TISCO: Temporal Scoping of Facts

Anisa Rula[a,c,*], Matteo Palmonari[a], Simone Rubinacci[b], Axel-Cyrille Ngonga Ngomo[d], Jens Lehmann[c], Andrea Maurino[a], Diego Esteves[c]

[a] *University of Milano-Bicocca,*
E-mail: (rula, palmonari, maurino)@disco.unimib.it
[b] *Department of Statistics, University of Oxford,*
E-mail: (s.rubinacci)@stats.ox.ac.uk
[c] *University of Bonn,*
E-mail: (anisa.rula,jens.lehmann,esteves)@cs.uni-bonn.de
[d] *University of Paderborn,*
E-mail: (axel.ngonga)@upb.de

Abstract

Some facts in the Web of Data are only valid within a certain time interval. However, most of the knowledge bases available on the Web of Data do not provide temporal information explicitly. Hence, the relationship between facts and time intervals is often lost. A few solutions are proposed in this field. Most of them are concentrated more in extracting facts with time intervals rather than trying to map facts with time intervals. This paper studies the problem of *determining the temporal scopes of facts*, that is, deciding the time intervals in which the fact is valid. We propose a generic approach which addresses this problem by curating temporal information of facts in the knowledge bases. Our proposed framework, Temporal Information Scoping (TISCO) exploits evidence collected from the Web of Data and the Web. The evidence is combined within a three-step approach which comprises matching, selection and merging. This is the first work employing matching methods that consider both a single fact or a group of facts at a time. We evaluate our approach against a corpus of facts as input and different parameter settings for the underlying algorithms. Our results suggest that we can detect temporal information for facts from DBpedia with an f-measure of up to 80%.

Key words: Temporal Information Extraction, Temporal Semantic Web, Temporal Scoping, Fact Checking

1. Introduction

The Web of Data can be regarded as a dynamic environment where information can change rapidly and cannot be assumed to be static [21]. Changes in the Web of Data sources should reflect changes in the real world [10, 41], otherwise data can soon become outdated. Some facts are not time variant and thus do not change over time e.g. `<CristianoRonaldo, bornIn, Portugal>` while others has a *validity time*[1] with a start and end e.g. `<CristianoRonaldo, playFor, ManchesterUnited>` refers to a fact valid from 2003 to 2009.

Most of knowledge bases store facts under a historical perspective. Facts in this knowledge bases have been true sometime until the current time. We refer to the representation of these facts in the knowledge bases as to *historification of dynamic facts*. Figure 1 shows examples of different teams for the same entity *Jennison Myrie-Williams* (facts extracted from DBpedia 2015-10). Historification of dynamic facts, which is frequently found in many and prominent datasets in Linked Data (LD), fails to provide details about the time interval when the facts have been true. These knowledge bases adopt a *temporal flattening approach* of representing dynamic facts. The incompleteness and the inaccuracy of temporal information in LD [35] is often due to the information extraction process (that can be error prone) or to the representation model (that requires very sophisticated meta-modeling strategies to represent *versioning metadata* in RDF). For instance, in DBpedia it is not possible to know the time interval of the fact `<Jennison Myrie-Williams, playFor, Stevenage>` since all time points are associated directly with the entity rather than the fact (see Figure 1 from f_5-f_{14}) and the semantics of the predicate is the

*Principal corresponding author
[1] The time interval in which the fact is true.

Figure 1: A set of facts F about the entity Jennison Myrie-Williams and a set of time points T occurring with the entity.

	subject	predicate	object
f_1	Jennison Myrie-Williams	playFor	Port Vale F.C.
f_2	Jennison Myrie-Williams	playFor	Stevenage_F.C
f_3	Jennison Myrie-Williams	playFor	Dundee United F.C.
f_4	Jennison Myrie-Williams	playFor	Cheltenham Town F.C.

subject	predicate	object	
Jennison Myrie-Williams	year	1988	f_5
Jennison Myrie-Williams	year	2002	f_6
Jennison Myrie-Williams	year	2006	f_7
Jennison Myrie-Williams	year	2007	f_8
Jennison Myrie-Williams	year	2008	f_9
Jennison Myrie-Williams	year	2009	f_{10}
Jennison Myrie-William	year	2010	f_{11}
Jennison Myrie-Williams	year	2011	f_{12}
Jennison Myrie-Williams	year	2012	f_{13}
Jennison Myrie-Williams	year	2013	f_{14}

same for the starting and the ending time points (e.g. year).

Despite the importance of the relationship between facts and time intervals, very few solutions are proposed. Most of them are concentrated more in extracting facts with time intervals from text [42, 24, 19, 28] rather than trying to map facts with time intervals. The system CoTS provided in [40] is similar to our system since it also detects validity time of facts. In contrast to our approach, CoTS relies on document metadata such as its creation date, to assign time intervals to facts. To the best of our knowledge, this is the first work employing both local and global matching approach, and a system for mapping facts to time intervals.

To map those facts to the correct time intervals, we have to address two main challenges. First, the set of time intervals is created as a combination of all time points extracted from the knowledge base for each entity where the starting time point is smaller than the ending time point. As shown from the example, each fact f_1 to f_4 will have 55 possible time intervals. This set needs to be reduced since it contains also noisy intervals such as all intervals starting with the birth year. Second, to find the correct intervals, we need to extract supporting evidence from external sources that indicates how often a fact occurs with each time point, and subsequently predict the possible time interval of each fact based on the acquired evidence.

In this work, we focus on curating time intervals associated with facts. We introduce an approach for detecting the temporal scope of facts referred to by triples (short: the temporal scope of the triples). Given a fact (i.e., an RDF triple), our approach aims to detect the time points at which the temporal scope of the triple begins and ends. Two sources can be envisaged for gathering such information: the document Web and LD. Our approach is able to take advantage of both: the Web is made use of by extending upon a fact validation approach [26], which allows detecting Web documents which corroborate a triple. In contrast to typical search engines, the system does not just search for textual occurrences of parts of the statement, but tries to find webpages which contain the actual statement phrased in natural language. The second source of information for time scopes is the Web of Data itself. Here, we use DBpedia, for possible time scopes and devise an algorithm for combining the results extracted from Web documents with those fetched from RDF sources. The algorithm consists of three main steps: First, the evidence extracted from Web documents is *matched* against a set of relevant time intervals to obtain a significance score for each interval. Second, a small set of more significant intervals is *selected*. Finally, the selected intervals are *merged*, when possible, by considering their mutual temporal relations. The set of disconnected intervals [1] returned by the algorithm defines the temporal scope of the fact. We also propose two *normalization* strategies that can be applied to the data extracted from Web documents before running the algorithm, to account for the significance of dates appearing in the documents corroborating the input fact.

This article makes the following contributions:

- We present an approach for modelling a space of relevant time intervals for a fact starting from dates extracted from RDF triples.

- We devise a three-phase algorithm for temporal scoping, i.e. for mapping facts to sets of time inter-

vals, which integrates the previous steps via matching, selection and merging.

- We describe two matching methods that consider facts in isolation or cluster them according to the main entity.
- We provide TISCO, a running prototype; the first system able to provide temporally annotated facts which are modelled according to a relationship-centric perspective [35].

This article is an extension of the initial description of work in [37]. The main additions are as follows:

- We describe in detail more alternative solutions, including an additional function in the matching phase of our approach and a normalization function of occurrences of dates. We present experimental results comparing them.
- We developed a prototype for annotating facts with temporal information and show how all the different matching functions and their combinations can be integrated in one framework.

The rest of this paper is structured as follows: We give an overview of the state of the art of relevant scientific areas in Section 2. In Section 3 we define the terminologies and the notations used in this paper. In this section, we provide a general overview and the system infrastructure of our approach. In Section 4, we describe how temporal information is extracted from web pages using a temporal extension of the DeFacto algorithm [26]. Section 5 shows how this information can be mapped to a set of time intervals specifying its temporal scope. Further time intervals are selected according to some criteria and merged when possible in Section 6. We then evaluate the approach by using temporal scopes from Yago2 as gold standard and facts from DBpedia as input in Section 8. Finally, we conclude in Section 9 and give pointers to future work.

2. Related work

The work presented in this paper relies on two areas of research: the extraction of time information and fact checking.

2.1. Extraction of Time Intervals

Most data-driven and Web applications need to manage temporal information in order to capture, model, explore, retrieve, and summarize information changing over time.

2.1.1. Information Extraction on the Web

Several approaches in the field of information extraction have started to consider the notion of time as an important aspect while querying the collections of documents [2, 7]. The extraction of temporal information from free text is difficult to be handled since there is a large diversity of ways in which time can be expressed. In the best case time can be expressed explicitly based on three granularity levels such as year, month and day. In the worst case, temporal information can be expressed either implicitly [36] or through relative temporal expressions [7]. Temporal taggers are used in the process of temporal information extraction by following rule based approaches. Taggers such as TempEx [29], GUTime[2] and HeidelTime[3] have the task of correctly identifying and normalizing temporal information after the text is preprocessed. Although temporal taggers are quite effective for temporal information extraction, they represents different challenges such as they are limited to one language and to one specific domain (e.g. the news domain). In addition temporal taggers may require additional effort for certain applications. Therefore, our approach uses straightforward regular expressions to look for temporal information.

Some approaches consider the detection of links between events and temporal information (e.g., dates) within one or more sentences of a document where the event is mentioned, as a classification problem and thus adopt machine learning techniques (e.g., [42, 24, 19]). Temporal Information Extraction (TIE) [28] is a more recent system that finds a maximal set of temporal annotations for events mentioned in a given sentence. Therewith, it can infer relations between these events using the temporal annotations. Instead of Allen-style intervals [1], TIE uses time points. However, this approach is not sufficient to extract the temporal scope of facts because it focuses on the micro-reading of temporal annotations in single documents or sentences.

