As Time Goes By: Comprehensive Tagging of Textual Phrases with Temporal Scopes

Erdal Kuzey Vinay Setty Jannik Strötgen Gerhard Weikum
Max Planck Institute for Informatics
Saarbrücken, Germany
{ekuzey,vsetty,jstroetge,weikum}@mpi-inf.mpg.de

ABSTRACT

Temporal expressions (TempEx's for short) are increasingly important in search, question answering, information extraction, and more. Techniques for identifying and normalizing explicit temporal expressions work well, but are not designed for and cannot cope with textual phrases that denote named events, such as "Clinton's term as secretary of state". This paper addresses the problem of detecting such *temponyms*, inferring their temporal scopes, and mapping them to events in a knowledge base if present there.

We present methods for this kind of temponym resolution, using an entity- and TempEx-oriented document model and the Yago knowledge base for distant supervision. We develop a family of Integer Linear Programs for jointly inferring temponym mappings to the timeline and knowledge base. This enriches the document representation and also extends the knowledge base by obtaining new alias names for events. Experiments with three different corpora demonstrate the viability of our methods.

Keywords

Temporal Tagging; Temporal Knowledge; Temponyms

1. INTRODUCTION

1.1 Motivation and Problem

Temporal expressions in text documents are important cues for searching information about events in web pages, news articles, and social media, and for analyzing historic perspectives over web and news archives [3, 35]. For example, a user searching for the "Maracana final 2014" should be shown information on the FIFA World Cup Final on July 14, 2014. The answers to a query on "summer festivals in Europe" should include the Roskilde Festival taking place in June/July, but should exclude the Tallinn Music Week taking place in March/April. Finally, a journalist or political analyst looking for the "Alexis Tsipras inauguration" should obtain news, blogs, and user posts on the last Greek election which was on September 20, 2015, and a business analyst looking for the "market reaction to the Alibaba IPO" should see documents from September 18, 2014, or later.

To provide good answers to such time-oriented information needs, it is essential to extract and normalize *temporal expressions* – TempEx's for short – in the underlying documents [38]. TempEx's take different forms:

1. *Explicit temporal expressions* denote a precise time point or period such as "25-01-2015", "Jan 25, 2015", "01/25/15" "January 2015", or "spring 2015". All but the last one have a unique interpretation. The last expression can be normalized as well, by imposing a convention (assuming that spring refers to the northern hemisphere spring) that the months of March, April and May count as spring.
2. *Relative temporal expressions* refer to dates that can be interpreted with respect to a reference date. Examples are "last week", "next Monday", "two days ago", etc. The reference date is typically the publication date of a news article or user post.
3. *Implicit temporal expressions* refer to special kinds of named events that have a unique meaning, often of periodic nature, such as "Valentine's day", "Christmas", etc.
4. *Free-text temporal expressions* refer to arbitrary kinds of named events or facts with temporal scopes that are merely given by a text phrase but have unique interpretations given the context and background knowledge about politics, sports, music, business, etc. Examples are "Roskilde festival", "Greek referendum", "Alibaba IPO", "German triumph in Maracana", "Clinton's time as First Lady", "second term of Angela Merkel", etc.

Figure 1 shows a text snippet about the football player Cristiano Ronaldo with temporal expressions highlighted and their ideal mappings to a knowledge base of events and subject-predicate-object (SPO) facts. Both events and facts have temporal scopes, in the form of time points when events happened or time spans during which facts hold. Our goal in this paper is to detect the free-text phrases on the left side and compute the correct mappings onto the right side. Although all phrases in the example correspond to events in time, some have to be mapped to general facts, such as `Cristiano_Ronaldo playedFor Real_Madrid` rather than entities of type event. The knowledge base has freedom to choose among different representations, and it contains facts about stateful relationships without necessarily having explicit events for the begin and end of the relationships.

WWW 2016, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4143-1/16/04.
http://dx.doi.org/10.1145/2872427.2883055 .

