World's true facts) has been a good "convergence point" that satisfies the separating criterion, the learning process can converge. However, if the size of pattern/tuple set is too small, it has a risk of not converging at the ground truth. So, we use a massive news dataset of around ten million news articles to generate a good performance in an unsupervised manner. Our experiments show that it converges within fewer than 60 iterations.

Here comes a detailed description of the learning process using Fig. 1. Suppose after structuring the text into "pattern-tuple" in Sect. 3.1, we have the 4 patterns and 10 EVT-tuples on the left-hand and middle parts of Fig. 1. How could we find out the precise temporal fact ("mexico", "vicente_fox", [2000, 2006]) if the long time expression (i.e., "from 2000 to 2006") were not available?

We assume that the most frequent EVT-tuples with either post-time $t^{(\text{post})}$ or temporal tag $t^{(\text{tag})}$, that were extracted by the *seed* pattern [$COUNTRY president $PERSON], are more likley to be trustworthy, such as ("u.s.", "george_w_bush", $2002^{(\text{post})}$) and ("u.s.", "george_w_bush", $2006^{(\text{tag})}$). Then we have a precise temporal fact ("u.s.", "george_w_bush", [2002, 2006]). This ends step (a) of the first iteration of the iterative unsupervised learning.

In step (b), based on the precise temporal fact, we know the unlabelled EVT-tuples ("u.s.", "george_w_bush", $2003^{(\text{tag})}$) and ("u.s.", "george_w_bush", $2005^{(\text{post})}$) are more likely to be true; with the constraint H2 (one president is likely to serve only one country), we know that ("iraq", "george_w_bush", $2003^{(\text{tag})}$) is likely to be false; with the constraint H3 (in 1 year, one country is likely to have only one president), we know that ("u.s.", "jimmy_carter",

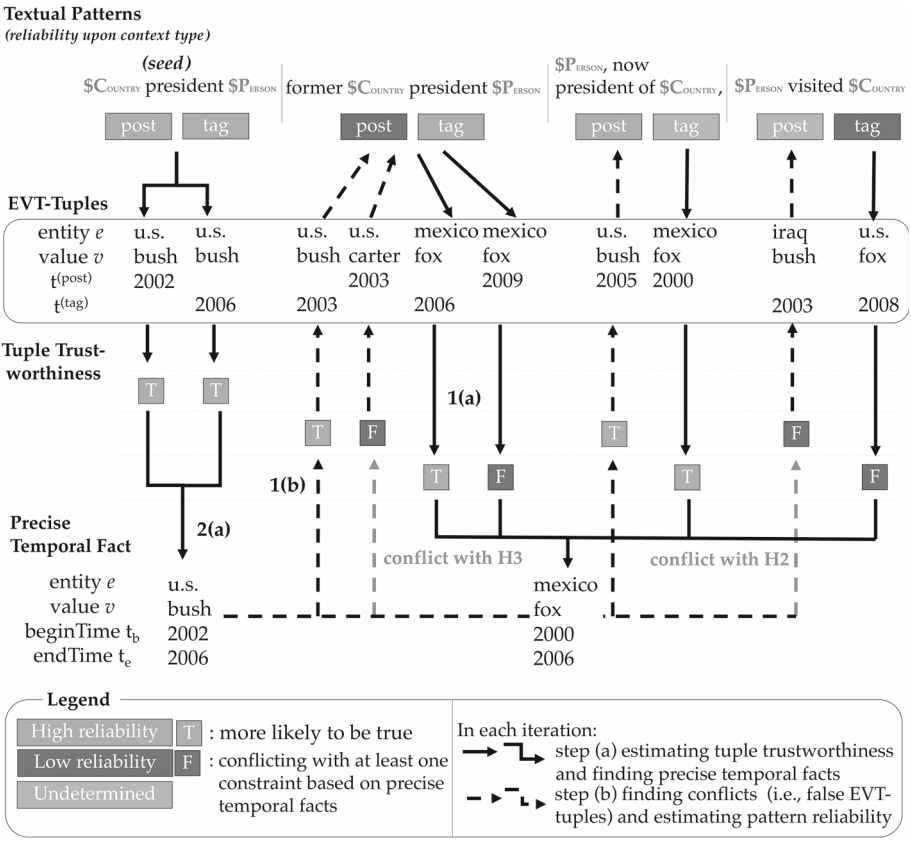


Fig. 1 Our proposed TFWIN framework iteratively estimates the reliability of textual patterns as information sources, infers trustworthiness of temporal facts, and resolves conflicts defined by the World's invariants (e.g., H2 and H3). We initialize our model by setting a seed pattern (with high reliability). Three example patterns are demonstrated in this figure to find the precise office term of "george_w_bush" (represented as "bush") and "vicente_fox" (represented as "fox"), and the country they served. Two types of time points (both post time and text time expression) are considered. Green label denotes a high reliability, while red label denotes a low reliability. And grey means we are not sure from this point. Then we extract sample values and time points from textual patterns, and format into EVT-tuples. Based on the "commonsense" learned from our data, we iteratively perform step (a) and (b) to generate the final result

$2003^{(\text{post})}$) is likely to be false. Then we can infer the pattern's reliability upon the temporal context types (post) and (tag) with the newly labelled EVT-tuples.

First, with the true ("u.s.", "george_w_bush", $2003^{(\text{tag})}$) and false ("u.s.", "jimmy_carter", $2003^{(\text{post})}$), we know that for pattern $p =$ [former COUNTRY president PERSON,], the reliability on temporal tag $r^{(\text{tag})}(p)$ should be high (i.e., close to 1) and the reliability on post time $r^{(\text{post})}(p)$ should be low (i.e., close to -1). Second, with the true ("u.s.", "george_w_bush", $2005^{(\text{post})}$), we know that for pattern $p =$ [PERSON, now president of COUNTRY,], the reliability $r^{(\text{post})}(p)$ should be high. Third, with the false ("iraq", "george_w_bush", $2003^{(\text{tag})}$), we know that for pattern $p =$ [PERSON visited COUNTRY], the reliability $r^{(\text{tag})}(p)$ should be low.