Although the aim of temporal bounding [11] and our approach is the same since both retrieve temporal constraints given a fact, there are fundamental differences. NLP techniques employed in temporal bounding are more sophisticated but at the same time more expensive and extract evidence from the text on a limited corpus. Our approach uses softer, but more efficient, NLP techniques to extract evidence from the whole web. In other words, we do not parse and analyze sentences using

[2] http://www.timeml.org/site/tarsqi/modules/gutime/download.html

[3] http://dbs.ifi.uni-heidelberg.de/index.php?id=form-downloads

part-of-speech tagging, which we consider a "deeper" approach to natural language understanding. In addition, our approach investigates how to complement evidence retrieved from texts with evidence from the web of data.

2.1.2. Information Extraction on the Web of Data

Several approaches are provided to represent temporal meta information in RDF [17, 10, 45, 23]. Although there have been several approaches for representing temporal meta information, there is still little understanding about the actual practice of using these representations [35]. As shown in [35], their usage is very limited since the complex mechanisms of modelling temporal meta information makes the querying of temporally annotated facts not easy.

PRAVDA [44] is an Hybrid Acquisition of Temporal Scopes for RDF Data, a proposed method to harvest basic and temporal facts from free text. The approach is based on a semi-supervised label propagation algorithm that determines the similarity between structured facts and textual facts. Yet, it does not use the verbalization of RDF triples to check for RDF triples in text like DeFacto does.

The system CoTS [40] is similar to our system since it also detects temporal scopes for facts. In contrast to our approach, CoTS relies on document meta-data such as its creation date to assign temporal scopes to facts. Yet, it does not use the verbalization of RDF triples to check for RDF triples in text like DeFacto does but it searches in a large text corpus by building query templates for each relation e.g. 'Kennedy-presidentOf-US' is represented by the query 'President Kennedy'. In Section 8, we show an objective comparison of our approach and CoTS.

More recent works [9**?**] provide an enrichment of time intervals in the knowledge bases. The work in [**?**] is very similar since they provide additional temporal scoping of facts and events in a knowledge base. The temporal scoping is extracted from free text based on a rule-based system and is matched against facts. This approach can be considered a complementary work of Temporal DeFacto which has to deal with verbalization and also disambiguation challenges which is out of the scope of this paper. Instead, the work in [9] identifies the correct temporal scoping of facts based on a set of constraints that can be extracted either through rule mining tools or designed by humans. The constraints capture the inconsistencies in the temporally scoped facts.

Knowledge bases publicly available on the web are DBpedia[4] [27], Freebase[5] [6], Wikidata[6] [43] and Yago[7] [39]. These knowledge bases are edited by the crowd such as Freebase and Wikidata or extracted from Wikipedia, a large-scale and semi-structured knowledge base. Other knowledge bases are also created such as NELL [8] by using information extraction methods for unstructured or semi-structured information. But a few of them extract temporal information to indicate the temporal validity of facts. Yago2 and Wikidata provide the most complete list of temporal scopes of facts. Yago2, which we use also in our work, provides temporal scopes by the property `occursSince` and `occursUntil` properties, e.g., the fact `<BillClinton, holdsPoliticianPosition, PresidentOfUnitedStates>` with identifier `f1` and temporal triple given as follows: `<f1, occursSince, 1993-##-##>`. Table 1 shows some statistics about the most occurring temporal properties and the completeness of facts with temporal scopes in Yago2. Wikidata uses qualifiers that indicate the time period during which the fact was true. For example, the fact about Bill Clinton in Wikidata will be represented as `<wd:Q1124, p:P39, wds:statementID>, <wds:statementID, ps:P39, wd:Q11696> <:statementID,pq:P580, "1993-01-20"xsd:date>`, which states that `Bill Clinton` has a political position of `President of the United States of America` under the qualifier property `ps:580`[8] and qualifier value `20 January 1993`. Table 2 shows some statistics in Wikidata about the three temporal properties seen in Yago2. Since the two properties `member of sports team` and `position held` can be used for all subcategories of athletes and human respectively, we added an additional filter. The filter adds the occupation association football player and politician for the two properties respectively. As we may notice from the two tables, the temporal property playFor for soccer players is the most complete.

2.2. Fact Checking

With respect to fact-checking for knowledge bases, a very recent framework has been proposed by Samadi et al. [38]. Similartly to DeFacto, it identifies relevant

[4] `dbpedia.org`
[5] `freebase.com`
[6] `wikidata.org`
[7] `yago-knowledge.org`
[8] indicates the start time

Table 1: Temporal properties and their completeness in Yago2

Temporal Property	% Facts
playFor	63%
holdsPoliticalPosition	14%
isMarriedTo	20%

Table 2: Temporal properties and their completeness in Wikidata

Temporal Property	% Facts
member of sports team	92%
position held	35%
spouse	30%

sources, extracts evidences from them, estimates source credibility and uses those credibility scores for improved claim evaluation. This work, dubbed ClaimEval, is an extension of the author's previous work [30], which also allows to evaluate the truth value of statements by querying the web, but without considering source credibility into account. The main differences among OpenEval, ClaimEval [30, 38] and DeFacto [26] is that the latter is optimised to extract proofs by considering a larger variety of features related to natural language patterns found on websites, besides its multilingual support (*German*, *French* and *English*) [14].

In another line of research on fact checking in [33, 34], trustworthiness is also a central element. The authors rely on a model based on hubs and authorities. This model allows to compute the trustworthiness of facts and websites by generating a k-partite network of pages and facts and propagating trustworthiness information across it. The approach returns a score for the trustworthiness of each fact. An older yet similar approach is that presented in [46]. Here, the idea is to use a 3-partite network of webpages, facts and objects and apply a propagation algorithm to compute weights for facts and webpages.

Other approaches on fact checking are described in more detail in [12]. The authors present a systematic review of the components of DeFacto. These approaches pose several challenges and issues to deal with and this explain the reason why there are not so many approaches proposed in the literature.

3. Problem Definition

In this section, we first give a technical background about the RDF data model (Section 3.1) and then we provide the problem definition of temporal scoping of facts (Section 3.2).

3.1. Background

LD adopts the Resource Description Framework (RDF) [22] for publishing and linking structured data on the Web. The basic idea of the RDF data model is a statement represented by a triple containing three *RDF terms* in the form of subject-predicate-object. An RDF term is an element that can be of three types: Uniform Resource Identifier (URI), literal and blank node. According to the RFC396 a URI is defined as follows:

"A URI is a compact sequence of characters that identifies an abstract or physical resource" – [4].

Example of resources that represent real world and abstract objects may be places, people and images. A literal may represent values such as strings, numbers, or dates. A blank node is neither a URI nor a literal, but is a local identifier that denotes an unnamed resource. Blank nodes are used inside a document that contains an RDF description and cannot be referenced outside of their originating scope.

The use of blank nodes is discouraged for LD [5] although in practice LD extensively adopt them. However, in our approach we do not consider blank nodes similarly to the work in [9]. We represent a fact by a triple with subjects being URIs and objects being either URIs or literals, (e.g., `<Cristiano Ronaldo, team, Manchester United>` represents a fact asserting that the relationship indicated by the `team` predicate holds between the entity `Cristiano Ronaldo` and `Manchester United`, denoting the subject and the object respectively). In other words facts describe real world *entities*. We use the term *entities*[9] as a short form for *named individuals* as defined in the OWL 2 specification [31].

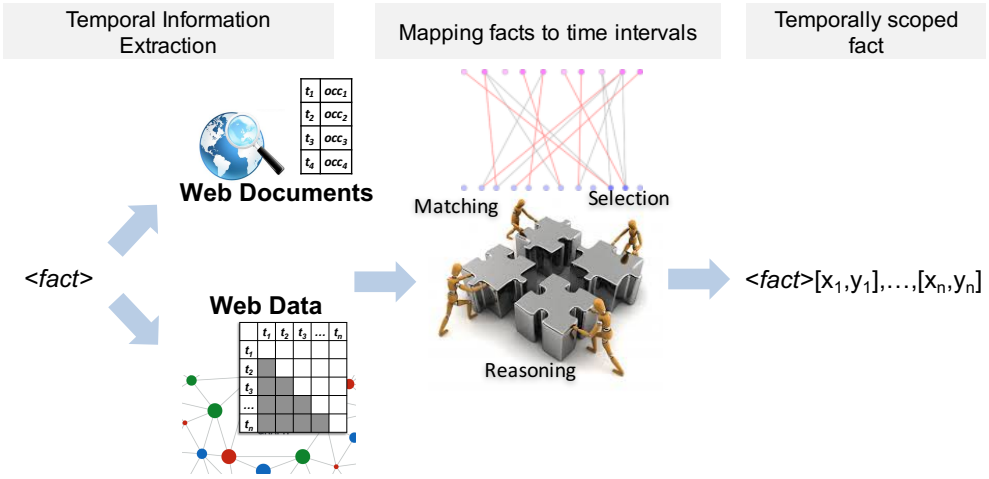
According to the third LD principle [5], we assume that each URI identifying an entity is dereferenceable. The dereferenceability of the URI entities relies on the HTTP mechanism, known as *content negotiation* [13]. This mechanism dereferences the URI that identifies the entity and returns the description of the entity in a specified data format and language indicated by a user agent. For short we call a *description of an entity*, an entity document and we formally define it as follows:

Definition 1 (Entity document)**.** *An entity document denoted as d is the description of the entity e which is returned after looking up the HTTP URI of e.*

Entity documents can be represented in the form of HTML pages when read by humans. Entity documents that are intended to be read by machines are represented

[9]Entities are represented by URIs.

Figure 2: Approach Overview



as RDF documents in order to enable different applications to process the standardized content.

Additionally to the RDF construct called *reification* [18] there exist other approaches used to represent temporal information in RDF such as n-ary relationship [35, 3]. Time information can be of two types: *time points* and *time intervals*. Time points can be represented as typed literal such as `xsd:date`, `xsd:dateTime` or even an `xsd:integer`[10]. It is possible to distinguish *temporal triples* which refer to triples of the form `<s,a,t>` where the object t represents a time point and a a *temporal annotation property* for short *temporal property*. We call *temporal property* every property used in a temporal triple such as `dateOfBirth`, `createdOn`, `start` or `end`. An example of temporal triples can be shown in Figure 1 from f_5 to f_{14}.