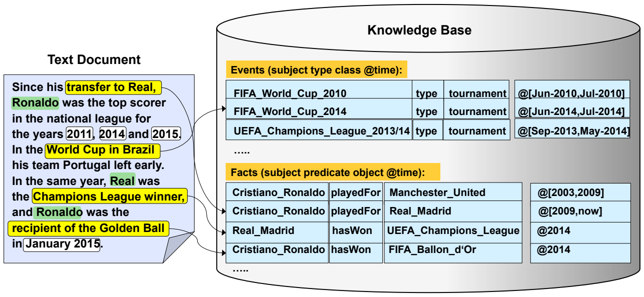

Figure 1: Example of text with temporal expressions and mappings to a knowledge base.

Limitations of State-of-the-Art Temporal Tagging: In recent years, good solutions have been developed for explicit, relative, and implicit dates. Most notably, tools like HeidelTime [36] and SUTime [9] perform TempEx extraction and normalization, and handle many such cases with very good precision and recall. However, there is hardly any work on free-text TempEx's, which are addressed in this paper.

Limitations of State-of-the-Art Entity Linking: These kinds of textual expressions about events are a special case of homonyms for individual entities: ambiguous phrases that denote people, places, companies, products, etc. Mapping names and phrases to entities in a data or knowledge base is known as *Entity Linking* or – more explicitly – as *Named Entity Recognition and Disambiguation (NERD)*, and there are ample papers and software tools on this task (see overviews like [8, 11, 34]).

Methods for NER, the recognition part, work well for names of people, places, organizations, and also for explicit TempEx's, but have poor recall for sophisticated expressions of temporal nature like the ones considered in this paper. General NED approaches for the disambiguation part produce decent results on people names. In addition, specialized solutions have been suggested for places, namely so-called toponym resolution approaches for geo-spatial entities [24, 26]. However, the normalization of events and facts is more challenging and to the best of our knowledge, there is no prior work that specifically tackles the issue of extracting and disambiguating free-text TempEx's. In analogy to toponyms, we refer to this class of TempEx's as *temponyms*.

Note that general-purpose NED is inadequate for free-text TempEx's for two reasons. First, NED works well when it can exploit coherence (relatedness) measures between the candidate entities for different textual mentions. For example, knowing that the entities `Cristiano_Ronaldo` and `Real_Madrid` are highly related, helps jointly disambiguating "Ronaldo" and "Real". However, the cues from the explicit TempEx's that co-occur with a mention do not fall under this regime, because values like 2011, 2014, 2015 are not entities – so NED methods do not have a coherence measure between say 2014 and the entity `FIFA_Ballon_d'Or`. Second, our solution space includes mapping temponyms not just to entities of type event, but possibly also to entire SPO facts such as ⟨`Cristiano_Ronaldo playedFor Real_Madrid`⟩. This is completely out of scope for NED.

Problem statement: The problem addressed in this paper is *temponym resolution*: given a knowledge base (KB) of events with precise time points or periods as well as other entities, take as input an arbitrary text document from the web, news or social media, detect all free-text TempEx's, extract them, infer their temporal scopes, and map them to proper events or facts in the KB, thus canonicalizing their repesentation. Especially, the mapping to SPO facts is a demanding and novel approach that is beyond the scope of NED and has not been considered in prior work.

By solving this problem, we create added-value markup of text documents, which is a key asset for semantic search, query understanding, summarization, deep text analytics, KB curation, and other tasks. In addition to enriching input documents with links to the KB, temponym resolution also enhances the KB itself by providing additional alias names (aka. paraphrases) for known events and detecting emerging events that are not yet in the KB at all.