Then we do the second iteration starting with step (a). Based on the newly-estimated reliability of the patterns, we can infer the trustworthiness of the EVT-tuples that were extracted by these patterns. Among these tuples, here are the four that contain the value (i.e., president name) "vicente_fox", and we can have their trustworthiness as follows. Firstly, ("mexico", "vicente_fox", $2006^{(\text{tag})}$) is likely to be true and ("mexico", "vicente_fox", $2009^{(\text{post})}$) is likely to be false, because the [former $COUNTRY president $PERSON] has high reliability on temporal tag and low reliability on post-time. Secondly, ("mexico", "vicente_fox", $2000^{(\text{post})}$) is likely to be true because the pattern [$PERSON, now president of $COUNTRY,] has high reliability on post time. Lastly, ("u.s.", "vicente_fox", $2008^{(\text{tag})}$) is likely to be false, because the pattern [$PERSON visited $COUNTRY] has low reliability. At this point, we have as many as four EVT-tuples that tell information about the temporal fact about Vicente Fox's presidential term. We are able to confidently generate a precise temporal fact ("mexico", "vicente_fox", [2000, 2006]).

Next we will introduce in details the method of deriving the World's invariants and the algorithm design of this unsupervised learning approach.

4.2 Deriving the data-driven commonsense

Our idea is to derive the World's invariants from data as a clue to detect conflicts in the process of truth finding. The invariants are constraints on the possible number, say, *one* or *multiple*, of values/entities associated with an entity/value with/without respect to a time: They include *time-irrelevant* constraints:

- $H_{1e-to-1v}$: one entity has only *one* value on the attribute;
- $H_{1v-to-1e}$: one value is associated with only *one* entity;

and *time relevant* constraints:

- $H_{(1t)1e-to-1v}$: at a time, one entity has only *one* value;
- $H_{(1t)1v-to-1e}$: at a time, one attribute value is associated with only *one* entity.

If $H_{1e-to-1v}/H_{1v-to-1e}$ is accepted, $H_{(1t)1e-to-1v}/H_{(1t)1v-to-1e}$ will be accepted, but the opposite is not true. Note that if one hypothesis is rejected, we tend to believe the number is not likely to always be one and it cannot be a constraint for defining conflicts.

We use binomial test, one of the popular statistical hypothesis testing methods, to verify the significance of the above constraints. We set the probability of success as 0.9 and set the significance level as 0.05. Take two specific attributes, *country's president* and *sports team's player*, as example.

For *country's president*, we can generate and validate (1) time-irrelevant hypotheses:

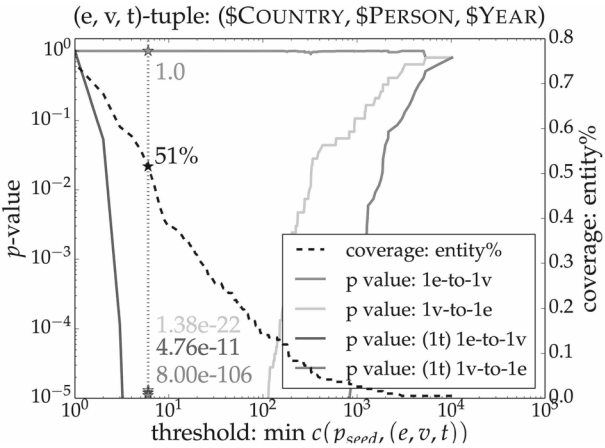
- H1: one country is likely to have *multiple* presidents in the history;
- H2: one president is likely to serve *only one* country;

and (2) time-relevant hypothesis:

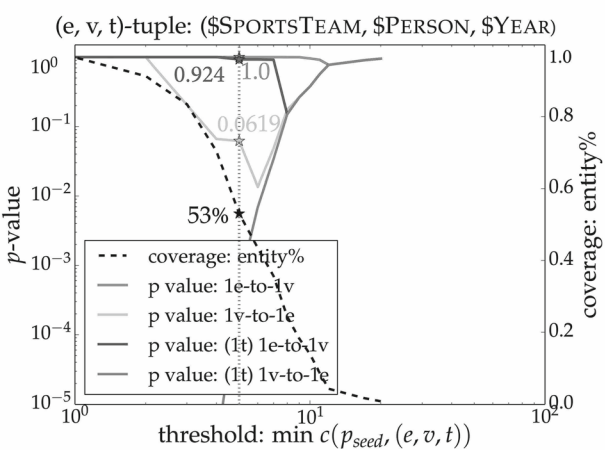
- H3: in 1 year, one country is likely to have *only one* president.

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),



(a) Invariant test on country's president



(b) Invariant test on sports team's player

Fig. 2 Hypothesis tests show that (a) for country's president, three invariants, $H_{1v-to-1e}$, $H_{(1t)1e-to-1v}$, and $H_{(1t)1v-to-1e}$ (yellow, blue, and green), are accepted; (b) for team's player, only one invariant, $H_{(1t)1v-to-1e}$ (the green), is accepted. They are valid for a wide range of the threshold of tuple's count

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.

The World's invariants found on the attribute *sport team's player* are: (1) time-irrelevant hypotheses:

- H4: one team has *multiple* players;
- H5: one player may serve *multiple* teams in his/her career;

and (2) time-relevant hypotheses:

- H6: in 1 year, one team has *multiple* players;
- H7: in 1 year, one player is more likely to serve *only one* team.

H7 can be used as a constraint to find conflicts between tuples.

Figure 2 shows the p-value v.s. threshold of the count of sample EVT-tuples for each attribute. For each attribute, we use just one seed pattern to collect a sample set of EVT-tuples: pattern [$COUNTRY president $PERSON] for country's president and [$PERSON of the $SPORTSTEAM] for team's player. With the threshold being smaller (from right to left), we have more evidence (i.e., more tuples) to test the hypotheses but also more noise from infrequent tuples when the threshold is too small. The coverage of entity names (countries and teams) is increasing. We look at the p values when the coverage is just above 50%. Here are our observation from the figures, with respect to the two types of attributes.