3.2. Temporal Scoping Problem

A vast amount of facts in knowledge bases are considered to be dynamic and are valid only in an interval of time. Providing facts with temporal information is relevant for many application domains. Several knowledge bases contain dynamic facts without explicitly annotating triples with a temporal scope that we define as follows:

Definition 2 (Temporal scope)**.** *Given a fact f and a time interval $[t_i, t_j]$ delimited by a starting time point t_i and an ending time point t_j, a temporal scope denoted as $\langle f, [t_i, t_j]\rangle$ is the specification of the time interval when the fact is true.*

[10]It might be interpreted as a year

In this paper we regard time as a discrete, linearly ordered finite domain, as proposed in [17]. A time interval denoted as $[t_i, t_j]$ satisfies the condition that $t_i <= t_j$. We adopt the same representation of a time point as of a time interval where the starting time point t_i is equal to the ending time point t_j. In our discrete time model, two intervals $[t_i, t_j]$ and $[t_h, t_k]$ are *disconnected* iff $t_j < t_h$, or $t_k < t_i$, and *connected* otherwise.

Temporal triples available in the knowledge base may be considered *relevant* to define the *temporal scope*. We consider a set of dynamic facts $F = \{f_1, f_2, ..., f_m\}$ where m is the number of facts and s_k is the subject of a generic fact f_k; and a set of time points $T = \{t_1, t_2, ...t_n\}$ extracted from n temporal triples of the form $\langle s_k, a, t_i\rangle$ where t_i is a generic time point. The number of time points is greater or equal to the number of facts ($n >= m$). The problem addressed in this paper is to find a set of temporal scopes denoted by TS where each time interval has a starting and ending time point from T. The task of assigning the correct set of time intervals to a fact is known as temporal scoping that we formally define as follows:

Definition 3 (Temporal scoping)**.** *Let $f \in F$ be a fact and $TS = \{[t_{i_1}, t_{j_1}], ..., [t_{i_n}, t_{j_n}]\}$ a set of temporal scopes. Temporal scoping problem finds a mapping between f and TS.*

4. Temporal Evidence Extraction

An overview of our approach is given in Figure 2. The temporal evidence for a given fact f is extracted from unstructured documents (see Section 4.1) and from RDF documents (see Section 4.2) where a space

of possible time intervals relevant to the fact is built; the evidence extracted from unstructured documents is matched against the space of relevant time intervals and after a selection and merging function is applied, the final set of temporal scopes are associated with the input fact.

4.1. Temporal Evidence Extraction from Unstructured Data

To extract evidence from unstructured data, we provide an RDF triple as input, which is then mapped to natural-language patterns. With this pattern we are able to search the corpus of indexed documents. Finally we return a distribution of all dates and their number of occurrences for a given fact. Hence, the output of the temporal evidence extraction from unstructured data for a fact f can be regarded as a vector UEV over all possible time points t_i whose i^{th} entry is the number of co-occurrences of s or o with t_i. We will use the function $\phi_i(f, t_i)$ to denote the value of $UEV[i]$ for the fact f. Figure 3 shows a set of facts and their correspondent UEV.

Figure 3: A set of facts and the UEV for each fact.

Web Documents

	subject	predicate	object	years	Occ
f_1	Jennison Myrie-Williams	team	Port Vale F.C.	*2007*	*1*
	Jennison Myrie-Williams	team	Port Vale F.C.	*2011*	*6*
	Jennison Myrie-Williams	team	Port Vale F.C.	*2012*	*5*
	Jennison Myrie-Williams	team	Port Vale F.C.	*2013*	*10*
f_2	Jennison Myrie-Williams	team	Stevenage_ F.C	*2011*	*12*
	Jennison Myrie-Williams	team	Stevenage_ F.C	*2012*	*10*
f_3	Jennison Myrie-Williams	team	Dundee United F.C.	*2009*	*2*
	Jennison Myrie-Williams	team	Dundee United F.C.	*2010*	*310*
f_4	Jennison Myrie-Williams	team	Cheltenham Town F.C.	*2007*	*1*
	Jennison Myrie-Williams	team	Cheltenham Town F.C.	*2008*	*2*

4.1.1. Temporal Evidence Extraction with DeFacto

Temporal DeFacto [14] is an extension to the DeFacto framework presented in [26]. The system takes an RDF triple as input and returns a confidence value for this triple as well as a possible evidence for the fact. The evidence consists of a set of webpages, textual excerpts from those pages and meta-information on the pages. The main steps of temporal DeFacto are given as follows:

Step 1 The first task of DeFacto is to retrieve webpages which are relevant for the given task. The retrieval is carried out by issuing several queries to a search engine. These queries are computed by verbalizing the RDF triple using natural-language patterns extracted by the BOA framework [15]. The highest ranked webpages for each query are retrieved, which are candidates for being sources for the input fact. Both the search engine queries as well as the retrieval of webpages are executed in parallel to keep the response time for users within a reasonable limit.

Step 2 Once a webpage has been retrieved, we extract plain text by removing HTML markup and apply our fact confirmation approach on this text. In essence, the algorithm decides whether the web page contains natural language formulations of the input fact. If no webpage confirms a fact according to DeFacto, then the system falls back on lightweight NLP techniques and computes whether the webpage does at least provide useful evidence.

Step 3 To also incorporate time information into the fact validation process we extended DeFacto as follows. On all retrieved webpages we apply the Stanford Named Entity Tagger[11] and extract all entities of the *Date* class. We then examine all occurrences $occ_{so} \in Occ_{so}$ of the subject and object label of the input fact (or their surface forms, e.g. "Manchester United F.C." might also be called "ManU") in a proximity of less than 20 tokens.

Step 4 In addition to fact confirmation, the system computes different indicators for the trustworthiness of a webpage as presented in [32]. These indicators are of central importance because a single trustworthy webpage confirming a fact may be a more useful source than several webpages with low trustworthiness.

Step 5 In addition to finding and displaying useful sources, DeFacto also outputs a general confidence value for the input fact. This confidence value ranges between [0, 1] and serves as an indicator for the user: Higher values indicate that the

[11] http://www-nlp.stanford.edu/software/CRF-NER.shtml

found sources appear to confirm the fact and can be trusted. Low values mean that not much evidence for the fact could be found on the Web and that the websites that do confirm the fact (if such exist) only display low trustworthiness.

Step 6 The generated provenance output can also be saved directly as RDF and abides by the PROV Ontology[12]. The source code of the DeFacto algorithms and DeFacto's user interface are open-source[13].

4.1.2. Normalizing Occurrences of Time Points

Two types of normalization functions can be envisaged: *local normalization* and *global normalization*. These functions aim to transform the output vector of the evidence extracted from unstructured data into a probabilistic time distribution vector. Each approach describes a frequency-based interpretation of the UEV output, which takes a year as input and returns a number representing its probability.

Local Normalization. The local normalization denoted by $norm_{local}$ of a time point t_i and a fact f is the probability (i.e., the relative frequency) of the co-occurrences of t_i with f. Assuming that ϕ is a function taking f and t_i as input and returning the co-occurrence of t_i with f, we can define the local normalization function as follows:

$$norm_{local}(x) = \frac{\phi(f,x)}{\sum_{y \in |T|} \phi(f,y)} \quad (1)$$

This function divides the occurrence of a time point by the sum of all occurrences of time points in T of a given fact. The main drawback of such a normalization is that it does not take into consideration a global view of facts sharing the same subject s.

Global Normalization. An alternative approach to compute the time distribution vector for a given fact, is to adopt a global view of time points for all facts sharing the same subject s.

Probability distribution. The global normalization denoted by $norm_{glob}$ of a time point t_i for a given fact f is the probability of the co-occurrence of t_i over all objects of s. The global normalization function can be computed as follows:

$$norm_{glob}(x) = \frac{\phi(f,x)}{\sum_{y \in |FS|} \phi(y,x)} \quad (2)$$

This function divides the ϕ of a time point x by the sum of all ϕ entries of x distributed over all objects of facts in FS that have the same subject s with $FS \subset F$. For instance, if a player associated with different teams for a particular time period, the global normalization balances the occurrence of the time point on all the teams.

Term frequency−inverse document frequency normalization. We propose another normalization function, inspired by the tfidf. We introduce the time point frequency and inverse fact frequency. This normalization combines time points frequency (calculated similarly to term frequency) to the global specificity of the time point (calculated similarly to inverse document frequency).

- Time point frequency (tf) is the number of occurrences of a time point divided by the sum of occurrences of time points in T for a given fact f. It operates at the level of the single fact f (local level) and it corresponds to the $norm_{local}$ function defined before.
- Inverse fact frequency (if) considers how many times a time point is used in some facts w.r.t. the total number of facts. It will penalize time points that occur more frequently in and will reward others that do not occur so frequently, thus considering the specificity of a time point. It operates at the level of objects and time points (global level).