1.2 Approach and Contribution

State-of-the-art TempEx taggers such as HeidelTime [36] and SUTime [9] are based on regular expression matching, handcrafted rules, and background dictionaries. For *temponym detection* in text documents, we adopt a similar approach and develop a rule-based system that uses similarity matching in a large dictionary of event names and known paraphrases. For example, in an input sentence like "Clinton served in the Obama administration", the temponym "Obama administration" could be matched to a paraphrase of the event "presidency of Barack Obama" which has its own Wikipedia article and is an explicit entity in large knowledge bases. The input sentence "Beyoncé toured with Jay-Z shortly after her marriage with him" is more challenging, as there is no dedicated article or entity on this marriage in Wikipedia or any knowledge base. In such cases, we attempt to match the temponym against subject-predicate-object (SPO) facts in a knowledge base, like ⟨`Beyoncé, spouse, Jay-Z`⟩ based on the cue that "marriage with" and "spouse of" are paraphrases of the same predicate. In knowledge bases like Freebase or Yago2 [17], such facts about relationships between two entities often have explicit time scopes denoting the validity timespans.

This kind of partial matching with paraphrase dictionaries yields candidates for temponym resolution, but tends to produce a fairly noisy space of hypotheses. The next step – *tempoynm disambiguation* – is the key challenge tackled in this paper. To this end we harness the insight that temponyms co-occur with other TempEx's and also with mentions of other entities involved in the event or fact to which the temponym should be mapped. By first resolving the simpler kinds of explicit, relative and implicit TempEx's and by mapping co-occurring names of people, places, etc. to entities, we can create a rich set of features around a temponym and leverage these features for disambiguation. We further develop this idea into an *Integer Linear Program (ILP)*, whose solution yields good results yet can be efficiently computed on a per-temponym basis.

One limitation of this *local ILP* is that it does not consider other temponyms in the proximity nor does it take into account that the NED mapping for non-temporal entities is error-prone. Therefore, we improve this approach and devise a *joint ILP* that is aware of the NED uncertainty and computes a joint mapping of all temponyms and other entity mentions within a given document. Figure 1 illustrates this situation. Finally, an additional aspect to consider for joint inference is that the same temponym may occur in different documents. To leverage this richer context, we devise a *global ILP* that processes all temponyms and entity mentions of multiple documents simultaneously.

In summary, our contributions in this paper are

- the development of the first model for free-text temponym resolution that uses joint inference for high-quality mappings to a knowledge base;
- tractable methods and a full system for enriching text documents by events and for gathering additional paraphrases of events;
- comprehensive experiments with three corpora (biographies, history documents, news articles) that demonstrate the viability and quality of our solution.

2. PRIOR WORK AND BACKGROUND

Temponym resolution as defined in this paper has not been addressed in any prior work. However, there are several related research topics and we drawn from some of their results as building blocks for our approach.

Temporal expressions in explicit, relative and implicit form (see Section 1.1) have been extensively studied as part of the TempEval competitions [38]. HeidelTime [36], SUTime [9] and Tarsqi [39] are some of the best performing systems that mostly rely on deterministic rules over regular expressions to perform both detection and normalization of TempEx's. The recent work of [23] pursues an alternative approach by learning context-dependent semantic parsers for TempEx's. None of this prior work addresses the class of free-text temponyms.

Event extraction in computational linguistics: There is also considerable work in NLP on events in narrative texts, based on the TimeML markup language [32], e.g., in "His first attempt to climb Everest was unsuccessful", "`to climb`" is an event. Work along these lines includes [6, 21, 31, 39]. Recent work has further extended this direction to detect and align events in narrative texts using machine learning techniques [13, 19], with the specific target of clinical reports. Here events refer to the course of diseases and therapies of patients. The event definition used in all these works differs fundamentally from our notion of temponyms.

Event extraction in web mining has focused on discovering events in news and generating storylines; see, e.g., [4, 12, 33]. However, the events found by these methods are not canonicalized and cannot be uniquely mapped to events in a knowledge base. Rather the output merely has the form of clusters of news articles or subgraphs of interrelated entities.