Country's president in Fig. 2a The p-value of $H_{1e-to-1v}$, which is the opposite of H1 in the introduction, is 1. We don't accept it. The p-values of the other three hypotheses, $H_{1v-to-1e}$, $H_{(1t)1e-to-1v}$, and $H_{(1t)1v-to-1e}$, are 1.38×10^{-22}, 4.76×10^{-11}, and 8×10^{-106}, respectively. We accept these hypothesis and use them to define conflicts. Actually, $H_{1v-to-1e}$ is the same as H2, $H_{(1t)1e-to-1v}$ is the same as H3, and $H_{(1t)1v-to-1e}$ is redundant. Here we have one time-irrelevant constraint and one time-relevant constraint:

– H2: one president is likely to serve *only one* country;
– H3: in 1 year, one country is likely to have *only one* president.

Team's player in Fig. 2b The p-values of $H_{1e-to-1v}$, $H_{1v-to-1e}$, and $H_{(1t)1e-to-1v}$ are 1, 0.92, and 0.06, respectively. We reject the hypotheses especially when the tuples are quite sparse in the data: our dataset is from general news, not massive sports news. The p-value of $H_{(1t)1v-to-1e}$ is as small as 5.31×10^{-4}. The hypothesis is equivalent with H7. So we have one time-relevant constraint:

– H7: in 1 year, one player is more likely to serve *only one* team.

The above figures show that the range of valid threshold values is wide. We study these two attributes and leave other attributes such as *country's foreign minister*, *person's parents*, and *person's spouse* for future work. The key challenge is on the collection of ground truth data.

4.3 The iterative learning algorithm

Generally, the algorithm is an iterative method. It starts with very light supervisory information—we use a textual pattern, called *seed pattern* (denoted by p_{seed}). It 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. Experiments will show the performance is insensitive to α.

Algorithm 1: The truth finding algorithm in TFWIN

Input : "pattern-EVT tuple" data $\mathcal{D}^{(*)}$, constraints $\mathcal{H}$, one seed pattern p_{seed} based on attribute a
Output: a list of temporal fact tuples $\mathcal{F}$ of the attribute a

Initialize $\mathcal{F} \leftarrow []$, $r^{(*)}(p_{\text{seed}}) = 1$, $r^{(*)}(p) = 0$ if $p \neq p_{\text{seed}}$;
Set of undetermined EVT-tuples $\mathcal{D}_{"U"} \leftarrow \mathcal{D}$;
do
  foreach $(e, v, t) \in \mathcal{D}_{"U"}$ **do**
    Infer trustworthiness score $w(e, v, t)$ by Eq. (8);
  end
  for $* \in$ *{"post", "tag"}* **do**
    for $(e, v, t) \in \mathcal{D}_{"U"}$ *sorted by* $w(e, v, t) \geq 0$ **do**
      switch $f((e, v, t), \mathcal{F}, \mathcal{H})$ **do**
        case *"T"*
          continue;
        endsw
        case *"F"*
          break;
        endsw
        case *"U"*
          if $\exists(e, v, [t_b, t_e]) \in \mathcal{F}$ **then**
            Update $(e, v, [\min\{t, t_b\}, \max\{t, t_e\}])$;
          else
            Add $(e, v, [t, t])$ into $\mathcal{F}$;
          end
        endsw
      endsw
    end
  end
  foreach $(e, v, t) \in \mathcal{D}$ *and* $\sum_{*,p} c^{(*)}(p, (e, v, t)) \geq \alpha$ **do**
    Assign polarized trustworthiness $w((e, v, t))$ by Eq. (6);
  end
  $\mathcal{D}_{"U"} \leftarrow \{(e, v, t) | w(e, v, t) = 0\}$;
  for $* \in$ *{"post", "tag"}* **do**
    foreach $p \in \mathcal{P}$ **do**
      Estimate the reliability $r^{(*)}(p)$ by Eq. (7);
    end
  end
while *Convergence ($\mathcal{F}$ doesn't change)*;
return $\mathcal{F}$;

4.3.1 Conflicts and negative label generation

Here we define a function of checking whether 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

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

is true. $f((e, v, t), \mathcal{F}, \{H_{1e-to-1v}\}) =$ "F'', if the statement

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

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

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

is true. $f((e, v, t), \mathcal{F}, \{H_{(1t)1e-to-1v}\}) =$ "F'', if the statement

$$\exists(e, v', [t_b, t_e]) \in \mathcal{F}, t_b \leq t \leq t_e \text{ and } v' \neq v \tag{4}$$

is true. $f((e, v, t), \mathcal{F}, \{H_{(1t)1v-to-1e}\}) =$ "F'', if the statement

$$\exists(e', v, [t_b, t_e]) \in \mathcal{F}, t_b \leq t \leq t_e \text{ and } e' \neq e \tag{5}$$

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.

Example If we have ("u.s.", "george_w_bush", [2002, 2006]) in $\mathcal{F}^{(i)}$, then w(("u.s.", "george_w_bush", 2003)) $= 1$, w(("u.s.", "jimmy_carter", 2003)) $= -1$, and w(("iraq", "george_w_bush", 2003)) $= -1$, because we hold $H_{1v-to-1e}$ and $H_{(1t)1e-to-1v}$.

4.3.2 Pattern reliability estimation

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

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

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.

4.3.3 Tuple trustworthiness inference

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

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

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

4.4 A supervised version of TFWIN: supTFWIN

TFWIN requires very little supervisory information. The algorithm needs only a seed pattern to initiate for finding all the temporal facts for any attribute. It would not be hard to find one pattern for an attribute, for example, we can use the pattern [$COUNTRY president $PERSON] for attribute *country's president*; we can use the pattern [$SPORTSTEAM player $PERSON] for attribute *sports team's player*. Practitioners do not have to manually label *concrete true temporal fact tuples* for learning.

However, we can absolutely extend the TFWIN algorithm into a supervised version, when the supervisory information, i.e., a set of true temporal fact tuples, is available for replacing the seed pattern in initialization. We name the algorithm "supTFWIN." We will investigate the performance of supTFWIN and compare it with the original TFWIN in experiments.