In particular, if a player is associated globally only to a team for a particular date (score > 0) it means that this time point is very specific. Otherwise, if the player is associated to several teams for a particular time point, the tfif penalizes the time point by decreasing the total score.

$$norm_{tfif}(x) = norm_{local}(x) * log(1+ \frac{|FS|}{|\{y \in FS : \langle y,x \rangle\}|}) \quad (3)$$

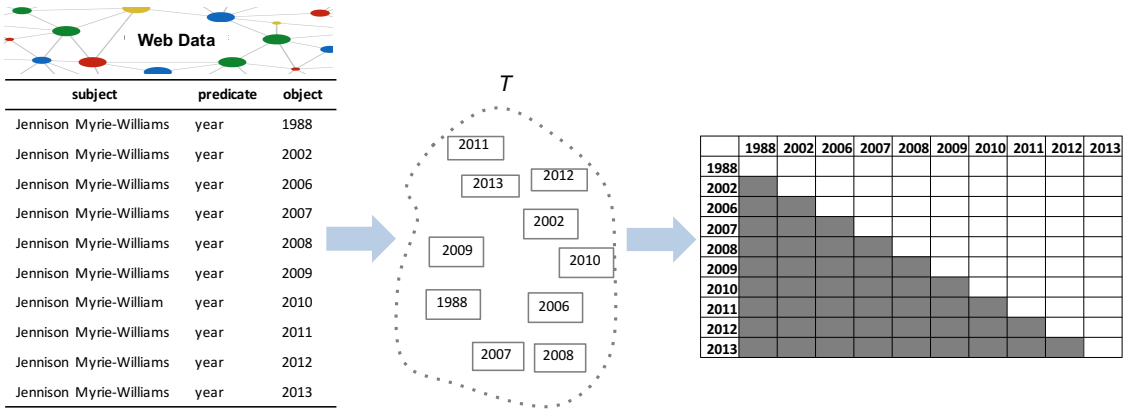
where $|FS|$ is the total number of facts having the same subject and predicate and $|\{y \in FS : \langle y,x \rangle\}|$ is the number of facts associated with a time point x which have non-zero values.

4.2. Temporal Information Evidence from Web of Data

Given a set of facts F to map to time intervals, we first identify the set of entities that occur as subjects. Given the subject s of the fact, we use the HTTP content negotiation mechanism to retrieve the entity document d. As an example, given

[12] http://www.w3.org/2011/prov/
[13] https://github.com/AKSW/DeFacto

Figure 4: RIM of the entity Jennison Myrie Williams with relevant time intervals.



the fact ⟨ `Jennison Myrie-Williams, team, Port Vale F.C.`⟩, we extract the RDF document describing `Jennison Myrie-Williams`. Once an entity document has been retrieved, we identify temporal triples[14] in the entity document d and extract dates by using regular expressions, which identify standard date formats and variations. In this step we adopt an approach that was used in previous work [35]. We add to this set of extracted dates a date representing the *current time*. As a result of this step, each fact $f \in F$ is associated with a set of *time points* T extracted from the RDF document describing the subject of the fact. In principle, our approach can consider dates represented at any granularity level; in the following examples and in the experiments, time is represented at the year level similarly as in other related work [28, 40].

Intuitively, we want to use the time points T associated with a subject s to identify a set of most *relevant time intervals* for scoping facts with s as subject; in this way, we can reduce the space of all possible time intervals considered for an individual fact. The set of time intervals relevant to a subject s is defined as the set of all time intervals whose starting and ending points are members of T. Relevant time intervals are represented using an upper triangular matrix.

Given a set of relevant time points T for a subject s, a relevant time interval matrix (*Relevant Interval Matrix* for short) RIM is an upper triangular matrix of size $|T| \times |T|$ defined as follows:

$$RIM = \begin{bmatrix} rim_{t_1,t_1} & \cdots & \cdots & rim_{t_1,t_n} \\ 0 & \ddots & & \\ 0 & 0 & \ddots & \\ 0 & 0 & 0 & rim_{t_n,t_n} \end{bmatrix} \quad (4)$$

Columns and rows of a relevant interval matrix RIM for a subject s are indexed by ordered time points in T; each cell rim_{t_i,t_j} with $i, j > 0$ represents the time interval $[t_i, t_j]$, where $t_i, t_j \in T$. At the moment we assign a placeholder value `null` to each cell $rim_{i,j}$ such that $i \leq j$. In the matching phase, we will use entity-level $RIMs$ (i.e., a RIM for each entity) as schemes for fact-level matrices; in these fact-level matrices `null` values will be replaced by scores that represent the significance of intervals for individual facts. Observe that the use of an upper triangular matrix is suitable for representing time intervals since the time intervals represented in the cells in the lower part of the matrix ($i > j$) are not valid by definition. Also note that, the cells in the diagonal of the RIM matrix represent time intervals whose start and end points coincide.

Figure 4 shows a RIM that has in columns and rows a set of ordered relevant time points extracted from the RDF document of the entity *Jennison Myrie Williams*. As it is shown by the RIM, the entries below the diagonal are not considered because they do not satisfy the condition that the starting time point is less than the ending time point.

An alternative way of building the RIM is to use an arithmetic progression of time points such that the difference between two consecutive time points is one. We select the smallest and the biggest time point from T and then we create the sequence of time points as an arithmetic progression.

[14] We use temporal triples available in the knowledge base under the assumption that this information can be *relevant* to define the scope of facts.

5. Matching Methods

The matching phase is based on the determination of a family of matching functions that we describe in the following sections. It is clear that in the general case the problem we are trying to solve requires a matching $*:*$ since a fact can be associated with several time intervals and vice versa. In our previous work [37] we used a local approach that we reassume in Section 5.1. However, it is also possible to apply a global matching approach as described in Section 5.2. All functions compute values between 0 and 1.

5.1. Local Matching Function

The inputs of the matching phase for a fact f with a subject s: are the following: a relevant interval matrix RIM extracted the entity document d and a time distribution vector UEV. Probabilistic time distribution vectors obtained by normalization (see Section 4.1.2) can be also used as input instead of the UEV. The matching phase returns an *interval-to-fact significance matrix* (Significance Matrix (SM) for short) associated with the fact f. An SM is a triangular square matrix having the same size and structure of the input RIM. As a next step, `null` values of a RIM are replaced with significance scores returned by a matching function.

In practice, to build an SM of a fact f with subject s, we match a fact-level UEV associated to the fact f against an entity-level RIM, i.e. the matching aims to inject a time distribution vector into RIM by producing a significance matrix SM. The matching function $match(UEV, RIM) = SM$, where s is a subject and f is a fact, is given as follows:

$$sm_{i,j} = \begin{cases} 0 & \text{if } rim_{i,j} = 0 \\ \frac{\sum_{k=i}^{j} \phi(f,k)}{(j-i)+1} & \text{if } rim_{i,j} = null \wedge i < j \\ \phi(f,i) * w_{i,j} & \text{if } rim_{i,j} = null \wedge i = j \end{cases} \quad (5)$$

Since the denominator $(j - i) + 1$ in the formula used in case two represents the number of time points included in the interval $[i, j]$, the formula is equal to the average of UEVs for the time points contained in the interval. As an example, the score for a cell $sm_{1995,2000}$ is defined as the average value of UEV for the time points between 1995 and 2000 (including the starting and ending points). Since the elements in the diagonal have length equal to 1, the formula used in case three is equivalent to multiplying the score computed with the formula used in case two for a weight $w_{i,j}$; we use this weight to penalize the scores in the diagonal as we discovered that formula in case two would assign high scores to the element in the diagonal, thus favoring time intervals with length equal to 1 in the selection phase. Intuitively we want to penalize elements in the diagonal unless they are the only significant values selectable in the SM matrix. The weight is defined as inversely proportional to the difference between the length of the considered interval (equal to 1) and the length of the UEV vector ($length(UEV)$) as follows:

$$w_{i,j} = \frac{1}{c * length(UEV)} \quad (6)$$

where c is a constant used to control the score reduction ratio applied to the elements in the diagonal of the SM matrices. To learn the constant we applied a cross-validation like approach by splitting the dataset such that one part was used for the training and the other part for testing the value of c. Figure 5 shows an example of the local matching function applied on each RIM for the subject `Jennison Myrie-Williams`.

5.2. Global Matching Function

In Section 5.1, we described a matching function that consider a single fact at a time, the following approach proposes a matching function that operates globally and considers more than one fact for the same subject. We extend our approach by adopting the matching problem on bipartite graphs [16]. A bipartite graph $G = (V, E)$ is a graph in which the vertex set V can be divided into two disjoint subsets such that no two graph vertices within the same set are adjacent.

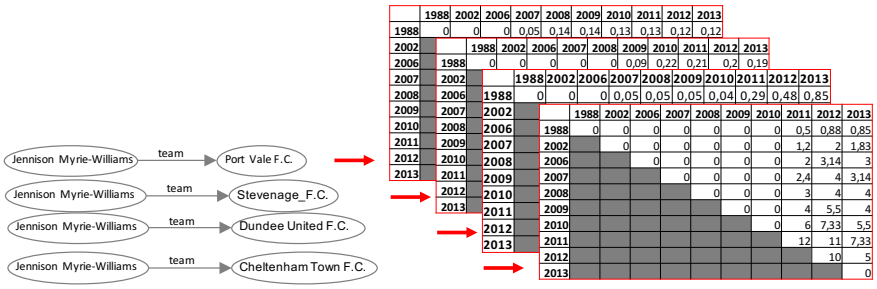
A possible solution of the matching problem on bipartite graphs is to represent this problem as an instance of a linear programming problem, in particular the Maximum Weight Bipartite Matching (MWBM). We define the matching problem of MWBM as follows:

Definition 4 (Maximum Weight Bipartite Matching)**.** *Given a weight $w_{ij} \in R$ for all $(i, j) \in E$, find a maximum weight of matching where the weight of matching M is given by $w(M) = \sum_{(i,j)\in M} w_{ij}$.*

A matching $M \subseteq E$ is a collection of edges such that every vertex of V is incident to at most one edge of M. If a vertex v has no edge of M incident to it then v is said to be *exposed* (or unmatched). A matching is *perfect* if no vertex is exposed; in other words, a matching is perfect if the cardinality of M is equal to the cardinality of the two disjoint subsets of V.

An instance of the MWBM problem can be described in our approach as follows: Given a set of facts $f \in FS$ where $FS \subset F$, we collect all SMs having the same

Figure 5: Local matching function.



subject. Thereafter, we create two sets A and B where A comprises all objects o of $f \in FS$ and B comprises the set T of all relevant time points for the given subject s. For each edge (i, j) of the bipartite graph there is a weight provided by SMs as shown in Table 3.

The MWBM proposes a matching with cardinality `1 to 1` while our problem of matching has cardinality `N to N`. A solution to this is to create an analogy with the well known problems of College Admission or Hospital Residents problem which uses a mapping with cardinality `1 to N`. Analogously, a matching is a mapping of the time intervals (resident) to the set of facts (hospitals). After that, we can transform the mapping with cardinality *1 to N* into a classical matching problem with cardinality `1 to 1` by replacing a fact f by N facts $f_1, f_2, \ldots f_N$. Table 4 shows that first we exchange objects with time points and the objects are replicated n-times with n the number of time points. Once the weight is assigned to each edge then the aim is to find the maximal matching based upon linear programming. We consider the linear programming in order to prove optimality of the matching. For this purpose, one would like to find upper bounds on the size of any matching and hope that the smallest of these upper bounds be equal to the size of the largest matching. This is a *duality* concept that will itself be a combinatorial optimization problem. The solution of a dual problem in this case would be a *minmax* algorithm which gives a maximum matching and a minimum vertex cover (for more details see [25]).

In Table 5 we show the results returned by our algorithm which returns the matching that maximize the sum of values. The final results returned is:

- Cheltenham → { 2007,2008 }
- Dundee United → { 2009 }
- Stevenage → { 2010, 2011 }
- Port Vale → { 2012 }

To notice that we do not use the time intervals but the time points for two reasons: (i) the matching algorithms consider the independence between the sets, (ii) there do not exist matching algorithms that can be applied to the poset of the set of all subsets of the elements in T.

We expect the global matching approach to work better on functional properties. However, since we use time points at year granularity, previously functional properties might become non-functional (e.g., a soccer player changing team in mid year). In our case, we conduct experiments (see Section 8) with properties that are *fairly* functional but where functionality is not always guaranteed: isMarriedTo (fairly functional), playFor (very close to be functional, with exception being players playing for national teams and loans and also if someone change team in the mid year), holdsPoliticalPosition (fairly functional).

6. Selection and Reasoning

6.1. Selection Function

Once we have a set of significance matrices $SM_1, ..., SM_n$, each one associated with a fact f having subject s, we then select the time intervals that might be mapped to the considered facts. We propose two basic selection functions that use SMs; both functions can select more than one interval to associate with a fact f. The *neighbor-x* selects a set of intervals whose significance score is close to the maximum significance score in the SM matrix, up to a certain threshold. In other terms, we define the *neighborhood of the time interval with maximum significance score* as the set of intervals whose significance scores fall in the range defined by the maximum score as upper bound and by a threshold based on a parameter x as lower bound. The threshold is linearly proportional to the maximum significance score, i.e. the threshold is higher when the maximum significance is higher and vice-versa. The threshold is defined to be proportional to the maximum score in such a way

Table 3: An example of the initial instance.

Objects/ Years	1988	2002	2006	2007	2008	2009	2010	2011	2012	2013
Cheltenham	0	0	0	0.017	0.033	0	0	0	0	0
Dundee United	0	0	0	0	0.033	0.033	0	0	0	0
Port Vale	0	0	0	0.005	0	0	0	0.003	0.003	0
Stevenage	0	0	0	0	0	0	0.022	0.022	0	0

Table 4: An example of the transformed instance.

Years/ Objects	Cheltenham	...	Dundee United	...	Port Vale	...	Stevenage	...
1988	0	0	0	0	0	0	0	0
2002	0	0	0	0	0	0	0	0
2006	0	0	0	0	0	0	0	0
2007	0.017	0.017	0	0	0.005	0.005	0	0
2008	0.033	0.033	0.033	0.033	0	0	0	0
2009	0	0	0.033	0.033	0	0	0	0
2010	0	0	0	0	0	0	0.022	0.022
2011	0	0	0	0	0.003	0.003	0.022	0.022
2012	0	0	0	0	0.003	0.003	0	0
2013	0	0	0	0	0	0	0	0

Table 5: An example of the output of the MWBM algorithm.

Years	Objects	score
2007	Cheltenham	0.017
2008	Cheltenham	0.033
2009	Dundee United	0.033
2010	Stevenage	0.022
2011	Stevenage	0.022
2012	Port Vale	0.003

that the similarity range from which we select intervals is higher when the upper bound of the range, i.e., the maximum similarity score, is higher. For example, if maximum score is 0.033, for x = 0.8, we select all time intervals with similarity score higher than 0.0264, with a range equal to 0.0066; if maximum score is 0.066, we select all time intervals with similarity score higher than 0.0528, with a range equal to 0.0132. The range is higher when the maximum score is higher because also absolute values that fall into the range are, in absolute terms, higher and, thus, deemed to be more valuable. The parametric function *neighbor-x* with an SM and a parameter x given as input is defined as follows:

$$neighbor(SM, x) = \{[i, j] \mid sm_{i,j} \geq maxScore - x * maxScore\} \quad (7)$$

The two basic functions *top-k* and *neighbor-x* can be combined into a function *neighbor-k-x* that selects the top-k intervals in the neighborhood of the interval with higher significance score. Observe that neighbor-0 is equal to top-1 for every value of the parameter x. The neighbor-k function behaves as a filter on the results of the top-k function, by selecting only intervals whose significance is close enough to the most significant interval.

6.2. Aggregation of Time Intervals

Finally, we use rules based on Allen's interval algebra to merge the selected time intervals and map each fact to a set of disconnected intervals. Let a and b be two time intervals associated with a fact f and defined respectively by $[t_i, t_j]$ and $[t_h, t_k]$; we merge a and b into

an interval defined by $[min(t_i, t_h), max(t_j, t_k)]$ whenever one of the following conditions, each one based on Allen's algebra relations [1], is verified:

- a overlap b or a is-overlapped-by b
- a meets b or a is-met-by b
- a during b or b during a
- a starts b or b starts a
- a finishes b or b finishes a

The temporal scope of a fact is defined by the set of disconnected time intervals mapped to it after the interval merging phase.

7. Web Interface

TISCO[15], is a prototype that supports experts of the matching problems to test their algorithms in a straightforward way. This system already implements the matching functions described in Section 5 and also the selection functions described in Section 6. The TISCO features include its extensible architecture that facilitates the integration of a variety of matching functions, its capability to evaluate and compare matching results, and its user interface with a control panel that drives all the matching methods and other configuration parameters.

The architecture of TISCO is shown in Figure 6 and it comprises the following components:

- The *DispatcherServlet* has the role of Front Controller, which provides a centralized entry point for handling requests [4]. It deals to map service URLs exposed with corresponding controller methods.
- The *Controllers* carry out the checks and calculations once the request has been forwarded to the responsible controller. Thereafter the responsible controller recalls and forward the response to the service layer methods needed.
- The *Services* have the role of managing services that have a business functionality. They also provide a logical grouping related functionality to certain aspects of the application.
- The *Core Library* contains and exposes the classes of the main algorithms and data structures used by the entire system.
- The *Spring data module* for MongoDB has the task of managing and facilitating the interactions with the database.

To allow easier use and immediate service we developed a front end interface. There are two features of the interface: the creation/management of a matching process through a control panel and the querying of temporally annotated results in the database.

8. Evaluation

This section describes the evaluation of our approach. The aim of the experiment is to show i) the correctness of our approach by comparing different configurations of normalization, matching and selection functions and ii) the efficiency and scalability of our approach.

8.1. Experiment settings

We test our approach by considering the DBpedia dataset[16].

Gold Standard. We choose data from Yago2 since significant parts of DBpedia and Yago2 are extracted from the same source (i.e., Wikipedia), it is possible to easily map facts from DBpedia to facts in Yago2. It is one of the few large open-source knowledge bases that provides temporal annotations for a significant number of facts (714,925 time points associated with facts). Facts in Yago2 are identified through an id which helps to match them against temporal meta information, represented by time points, that are stored in a separate file. To evaluate our approach we extract 14 properties of Yago. We choose the 3 most occurring properties which can be easily framed in timespans that are precise to the year. We extracted facts for the three relations as shown in Table 7. The three relations represents three different domains: sport, politics and celebrities[17].

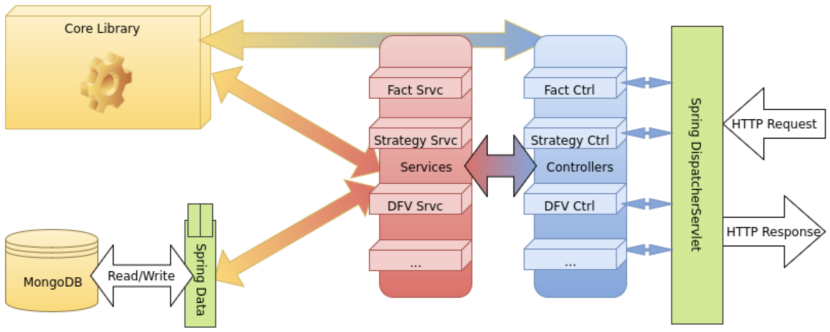
In order to create time intervals we had to identify for the same fact its starting and ending time point that are represented by the `occursSince` and `occursUntil` relation respectively. We had to manually curate the data by solving incorrectly matched facts and time points. For instance, a fact identified by a unique id such as (`X playFor Y`) may happen more than once during the career of a soccer

[15] `http://tisco.disco.unimib.it/temporal-interval-scoping/`

[16] As we showed in our previous work [37], Freebase is incomplete and as such the precision and recall results drop significantly. DBpedia is improved in the latest release and thus we consider the latest version (2015-10) which might be more accurate. Freebase is not maintained since 2014.

[17] The used dataset can be found in `https://goo.gl/ioeWM4`

Figure 6: The system architecture.



player. In such situation, the fact may be associated with the wrong time interval where the starting date may not correspond to the ending date identified automatically. Two authors of the paper (the first and the third) had to do the correction by completing the missing facts. We found 5% to be wrong match on time intervals for the property *playFor*, *marriedTo* and `holdsPoliticalPostion`. Around 2% of date values were 0 for the property `marriedTo`.