To maintain the fairness of comparing with TFWIN, we are not going to feed supTFWIN with *manually* labelled tuples. We assume that more frequent tuples are more likely to be true. In supTFWIN, we sort raw EVT-tuples (e, v, t) by their frequency $\sum_* \sum_p c^{(*)}(p, (e, v, t))$ in the data and use the top ranked tuples as "labeled samples". We will investigate the performance of supTFWIN w.r.t the number of labeled samples. The intuition is that having a small set of labels will maintain a high precision but make a low recall; and having a big set may improve recall but hurt precision. Experimental results will answer whether it is more effective to use a seed pattern for initialization or a set of frequent tuples.

4.5 Complexity analysis

In Algorithm 1, for each iteration, the time complexity is $\mathcal{O}(n_p + n_t + n_t \log n_t)$, where (a) $n_p = |\mathcal{P}|$ is the size of the set of patterns $\mathcal{P}$ and (b) $n_t = |\mathcal{D}|$ is the size of the set of all EVT-tuple candidates $\mathcal{D}$. First, based on Eqs. (7) and (8), the complexity of updating the pattern reliability is linear to the number of patterns n_p, and the complexity of updating tuple trustworthiness is linear to the number of tuples n_t. Second, a sorting function must be used when putting trustworthy tuples of higher score than a certain threshold into the true tuple set. So the complexity is quasi-linear to the total number of tuples n_t.

The algorithm convergence has been discussed in the Overview section. Empirical study will show that n_p and n_t are proportional to the size of text corpus; the number of iterations is around 50. So our algorithm is scalable for mining temporal facts from massive text data: In fact, to overcome the incompleteness and noise in temporal contexts, it is desired that more related text data, better performance (as demonstrated in the experiments, Fig. 5b).

5 Experiments

Here we conduct experiments to answer the following questions:

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?

Q3. (Scalability and parameter-insensitivity) Is TFWIN efficient and scalable to the corpus size? We also investigate different settings for reliability estimation ($r(p)$ in Eq. (7)) and trustworthiness inference ($w((e, v, t))$ in Eq. (8)) and the sensitivity of parameter α.

In this section, we will first introduce experimental settings and baseline methods and then provide both quantitative analysis and qualitative analysis to answer the above questions.

5.1 Experimental settings

In this section, we first give text data description following with the ground truth introduction. Then we present the evaluation methods, baseline methods, and parameter settings.

5.1.1 Text data description

The dataset has 9,876,086 news articles (4 billion words) published from 1994 to 2010 by 6 international newswire sources, including Agence France-Presse, Central News Agency of Taiwan, Los Angeles Times/Washington Post, Associated Press Worldstream, New York Times, and Xinhua News Agency. Date of being posted is available for every document.

5.1.2 Structured data size and ground truth

We focus on finding precise temporal facts on *country's president* and *sportsteam's player*. We first extract temporal information, both post time and tag time, from the large, raw text dataset. For post time, we attach it to all the fact tuples in that news document. For tag time, we assign the nearest temporal tag, if available within a 20-word window, to fact tuples. 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 3175 (e, v, t)-tuples. For *sportsteam's player*, we have 125,726 patterns of $SPORTSTEAM and $PERSON, 107,636 unique EVT-tuples and 287,319 extractions. It was expensive to collect the ground truth, so we only conduct qualitative analysis.

5.1.3 Evaluation methods

We evaluate the performance of our method and all baselines on mining the 3175 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.

5.1.4 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 2, the philosophy, as well as the settings, of the methods are given as follows.

- MajVote-t [12]: it uses the weighted majority voting strategy and returns the most frequent (e, v, t)-tuples, no hypotheses are considered;
- MajVote-$[t_s, t_e]$: we modify the general MajVote by compositing frequent (e, v, t)-tuples into $(e, v, [t_s, t_e])$-tuples, where $t_s = \min t$ and $t_e = \max t$, no hypotheses are considered;
- TruthFinder-$H_{1e-to-1v}$ [44]: 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.
- LTM [46]: This is a probabilistic graphical model. Sensitivity and specificity are used to infer the truth. It assumes hypothesis $H_{1v-to-1e}$;
- CRH [20]: we modify CRH to only apply on categorical data, it considers source reliability and assumes hypothesis $H_{1v-to-1e}$;
- TruePIE [18]: it assumes hypothesis $H_{1v-to-1e}$ without considering time-relevant invariants which are very important for temporal fact extraction;
- 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)).
- supTFWIN: it is the supervised version of TFWIN. It uses a set of the most frequent fact tuples for initialization (like training). It has a parameter, i.e., the number of training samples. We will investigate its performance w.r.t. the parameter.

We study modeling two time signal sources (post time and temporal tag) in the structure. We will investigate methods using each source only and using an integration. For the integration, source-unaware modeling assumes that one pattern has the same reliability regardless to the time source and source-aware modeling learns different reliabilities of the same pattern for different sources. We set the default threshold, i.e., minimum tuple frequency, as $\alpha = 10$.

5.2 Experimental results

In this section, we present experimental results to demonstrate the effectiveness, interpretability, and practical properties (efficiency and parameter insensitivity) of the proposed method.

5.2.1 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. Figure 3 presents the precision–recall curves.

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**%).

Table 2 Methods and their corresponding usage of weight in pattern reliability estimation and tuple trustworthiness inference

Method	Weight in pattern reliability estimation in Eq. (7)	Weight in tuple trustworthiness inference in Eq. (8)
MajVote-t [12]: return (e,v,t)-tuples		
MajVote-$[t_s, t_e]$: return $(e, v, [t_s, t_e])$-tuples		
TruthFinder [44]	$H_{1e-to-1v}$	
LTM [46]	$H_{1e-to-1v}$	
CRH [20]	$H_{1e-to-1v}$	
TruePie [18]	$H_{1v-to-1e}$	
TFWIN	✗: 1	✗: 1
($H_{1v-to-1e}$,	✔: $c(p,(e,v,t))$	✗: 1
$H_{(1t)1e-to-1v}$,	✗: 1	✔: c(p,(e,v,t))
$\alpha = 10$)	✔: $c(p,(e,v,t))$	✔: $c(p,(e,v,t))$
TFWIN ($\alpha = 5$)	✔:$c(p,(e,v,t))$	✔: $c(p,(e,v,t))$
TFWIN ($\alpha = 7$)	✔: $c(p,(e,v,t))$	✔: $c(p,(e,v,t))$
supTFWIN	✔: $c(p,(e,v,t))$	✔: $c(p,(e,v,t))$

The red curve with dots in Fig. 3 shows the (precision, recall)-scores at each iteration. The curve starts with (1.0, 0.029) generated from the seed pattern and then in each iteration, the precision drops a little bit and the recall significantly increases till a convergence of (0.907, 0.646) at the 62nd iteration. Details of different experiments settings will be discussed as follows:

Understanding the performance with upper bound Take the attribute *country's president* as an example. As we explained in the paper, we collected the ground truth fact tuples by manually searching on Wikipedia and Google. Without looking at the news documents, we tried our best to find as many true tuples as possible. Therefore, the news data cannot fully cover all the ground truth fact tuples: only when there was at least one sentence that could be matched to one textual pattern for a specific fact tuple, the fact tuple would have a chance to be extracted (and inferred as a true fact at the end).

Figure 3 reflects the coverage of the news data we have. The method MajVote would eventually include all the fact tuples into the true tuple set. Ideally, if the news data can cover all the tuples in the ground truth, though the precision would significantly drop, the recall would extend to close to 1. However, the curve of MajVote stops at recall of 0.67. So the maximum F1 score is only 0.802 (when precision is 1.0).

Compared to this upper bound (F1 score of 0.802), the performance of our proposed algorithm, with F1 score of 0.765, is very good. It demonstrates the salience of the information extraction system. More evidence is that in Fig. 3, our proposed TFWIN achieved a very similar recall with the method MajVote. Again, our method improved F1 relatively by 25%.

Truth finding versus majority voting In Fig. 3, the yellow curve shows the (precision, recall)-scores of MajVote-$[t_s, t_e]$. The curve ends at (0.530, 0.651): Note that 0.651 is the highest recall that *any* algorithm can achieve based on the "pattern-tuple" structures from the text data. We collected a bigger set of the ground truth (i.e., presidential terms)

Table 3 The proposed TFWIN model outperforms existing methods in terms of AUC, and F1 scores on finding true *country's president* from unstructured data

Method	Post time only		Temporal tag only		Post time + Temporal tag			
					Source unaware		Source aware	
	AUC	F1	AUC	F1	AUC	F1	AUC	F1
MajVote-t	0.3022	0.4101	0.1356	0.2815	0.3336	0.4318	N/A	
MajVote-$[t_s, t_e]$	0.4202	**0.6049**	0.1670	0.3458	**0.4958**	0.5927		
TruthFinder	AUC = 0.0006, F1 = 0.0012							
LTM	AUC = 0.0957, F1 = 0.1193							
CRH	AUC = 0.1946, F1 = 0.2113							
TruePie	AUC = 0.0587, F1 = 0.1430							
TFWIN	0.4411	0.6140	0.0818	0.1533	0.4403	0.6278	0.4614	0.6313
($H_{1v-to-1e}$,	0.4440	0.6144	0.1094	0.1998	0.4209	0.6277	0.4680	0.6404
$H_{(1t)1e-to-1v}$.	0.4713	0.6413	0.2335	0.3974	**0.5437**	**0.7242**	0.5822	0.7370
$\alpha = 10$)	**0.4764**	**0.6460**	0.2699	0.4340	0.4789	0.7065	**0.6146**	**0.7572**
TFWIN ($\alpha = 5$)	0.4737	0.6459	0.2979	0.4651	0.6101	0.6802	0.6101	0.7591
TFWIN ($\alpha = 7$)	0.4731	0.6448	0.2955	0.4670	0.4471	0.6829	**0.6212**	**0.7654**
supTFWIN	0.3900	0.6032	0.2559	0.4298	0.3026	0.5372	**0.5696**	**0.7382**

Bold values are the highest indicating that the method delivers the best performance
Associate with Table 2, relatively higher performance are found, when consider both weight count in pattern and tuple reliability estimate. The best of TFWIN ($\alpha = 7$) shows an AUC of 0.6216 and an F1 of 0.7654

from Wikipedia. Our TFWIN achieves a recall of 0.646 (**−0.7**%) and a precision of 0.907 (**+71.1**%). This demonstrates the effectiveness of estimating pattern reliability and finding truth.

Table 3 shows that MajVote-$[t_s, t_e]$ performs significantly better than MajVote-*t* (+48.6% AUC; +37.3% F1) because the attribute *country's president* has $[t_s, t_e]$-period property.

The power of the World's Invariants Comparing with existing iterative-based truth finding methods: TruthFinder [44] 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 [18] 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.

Integrating two time information sources Post time signals cover a range of years [1994, 2010] and temporal tags cover a range of [1600, 2050]. Table 3 shows that the performances of *temporal tag only* are consistently worse than the performances of *post time only*. Figure 3 also shows that TFWIN only with post time can achieve a similar performance as TFWIN with both time sources. While TFWIN with temporal tag can not. The reason is that temporal tag signals are *sparser and noisier* than post time signals. MajVote has no source reliability estimation on noisy EVT-tuples, so the integration does not improve much over the post-

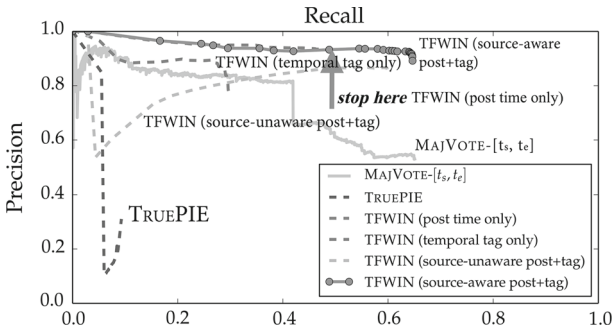


Fig. 3 Precision–recall curves that compare truth discovery algorithms on finding *country's president* from text. The performance of TruthFinder is too fragile to shown in this figure, so we do not include it here. The green line (TFWIN(post time only)) has a quite similar performance as (TFWIN(source-aware post+tag)), but it stops earlier than the green one

time-only. For our TFWIN, the source-unaware integration, that simply merges the two "pattern-tuple" structures, consistently wins over any single source: the best single-source holds an AUC of 0.4764 and an F1 of 0.6460. The best AUC of the integration is 0.5437 (**+14.1**%) and the best F1 is 0.7242 (**+12.1**%).