External Evidence. Regarding the external evidence we used two data sources:

- Google Books Ngram[18] is a corpus of scanned books. The resulting corpus contains n-grams of length 1 to 5 and a year which indicate the period of time the n-gram was found and three different counts for each n-gram which indicate: the occurrence of the n-gram, the number of distinct books having the n-gram and the number of pages that contain the n-gram always referring to a single year. This corpus contains data with a time coverage spanning from 1500 to 2008. We use the same experimental dataset as proposed in [40]. Therefore, we choose three US Administration office properties as shown in Table 6. For Google Books Ngrams dataset, we index only the English 5-grams published during the period 1960 - 2008 and we search each fact of our dataset in the n-gram datasets by first mapping the fact to the query template that has the following structure 'Office LastName' e.g., "president clinton". To note that the version of the dataset is not specified in the paper so we consider version 2 that corresponds to the id number 20120701.
- Temporal DeFacto. Regarding the evidence extracted by temporal DeFacto, we perform the experiments on a subset of facts. The limit number of queries that can be sent through temporal DeFacto is imposed by traffic limitations of its underlying search engine. Therefore, we apply some selection rules as follows: the top 1043 facts on the most important soccer players who are born after 1983 ($\leq$30 years old), the top 1000[19] facts on politicians born after 1940, and the top 1000[20] facts on celebrities born after 1930.

Table 6: Properties of interest and the number of facts for each property

Property	Selec. Facts
president	9
vice president	12
secretary	27

Table 7: Properties of interest and the number of facts for each property

Property	Entities	Selec. Facts	Total Facts
<playFor>	180	1043	315,486
<holdsPoliticalPosition>	621	719	5,610
<ismarriedTo>	710	876	5,582

Measure In order to evaluate the accuracy of our method, we measured the degree to which the temporal scope we retrieved is correct w.r.t. the gold standard. Therefore, for each fact, we considered the degree

[18] `http://storage.googleapis.com/books/ngrams/books/datasetsv2.html`

[19] But only 719 of them have temporal triples.

[20] But only 876 of them have temporal triples.

of overlap between the retrieved intervals and the intervals in the gold standard. This degree of overlap can be computed by adapting the well-known metrics of precision, recall and F_1-measure to this problem leveraging the discrete time model. Intuitively, the precision of a temporal scope can be measured by the number of time points in the temporal scope generated by our solution that fall into the time interval in the gold standard. The recall of our solution can be measured by the number of time points in the gold standard that are covered by the temporal scope.

Let $R(f)$ be the set of time points in the temporal scopes[21] retrieved for a fact f and Ref(f) be the set of time points included in the reference temporal scopes for f; the following formulas capture the intuitions described above:

$$\textit{precision}(f) = \frac{|R(f) \cap \textit{Ref}(f)|}{|R(f)|}. \quad (8)$$

$$\textit{recall}(f) = \frac{|R(f) \cap \textit{Ref}(f)|}{|\textit{Ref}(f)|}. \quad (9)$$

Precision and recall for a fact f can be combined as usual in F_1-measure defined as the harmonic mean of precision and recall. Note that: when $precision(f) = 1$, each interval in the retrieved temporal scope is included in the interval of the gold standard; when $recall(f) = 1$, all the time points in the interval of the gold standard are covered by the retrieved temporal scopes; when $F_1(f) = 1$ the temporal scope contains exactly the same time points as the gold standard.

Baseline Given that no prior algorithm aims to tackle exactly the task at hand, we computed the precision, recall and F-measure that a random approach would achieve. To this end, we assumed that given the restrictions we set on the intervals within which our solutions must lie (e.g., 1983-2014 for soccer players), a random solution would simply guess for each date whether it should be part of the final solution. This serves as a lower bound for the score a temporal scoping algorithm should achieve.

Configurations For all facts we applied different configurations in the experimental settings:

Matchers

- `loc-no-diag-penal`: local matching function without the penalization on the diagonal,
- `loc-diag-penal`: local matching function with diagonal penalization,
- `glob-disj-intervals`: global matching function without overlapping of time intervals and
- `glob-conti-intervals`: global matching function with time interval overlapping.

Selections

- `topk2`: *top-k* function with k=2,
- `proxy3`: *neighbor-x* function with x=3 and
- `topk2proxy10`: *neighbor-k-x* with k=2 and x=10

Normalizations

- `nNorm`: no normalization applied,
- $\texttt{norm}_{local}$: local normalization applied,
- $\texttt{norm}_{glob}$: global normalization applied,
- $\texttt{norm}_{tfif}$: term frequency - inverse document frequency.

8.2. Comparison on US Administration properties

In this section we show a comparison of our approach against CoTS [40]. First, we extract all the facts from DBpedia having the template: `<USpolitician, office, USadministrationOffice>` e.g., `<dbo:Bill_Clinton, dbo:office, dbo:President_of_the_United_States>`. For each type of the US politician (see Table 6) we extract the relevant time points, from their entity documents, to create the RIM matrix. Then we use the evidence from the ngram corpus to inject data in the RIM matrix. The measures of our work (*DBpedia-ngram*) are slightly different from the traditional precision and recall used in the CoTS approach [40]. Therefore, the metrics in CoTS metrics capture if the fact is true or false at a given time interval. We observe that our approach significantly improve temporal scoping performance as shown in Table 8[22]. The main reasons of this improvement is due to the use of the RIM matrix build with relevant time points extracted from DBpedia. If we use another RIM matrix build with relevant time points extracted from ngram corpus

[21] The reasoning on temporal scopes is already applied.

[22] As we can notice, the total number of facts is different. This is due to the fact that a lot of secretary position in the English version of the 5-grams from where we extract the evidence was missing.

and the evidence injected from the same ngram corpus (*ngram-ngram*), the result gets worse. The main reason is that the sequence of dates creating the RIM and dates used for the evidence will have the same characteristic, i.e., the distance of successive members will be the same in both cases and the matching function will be equal to the occurrences not normalized[23].

Table 8: Properties of interest and the number of facts for each property

Approach	Selec. Facts	F1
CoTS	48	63.63
DBpedia-ngram	31	**79.51**
ngram-ngram	31	44.85

Based on these results, in all subsequent experiments, we analyse in detail the components of our approach.

8.3. Results on DBpedia properties

In this section, we show the experiments on the properties of Table 7 evaluated with our metrics.

`playFor` Property Figure 7 compares our approach against the baseline. Our approach obtains higher precision above 0.6 and higher recall above 0.7 and it improves F-measure over baseline methods by 30-49%. We next give a detailed comparison of the `playFor` property being a non functional property. It means that there is an overlapping between the time intervals when a soccer player plays for a club and the national team. Another case of overlapping is when a player can play for a club and can be borrowed by another club. We show next the best results considering different configurations.

Comparison of matchers Figure 7 compares the results of applying *local matcher* with/without diagonal penalization and *global matcher* with/without overlap. We observe that while precision (above 0.75) is better for global matcher with disjoint intervals, the recall (above 0.85) is better for global matcher with continuous intervals. Although these two matching functions lead to very high results on precision and recall, the F-measure is higher for local matcher with diagonal penalization (above 0.65). The best result is achieved after applying the configuration with selection function neighbor-10 and normalization function is global.

Since it is not possible to understand all the details of the results through an average metric, we group the results of precision, recall and F-measure into four categories defined as: minor than 0.25, from 0.25 to 0.50, from 0.50 to 0.75 and greater than 0.75. Table 9 shows the distribution of the results according to the categories. For local matchers and global matcher with disjoint intervals, precision values above 0.75 are higher than global matcher with continuous intervals. Instead the latter one has higher values above 0.75 for recall than local matchers and global matcher with disjoint intervals. As it is shown in the last column, local matcher has more values above 0.75 (43% of values are greater than 0.75) and below 0.25 (16.4% of values) than the global matcher with disjoint intervals. This is an indication that a local matcher produces very good results but also very bad results in contrast to a global matcher with disjoint intervals that has most of the values within 0.25-0.50 and 0.50-0.75.

Comparison of selection function Figure 8 shows the selection functions that comprises top-k, neighbor-x and their combination. As we may notice, top-2 considerably increases recall and decreases precision. F-measure is slightly higher for neighbor-10 and the combination function (top-2, proxy-10) than the top-2 function.

Comparison of normalization function Figure 9 compares the estimated precision, recall and F-measure on different normalization functions. The differences are quite small and F-measure is the smallest for tfif although it has the highest precision.

`holdsPoliticalPosition` Property Figure 10 shows the contribution of applying two matching functions and two variations of them. Applying *glob-disj-intervals* increases precision, but it is at the price of a big drop in recall. This is very similar to what happen to the playFor property. The *glob-conti-intervals* matcher, on the other hand, increases the recall but not significantly as the *loc-diag-penal*. We observe that F-measure is improved over baseline methods by 20-45%. The best result is achieved when applied the configuration with selection function neighbor-10 and normalization function is local.

`ismarriedTo` Property Figure 11 shows the contribution of applying again the 4 matching functions. We observe that global matchers as for the other two properties improve precision and recall, but overall the average F1 is better for the local matcher with the penalization on the diagonal. The best result is achieved when applied the configuration with selection function neighbor-7 and normalization function is local.