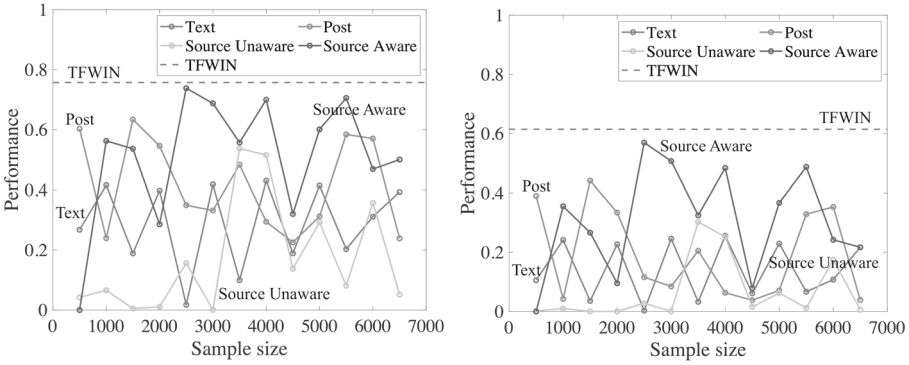
Source-aware reliability modeling

This design in our TFWIN assumes that upon different time sources, one textual pattern may have different reliability scores. Source-unaware modeling assigns only one reliability score to each pattern regardless to the time source. The novel design improves the AUC from 0.4789 to 0.6146 (**+28.3**%) and improves the F1 from 0.7065 to 0.7572 (**+7.18**%). We will provide pattern examples on comparing their reliability scores in the next subsection (Interpretability).

Weight in pattern reliability estimation and trustworthiness inference Table 3 shows that it can significantly improve the performance when the method considers pattern-to-tuple counts $c(p, (e, v, t))$ in trustworthiness inference (see Eq. (8)), because if a tuple is more often extracted by a reliable pattern, it tends to be more trustworthy. We find that it improves the AUC and F1 a little bit when the method considers $c(p, (e, v, t))$ in pattern reliability estimation (see Eq. (7)), because if a pattern more often extracts trustworthy EVT-tuples, the pattern tends to be more reliable.

Supervised TFWIN We used grid search to tune supTFWIN to the best performance. It achieved the highest accuracy and F1 score when the sample size was 2500. Compared to the total number of unique EVT-tuples (116,631), it takes only 2.1%; however, it would be rather difficult, or even impossible, for human annotators to label as many as 2500 true fact tuples for one attribute. So the original TFWIN algorithm, which was initialized with only one seed pattern, would be much more practical than supTFWIN.

Then we discuss about supTFWIN's performance from two perspectives. First, the last row of Table 3 presents the AUC and F1 score of supTFWIN, given different settings such as (1) post time only, temporal tag only, or both, and (2) source-unaware or source-aware. We observe that supTFWIN consistently falls behind the best TFWIN algorithm: through all possible values of number of label samples, the best AUC is 0.5696, still smaller than TFWIN's 0.6212; the highest F1 score is 0.7382, still lower than TFWIN's 0.7654. We also observe that when using post time only or temporal tag only, supTFWIN's performances were much worse than combining the two types of signals together; and when combining them



(a) F1 Score of Supervised TFWIN model **(b)** AUC Score of Supervised TFWIN model

Fig. 4 Performance of supervised TFWIN w.r.t. the number of label samples (i.e., frequent fact tuples for initialization). **a** shows F1 score and **b** gives AUC score. They make the same conclusion that TFWIN model (based on seed pattern) performs better than supTFWIN. In supTFWIN, source-aware modeling achieves the best performance, and tag-only modeling performs the worst. The supervised algorithm has *poor* stability

together, the source-aware algorithm can perform significantly better than the source-unaware version.

Second, Fig. 4 shows the AUC and F1 score of supTFWIN w.r.t. different values of number of label samples, compared to the dashed line which is given by the seed pattern-based TFWIN method. Clearly, the supervised version of TFWIN, supTFWIN, has very poor stability. The performance (i.e., scores) varies significantly when the parameters are different, and even worse, the change is unpredictable. We observe that the curves of supTFWIN under different settings are all below the dashed line. This demonstrates that initializing the truth discovery algorithm with one seed pattern is a better idea than doing it with a set of samples.

5.2.2 Interpretability

Table 4 presents the reliability scores given to a few pattern examples by our algorithm, where $r^{(\text{post})}(p)$ and $r^{(\text{tag})}(p)$ 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 on both temporal context types, 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 trustworthiness of the corresponding slots.

Pattern reliability modeling From Table 4, 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 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

Table 4 Pattern's reliability scores for *country's presidential term* and *sports team's player career history*

Textual pattern p	$r^{(\text{post})}(p)$	$r^{(\text{tag})}(p)$
$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
$SPORTSTEAM star $PERSON	1	1
$SPORTSTEAM quarterback $PERSON	1	1
$SPORTSTEAM center $PERSON	0.97	1
$PERSON of the $SPORTSTEAM	0.97	0.96
$PERSON and the $SPORTSTEAM	0.57	0.20
$SPORTSTEAM and $PERSON	−0.60	−0.64

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.

For the attribute *sportsteam's player*, we find that patterns of words on player's positions such as "quarterback" and "center" can be automatically predicted as reliable patterns.

Results on temporal fact mining Table 5 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 1 year difference from the ground truth. The prediction well preserves the valid invariants $H_{1v-to-1e}$ and $H_{(1t)1e-to-1v}$. For *sportsteam's player*, TFWIN uses the invariant $H_{(1t)1v-to-1e}$ and it finds Karl Malone played for Utah Jazz 1996–2001 and for Lakers in 2004: one player does not play for multiple teams in 1 year. And it finds Kobe Bryant played for L.A. Lakers 1997–2010 with 1-year overlap with Karl Malone because we know $H_{(1t)1e-to-1v}$ does not hold.