`US administration properties.` For all the US administration office properties as shown in Table 6, we construct the RIM and the external evidence

[23] The denominator will be always one

Figure 7: Comparing different matching function

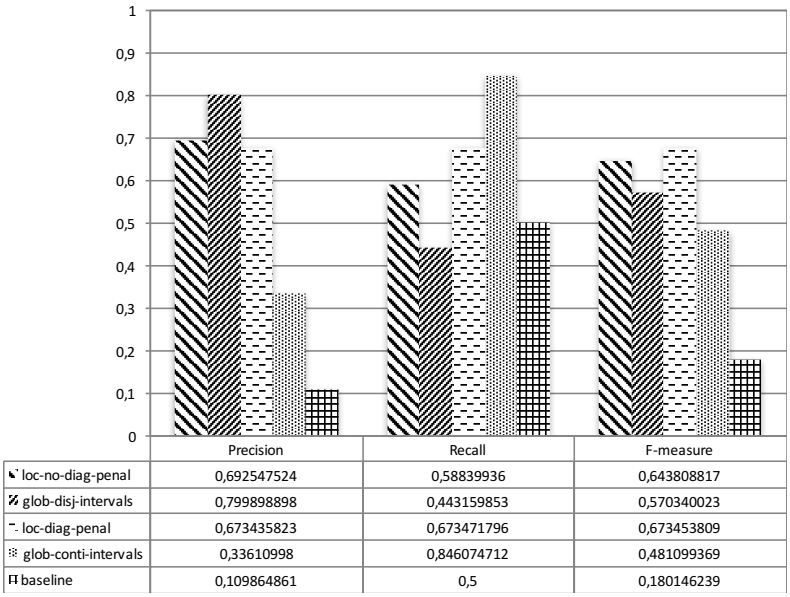


Table 9: A detailed analysis on the three matching function for the `playFor` property.

	Precision			Recall			F-measure		
	local-no-diag-penal	glob-conti-intervals	glob-disj-intervals	local-no-diag-penal	glob-conti-intervals	glob-disj-intervals	local-no-diag-penal	glob-conti-intervals	glob-disj-intervals
<0.25	15%	50%	6.1%	19%	7.4%	47%	16.4%	32%	11%
0.25-0.50	10%	25.5%	1.9%	12.6%	7.3%	18%	15.2%	35.6%	19%
0.50-0.75	23%	11.5%	15.9%	12.4%	8.3%	23%	25.4%	20.4%	31%
>0.75	52%	13%	76.1%	56%	77%	12%	43%	12%	39%

as explained in Section 8.2. Figure 12 shows the precision, recall and f-measure for the three US administration properties. The results are very good with respect to the other DBpedia properties. The accuracy of our approach is related to the quality of the input which in this case shows that the Google Book ngram has more accurate data available than DeFacto that collects data from the Web.

8.4. Scalability

To test the scalability of our approach including all the steps, we computed the running times by varying the sizes of the data. In particular we used the data of the `playFor` property. We equally divided the data into 10 subsets. We tried to keep the size of the splits similar although in some cases it was not possible because the same subjects were kept in the same subset. We ran the experiments on the first subset and incrementally added all the others. Figure 13 shows the execution time of our approach. We observe that (i) our approach terminated in 3905 milliseconds for 180 distinct subjects (around 1000 facts) and (ii) the execution time grows nearly linear in the size of the data which shows the scalability of our approach.

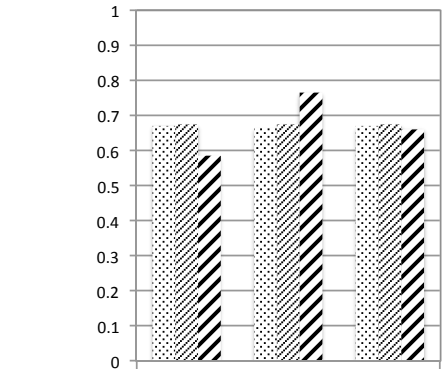
9. Summary and Discussions

This paper studies the problem of determining and mapping time intervals to dynamic facts. We proposed a framework comprising several functions and configuration parameters that can efficiently provide the matching and the selection of the set of time intervals that maximize the effectiveness of our approach. In addition, we proposed a running prototype, TISCO that will support

the users in exploring facts with temporal scopes and simplify the testing of new algorithms for matching and selection functions. We evaluated our approach on facts extracted from DBpedia by using cleaned-up temporal scopes extracted from Yago2. We provide the benchmarking dataset to support the users in testing their approaches since it is difficult to perform the task at hand.

Our approach achieved promising results for temporal scoping of facts by employing various matching functions. The worse results are usually related to the shape of the input data: relevant time points and unstructured evidence vector. The UEV has an exponential growth and thus our algorithm tends to capture the most recent time intervals while failing with less recent dates.

Figure 8: Comparing different selection function



	Precision	Recall	F-measure
top-2,proxy-10	0.672388148	0.66462162	0.668482327
proxy-10	0.673435823	0.673471796	0.673453809
top-2	0.583453964	0.763957592	0.661615352

Figure 9: Comparing different normalization functions.

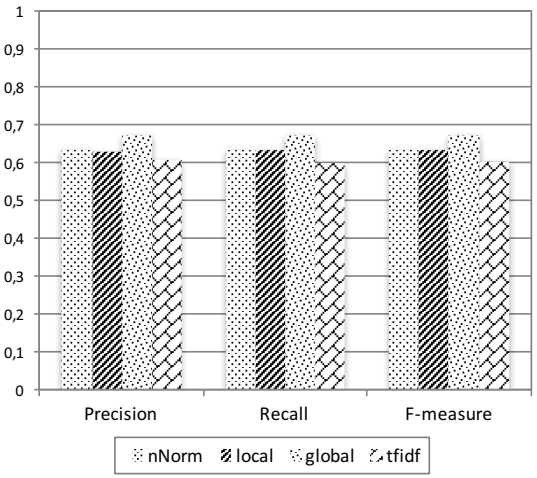
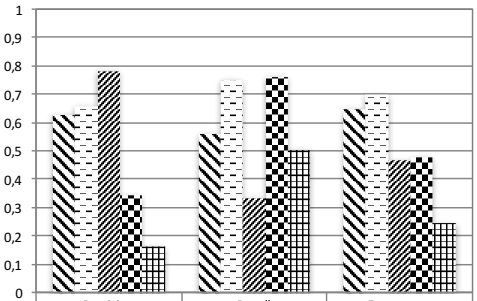
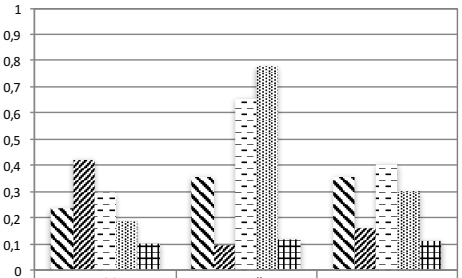


Figure 10: Comparing different matching functions on `holdsPoliticalPosition` property



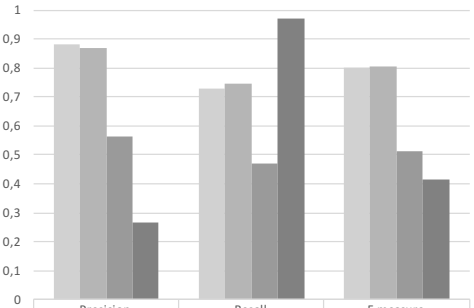
	Precision	Recall	F-measure
loc-no-diag-penal	0,625153668	0,558064799	0,648904218
loc-diag-penal	0,654361792	0,75068105	0,699219956
glob-disj-intervals	0,780444444	0,333789056	0,46759286
glob-conti-intervals	0,344916661	0,758712748	0,474240113
baseline	0,164327946	0,500701418	0,247445421

Figure 11: Comparison on matching functions on `ismarriedTo` property



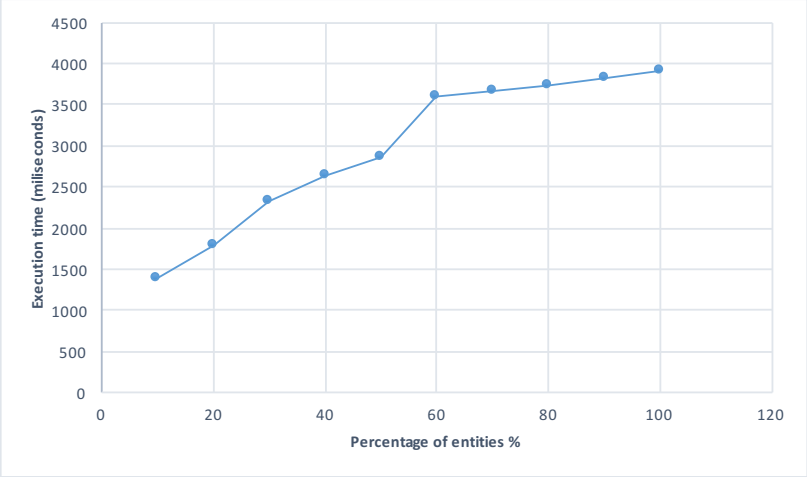
	Precision	Recall	F-measure
loc-no-diag-penal	0,235927371	0,358425129	0,357893435
glob-disj-intervals	0,423076923	0,099418685	0,161003273
loc-diag-penal	0,295914438	0,651564472	0,40699024
glob-conti-intervals	0,186587298	0,77943278	0,301095721
baseline	0,104034087	0,118767019	0,110913439

Figure 12: Comparison of matching functions on `US administration office` properties



	Precision	Recall	F-measure
loc-no-diag-penal	0,882380952	0,729708995	0,798815623
loc-diag-penal	0,869047619	0,747566138	0,803742476
glob-disj-intervals	0,563462669	0,468930041	0,511868342
glob-conti-intervals	0,265573077	0,972016461	0,417168042

Figure 13: Scalability of our algorithm.



Another case, is the length of the UEV. In cases when UEV has length two and the same UEV is captured for different facts of the same subject, the local matching functions fail while the global matching functions get better results. We get better results for politician and players since more evidence is found on the Web than for celebrities. In addition, politician facts get better results since they are considered as functional properties while soccer players playing for a club may be at the same time loaned out to other club and they also plays in a national team.

However there is still room for improvement. Next, we discuss possible future directions. First, it would be interesting to apply the hybrid approach of extracting temporal information within further domains. Therefore, we plan to enlarge the unstructured evidence also by considering tools that extract temporal expression such as HeidelTime or other temporal expressions temponyms as proposed in [**?**]. In addition, we plan to compare our approach with the approaches proposed in the Text Analysis Conference(TAC)[24] in the KBP research area in particular in the Temporal Slot Filling Task [20]. As we already mentioned our approach is completely unsupervised and the NLP techniques we adopt are softer with respect to the NLP techniques employed in TAC 2010 KBP task or other similar works [11]. Moreover, our approach will investigate how to complement evidence retrieved from texts with evidence from the Web of Data.

[24] http://www.nist.gov/tac/

Acknowledgements

This research has been supported in part by the research grant number 17A209 from the University of Milano-Bicocca and by a scholarship from the University of Bonn.

References

[1] ALLEN, J. F. Maintaining knowledge about temporal intervals. *Communications of the ACM 26*, 11 (1983), 832–843.

[2] ALONSO, O., STRÖTGEN, J., BAEZA-YATES, R., AND GERTZ, M. Temporal Information Retrieval: Challenges and Opportunities. In *1st Temporal Web Analytics Workshop at WWW* (2011), pp. 1–8.

[3] BATSAKIS, S., PETRAKIS, E. G. M., TACHMAZIDIS, I., AND ANTONIOU, G. Temporal representation and reasoning in OWL 2. *Semantic Web 8*, 6 (2017), 981–1000.

[4] BERNERS-LEE, T., FIELDING, R. T., AND MASINTER, L. M. Uniform Resource Identifier (URI): Generic Syntax. RFC 3986, 2005.

[5] BIZER, C., HEATH, T., AND BERNERS-LEE, T. Linked Data - The Story So Far. *IJSWIS* (2009), 1–22.

[6] BOLLACKER, K., EVANS, C., PARITOSH, P., STURGE, T., AND TAYLOR, J. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data* (2008), SIGMOD '08, ACM, pp. 1247–1250.

[7] CAMPOS, R., DIAS, G., JORGE, A. M., AND JATOWT, A. Survey of temporal information retrieval and related applications. *ACM Comput. Surv. 47*, 2 (Aug. 2014), 15:1–15:41.

[8] CARLSON, A., BETTERIDGE, J., WANG, R. C., HRUSCHKA, JR., E. R., AND MITCHELL, T. M. Coupled semi-supervised learning for information extraction. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining* (2010), WSDM '10, ACM, pp. 101–110.

[9] CHEKOL, M. W., PIRRÃŠ, G., SCHOENFISCH, J., AND STUCKENSCHMIDT, H. Marrying uncertainty and time in knowledge graphs. In *AAAI* (2017), S. P. Singh and S. Markovitch, Eds., AAAI Press, pp. 88–94.

[10] CORRENDO, G., SALVADORES, M., MILLARD, I., AND SHADBOLT, N. Linked timelines: Temporal representation and management in linked data. In *1st International Workshop on Consuming Linked Data at International Semantic Web Conference*. 2010.

[11] DERCZYNSKI, L., AND GAIZAUSKAS, R. Information retrieval for temporal bounding. In *4th ICTIR* (2013), ACM, pp. 29:129–29:130.

[12] ESTEVES, D., RULA, A., REDDY, A. J., AND LEHMANN, J. Toward veracity assessment in RDF knowledge bases: An exploratory analysis. *J. Data and Information Quality 9*, 3 (2018), 16:1–16:26.

[13] FIELDING, R., GETTYS, J., MOGUL, J., FRYSTYK, H., MASINTER, L., LEACH, P., AND BERNERS-LEE, T. Hypertext transfer protocol – http/1.1 (rfc 2616). Request For Comments, 1999. available at `http://www.ietf.org/rfc/rfc2616.txt`, accessed 7 July 2006.

[14] GERBER, D., ESTEVES, D., LEHMANN, J., BÜHMANN, L., USBECK, R., NGONGA NGOMO, A.-C., AND SPECK, R. Defacto-temporal and multilingual deep fact validation. *Web Semant. 35*, P2 (Dec. 2015), 85–101.

[15] GERBER, D., AND NGOMO, A.-C. N. Extracting Multilingual Natural-Language Patterns for RDF Predicates. In *18th EKAW* (2012), Springer.

[16] GUSFIELD, D., AND IRVING, R. W. *The Stable marriage problem - structure and algorithms*. Foundations of computing series. MIT Press, 1989.

[17] GUTIÉRREZ, C., HURTADO, C. A., AND VAISMAN, A. A. Temporal RDF. In *2nd ESWC* (2005), pp. 93–107.

[18] HAYES, P. J., AND PATEL-SCHNEIDE, P. F. RDF 1.1 Semantics, W3C Recommendation 25 February 2014, Feb. 2014.

[19] HOVY, D., FAN, J., GLIOZZO, A., PATWARDHAN, S., AND WELTY, C. When did that happen?: linking events and relations to timestamps. In *13th EACL* (2012).

[20] JI, H., GRISHMAN, R., AND DANG, H. Overview of the TAC2011 knowledge base population track. In *TAC 2011 Proceedings Papers* (2011).

[21] KÄFER, T., ABDELRAHMAN, A., UMBRICH, J., O'BYRNE, P., AND HOGAN, A. Observing linked data dynamics. In *ESWC* (2013), pp. 213–227.

[22] KLYNE, G., AND CARROLL, J. J. Resource description framework (RDF): Concepts and abstract syntax. World Wide Web Consortium, Recommendation REC-rdf-concepts-20040210, February 2004.

[23] KOUBARAKIS, M., AND KYZIRAKOS, K. Modeling and Querying Metadata in the Semantic Sensor Web: the Model stRDF and the Query Language stSPARQL. *The 7th ESWC* (2010), 425–439.

[24] KUZEY, E., AND WEIKUM, G. Extraction of temporal facts and events from wikipedia. In *Proceedings of the 2Nd Temporal Web Analytics Workshop* (New York, NY, USA, 2012), TempWeb '12, ACM, pp. 25–32.

[25] LAWLER, E. L. *Combinatorial optimization: networks and matroids*. Courier Corporation, 1976.

[26] LEHMANN, J., GERBER, D., MORSEY, M., AND NGONGA NGOMO, A.-C. DeFacto - deep fact validation. In *11th ISWC* (2012), Springer, pp. 312–327.

[27] LEHMANN, J., ISELE, R., JAKOB, M., JENTZSCH, A., KONTOKOSTAS, D., MENDES, P. N., HELLMANN, S., MORSEY, M., VAN KLEEF, P., AUER, S., AND BIZER, C. Dbpedia - A large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web 6*, 2 (2015), 167–195.

[28] LING, X., AND WELD, D. S. Temporal information extraction. In *25th AAAI* (2010).

[29] MANI, I., AND WILSON, G. Robust temporal processing of news. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics* (Stroudsburg, PA, USA, 2000), ACL '00, Association for Computational Linguistics, pp. 69–76.

[30] MEHDI SAMADI, MANUELA VELOSO, M. B. OpenEval: Web information query evaluation. In *27th AAAI* (2013).

[31] MOTIK, B., PATEL-SCHNEIDER, P. F., AND PARSIA, B. Owl 2 web ontology language: Structural specification and functional-style syntax (second edition). *W3C Recommendation, 11 December 2012* (2012).

[32] NAKAMURA, S., KONISHI, S., JATOWT, A., OHSHIMA, H., KONDO, H., TEZUKA, T., OYAMA, S., AND TANAKA, K. Trustworthiness analysis of web search results. In *11th ECDL* (2007).

[33] PASTERNACK, J., AND ROTH, D. Generalized fact-finding. In *20th WWW* (2011).

[34] PASTERNACK, J., AND ROTH, D. Making better informed trust decisions with generalized fact-finding. In *20th IJCAI* (2011).

[35] RULA, A., PALMONARI, M., HARTH, A., STADTMÜLLER, S., AND MAURINO, A. On the diversity and availability of temporal information in linked open data. In *11th ISWC* (2012).

[36] RULA, A., PALMONARI, M., AND MAURINO, A. Capturing the Age of Linked Open Data: Towards a Dataset-independent Framework. *DQMST at IEEE ICSC* (2012).

[37] RULA, A., PALMONARI, M., NGOMO, A. N., GERBER, D., LEHMANN, J., AND BÜHMANN, L. Hybrid acquisition of temporal scopes for RDF data. In *11th ESWC* (2014), pp. 488–503.

[38] SAMADI, M., TALUKDAR, P., VELOSO, M., AND BLUM, M. Claimeval: Integrated and flexible framework for claim evaluation using credibility of sources. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence* (2016), AAAI'16, AAAI Press, pp. 222–228.

[39] SUCHANEK, F. M., KASNECI, G., AND WEIKUM, G. YAGO: A large ontology from wikipedia and wordnet. *J. Web Sem. 6*, 3 (2008), 203–217.

[40] TALUKDAR, P. P., WIJAYA, D. T., AND MITCHELL, T. Coupled temporal scoping of relational facts. In *5th WSDM* (2012), pp. 73–82.

[41] UMBRICH, J., HAUSENBLAS, M., HOGAN, A., POLLERES, A., AND DECKER, S. Towards Dataset Dynamics: Change Frequency of Linked Open Data Sources. In *3rd Linked Data on the Web Workshop at WWW* (2010).

[42] UZZAMAN, N., AND ALLEN, J. F. Trips and trios system for tempeval-2: Extracting temporal information from text. In *SemEval* (2010), ACL, pp. 276–283.

[43] VRANDEČIĆ, D., AND KRÖTZSCH, M. Wikidata: A free collaborative knowledgebase. *Communications of the ACM 57*, 10 (2014), 78–85.

[44] WANG, Y., DYLLA, M., REN, Z., SPANIOL, M., AND WEIKUM, G. Pravda-live: interactive knowledge harvesting. In *21st CIKM* (2012), ACM, pp. 2674–2676.

[45] WANG, Y., ZHU, M., QU, L., SPANIOL, M., AND WEIKUM, G. Timely YAGO: Harvesting, Querying, and Visualizing Temporal Knowledge from Wikipedia. In *13th EDBT* (2010), pp. 697–700.

[46] YIN, X., HAN, J., AND YU, P. S. Truth discovery with multiple conflicting information providers on the web. *IEEE Trans. Knowl. Data Eng. 20*, 6 (2008), 796–808.