Limitations (1) It predicts errors at boundary time values when the time is at coarse-grained level (like *year*) but if it goes to fine-grained level (*month* or *day*), the signals become too sparse. (2) The constraints are hard not soft, so it does not allow two presidents in the same year but Burundi's President Ntaryamira passed away in 1994 when he was in office and President Ntibantunganya took the office in the same year, and it allows one president to have only one presidential term which is not always true in the World. (3) For *player* and many other attributes, the text data do not often contain most of the facts in the post period (e.g., Kobe Bryant in 2016 or Abdul-Jabbar in 1989). Then it becomes impossible to extract them.

Table 5 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. Reagan	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
Entity e = "L.A. Lakers", attribute a = "player"		
K. Abdul-Jabbar	1976 (1975)	1977 (1989)
Kobe Bean Bryant	1997 (1996)	2010 (2016)
Karl Malone	2004	2004
Entity e = "Utah Jazz", attribute a = "player"		
Karl Malone	1996 (1985)	2001 (2003)

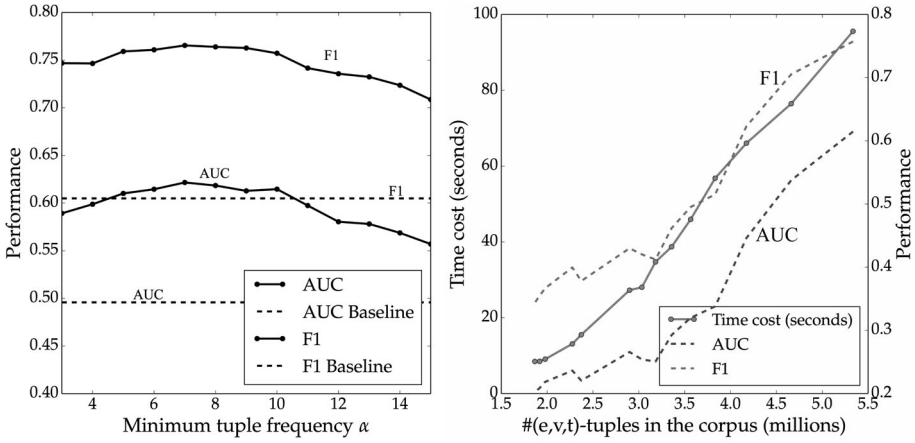
Wrong results are shown in front of brackets, with their ground truth in the brackets. Most of the errors only have 1 year difference

5.2.3 Practical properties

Figure 5 presents two important properties of our TFWIN method as below on putting into practice.

Parameter-insensitivity The AUC and F1 scores are not sensitive when the minimum tuple frequency threshold α ranges from 3 to 15. The proposed method consistently outperforms the baseline.

Scalability We apply our method on a laptop with 2.5 GHz Intel Core i7 using only one thread. It takes only 96 s and the time cost is linear to the number of EVT-tuples in the text data which is proportional to the corpus size. The AUC and F1 improve along with the data size. We believe that with extraordinary massive text data, our method will perform even better.



(a) Insensitivity to α **(b)** Scalablity to corpus size

Fig. 5 Two important properties of TFWIN

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 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 proposed a novel unsupervised, pattern-based, truth finding-driven framework to find precise temporal facts from massive general corpora in quasi-linear time with no requirement of human annotations. Experiments on two attributes (country's president and sports team's player) demonstrated the effectiveness and efficiency. AUC and F1 were improved by 25+% over the state-of-the-art.

Acknowledgements Our research was supported by National Science Foundation IIS-1849816.

References

1. Angeli G, Premkumar MJJ, Manning CD (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, pp 344–354
2. Banko M, Cafarella MJ, Soderland S, Broadhead M, Etzioni O (2007) Open information extraction from the web. In: 'IJCAI', vol 7, pp 2670–2676
3. Berti-Equille L (2015) Data veracity estimation with ensembling truth discovery methods. In: 2015 IEEE international conference on big data (big data). IEEE, pp 2628–2636
4. Chekol MW (2017) Scaling probabilistic temporal query evaluation. In: Proceedings of the 2017 ACM on conference on information and knowledge management. ACM, pp 697–706
5. Culotta A, Sorensen J (2004) Dependency tree kernels for relation extraction. In: Proceedings of the 42nd annual meeting on association for computational linguistics. Association for Computational Linguistics, p 423
6. Dligach D, Miller T, Lin C, Bethard S, Savova G (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, pp 746–751

7. Dong XL, Berti-Equille L, Srivastava D (2009) Integrating conflicting data: the role of source dependence. Proc VLDB Endow 2(1):550–561
8. Etzioni O, Fader A, Christensen J, Soderland S, Mausam M (2011) Open information extraction: the second generation. In: 'IJCAI', vol 11, pp 3–10
9. Fundel K, Küffner R, Zimmer R (2006) Relexrelation extraction using dependency parse trees. Bioinformatics 23(3):365–371
10. Galland A, Abiteboul S, Marian A, Senellart P (2010) Corroborating information from disagreeing views. In: Proceedings of the third ACM international conference on Web search and data mining. ACM, pp 131–140
11. Gashteovski K, Gemulla R, Del Corro L (2017) Minie: minimizing facts in open information extraction. In: Proceedings of the 2017 conference on empirical methods in natural language processing, pp 2630–2640
12. Goldman SA, Warmuth MK (1995) Learning binary relations using weighted majority voting. Mach Learn 20(3):245–271
13. Gupta R, Halevy A, Wang X, Whang SE, Wu F (2014) Biperpedia: an ontology for search applications. Proc VLDB Endow 7(7):505–516
14. Halevy A, Noy N, Sarawagi S, Whang SE, Yu X (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, pp 939–949
15. Hirschberg J, Manning CD (2015) Advances in natural language processing. Science 349(6245):261–266
16. Hoang-Vu T-A, Vo HT, Freire J (2016) A unified index for spatio-temporal keyword queries. In: Proceedings of the 25th ACM international on conference on information and knowledge management. ACM, pp 135–144
17. Jiang M, Shang J, Cassidy T, Ren X, Kaplan LM, Hanratty TP, Han J (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, pp 877–886
18. Li Q, Jiang M, Zhang X, Qu M, Hanratty TP, Gao J, Han J (2018) Truepie: discovering reliable patterns in pattern-based information extraction. In: 'Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 1675–1684
19. Li Q, Li Y, Gao J, Su L, Zhao B, Demirbas M, Fan W, Han J (2014) A confidence-aware approach for truth discovery on long-tail data. Proc VLDB Endow 8(4):425–436
20. Li Q, Li Y, Gao J, Zhao B, Fan W, Han J (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, pp 1187–1198
21. Li X, Meng W, Clement TY (2016) Verification of fact statements with multiple truthful alternatives. In: 'WEBIST (2)', pp 87–97
22. Li X, Meng W, Yu C (2011) T-verifier: verifying truthfulness of fact statements. In: 2011 IEEE 27th international conference on data engineering (ICDE). IEEE, pp 63–74
23. Li Y, Gao J, Meng C, Li Q, Su L, Zhao B, Fan W, Han J (2016) A survey on truth discovery. ACM Sigkdd Explor Newsl 17(2):1–16
24. Li Y, Li Q, Gao J, Su L, Zhao B, Fan W, Han J (2015) On the discovery of evolving truth. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 675–684
25. Lin C, Miller T, Dligach D, Bethard S, Savova G (2017) Representations of time expressions for temporal relation extraction with convolutional neural networks. BioNLP 2017:322–327
26. Mintz M, Bills S, Snow R, Jurafsky D (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, pp 1003–1011
27. Nakashole N, Weikum G, Suchanek F (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, pp 1135–1145
28. Parker R, Graff D, Kong J, Chen K, Maeda K (2009) English gigaword fourth edition ldc2009t13. Linguistic Data Consortium, Philadelphia
29. Reimers N, Dehghani N, Gurevych I (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, pp 2195–2204
30. Ren X, Wu Z, He W, Qu M, Voss CR, Ji H, Abdelzaher TF, Han J (2017) Cotype: joint extraction of typed entities and relations with knowledge bases. In: Proceedings of the 26th international conference on world wide web', international world wide web conferences steering committee, pp 1015–1024

31. Riedel S, Yao L, McCallum A, Marlin BM (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, pp 74–84
32. Schmitz M, Bart R, Soderland S, Etzioni O 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. Association for Computational Linguistics, pp 523–534
33. Sil A, Cucerzan S-P (2014) Towards temporal scoping of relational facts based on wikipedia data. In: Proceedings of the eighteenth conference on computational natural language learning, pp 109–118
34. Sobrino A, Puente C, Olivas JÁ (2017) Mining temporal causal relations in medical texts. In: International joint conference SOCO17-CISIS17-ICEUTE17 León, Spain, September 6–8, 2017, Proceeding. Springer, pp 449–460
35. Strötgen J, Gertz M (2015) A baseline temporal tagger for all languages. In: Proceedings of the 2015 conference on empirical methods in natural language processing, pp 541–547
36. Tsurel D, Pelleg D, Guy I, Shahaf D (2017) Fun facts: Automatic trivia fact extraction from wikipedia. In: Proceedings of the tenth ACM international conference on web search and data mining. ACM, pp 345–354
37. Vydiswaran V, Zhai C, Roth D (2011) Content-driven trust propagation framework. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp 974–982
38. Waguih DA, Berti-Equille L (2014) Truth discovery algorithms: an experimental evaluation. arXiv:1409.6428
39. Wang D, Kaplan L, Le H, Abdelzaher T (2012) On truth discovery in social sensing: a maximum likelihood estimation approach. In: Proceedings of the 11th international conference on information processing in sensor networks. ACM, pp 233–244
40. Xiao H, Gao J, Li Q, Ma F, Su L, Feng Y, Zhang A (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, pp 1935–1944
41. Xiao H, Li Y, Gao J, Wang F, Ge L, Fan W, Vu LH, Turaga DS (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, pp 397–405
42. Yahya M, Whang S, Gupta R, Halevy A (2014) Renoun: fact extraction for nominal attributes. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 325–335
43. Yao L, Su L, Li Q, Li Y, Ma F, Gao J, Zhang A (2018) Online truth discovery on time series data. In: Proceedings of the 2018 SIAM international conference on data mining. SIAM, pp 162–170
44. Yin X, Han J, Philip SY (2008) Truth discovery with multiple conflicting information providers on the web. IEEE Trans Knowl Data Eng 20(6):796–808
45. Yin X, Tan W (2011) Semi-supervised truth discovery. In: Proceedings of the 20th international conference on world wide web. ACM, pp 217–226
46. Zhao B, Rubinstein BI, Gemmell J, Han J (2012) A bayesian approach to discovering truth from conflicting sources for data integration. Proc VLDB Endow 5(6):550–561
47. Zhi S, Yang F, Zhu Z, Li Q, Wang Z, Han J (2018) Dynamic truth discovery on numerical data. In: 2018 IEEE international conference on data mining (ICDM). IEEE, pp 817–826

Xueying Wang is a master student in the Department of Computer Science and Engineering at the University of Notre Dame. She obtained her bachelor degree in communication engineering from Shanghai Normal University Tianhua College and first master degree in engineering science from University of the Pacific. Her research interests include information extraction, text data mining, and natural language processing. She was acknowledged as Grad Member Spotlight of Society of Women Engineers in 2016. She won the Outstanding Research Poster Award at a departmental event in 2018.

Meng Jiang is an assistant professor in the Department of Computer Science and Engineering at the University of Notre Dame. He received his Ph.D. degree and B.E. degree from the Department of Computer Science and Technology of Tsinghua University in 2015 and 2010, respectively. He was a postdoctoral research associate at the Computer Science Department in the University of Illinois at Urbana-Champaign from 2015 to 2017. His research interests focus on data mining, user behavior modeling, and information extraction. His paper was selected as one of the best finalists at the ACM SIGKDD Conference on Knowledge Discovery and Data Mining 2014.