Knowledge bases (KB's) are large repositories of individual entities like people, places, organizations, creative works (books, songs, etc.) and events, their memberships in semantic classes (aka. `type` or `instanceOf` predicate), and their attributes and relationships with other entities. Popular, publicly available KB's are DBpedia (`dbpedia.org`), Freebase (`freebase.com`), Wikidata (`wikidata.org`) and Yago (`yago-knowledge.org`). The contents of these KB's are in the form of subject-predicate-object (SPO) triples, following the RDF data model. For example, Hillary Clinton's position as secretary of state is captured by the triple

⟨`Hillary_Clinton holdsPosition US_Secretary_of_State`⟩,
and her marriage has the form

⟨`Hillary_Clinton isMarriedTo Bill_Clinton`⟩.

Events associated with time points are captured as entities with their respective types, for example:

⟨ `2014_FIFA_World_Cup_Final type football_tournament`⟩.

Temporal knowledge: A few KB's, most notably Freebase and Yago2, have augmented basic SPO facts by temporal (and also spatial) meta-facts. Yago2 [17], which is used in our work, provides temporal scopes either by its `happenedOn` predicate, for example ⟨`2014_FIFA_World_Cup_Final happenedOn 2014-07-13`⟩, or assigns time points or periods to reified facts. For example, for Clinton's term as secretary of state with fact id `f1` and for her marriage with fact id `f2`, the temporal meta-facts have the form:
⟨`f1 validDuring [2009-01-21,2013-02-01]`⟩ and
⟨`f2 validDuring [1975-10-11,now]`⟩.
The Yago2 methods for harvesting this temporal knowledge tap into infoboxes, categories, and lists of Wikipedia and use consistency reasoning for high-quality output [17, 22]. Examples of such *SPOT facts* with their temporal scopes taken from the Yago2 knowledge base are shown in Table 1.

Other methods for extracting temporal facts from text web sources or inferring the temporal scopes of known facts have been developed by [27, 37, 40, 41]. None of these machine-learning-based techniques has succeeded in scaling to large input and yielding high-quality output with precision above 90% and decent recall.

Named entity recognition and disambiguation (NER/NED) is the general task of detecting names and phrases that denote entities (NER) and mapping them to canonicalized entities in a knowledge base (NED). NER is typically based on trained CRF's using lexico-syntactic linguistic features. The most popular tool is the Stanford NER Tagger [15]. The best NED methods combine statistical priors about surface names, the contextual similarity between a mention in an input text and descriptions and properties of candidate entities, and the semantic coherence between candidate entities for different mentions. [8, 11, 34] are overviews of different methods and tools, and their experimental behavior. The special case of toponym resolution, for geo-entities, exploits spatial relations between candidate places (their distance). State-of-the-art techniques include [26, 24]. However, the special case of temponym resolution has not received any attention so far.

Time-sensitive information retrieval has recently gained much attention, as a substantial fraction of web queries have temporal aspects [3]. Ranking models that capture the temporal scope of queries and documents have been developed in [5, 10, 25, 29]. In addition, there is growing interest in the role of time for search-result snippet generation [1], query classification [16, 20], timeline visualization [2, 43, 44], mining web archives and online communities [35, 42], and further tasks in web contents analytics.

3. SYSTEM OVERVIEW

Our system takes different text sources (news, biographies, encyclopedic articles) as input. The entire process of temponym resolution is divided into two steps. First, a set of significant phrases are extracted from the input text, performing the *temponym detection*. Second, these phrases are disambiguated onto canonicalized events or facts in the KB, performing the *temponym disambiguation*.

Figure 2 illustrates the architecture of our system. The pipeline starts with processing input text documents to detect noun phrases and named entity mentions. By using a mention-entity dictionary [18], we obtain a set of candidate entities for each mention. Similarly, by using a pattern dictionary, we obtain the noun phrases that are possible tem-

S	P	O	T
SyrianCivilWar	type	Event	[2011-03-15,now]
WorldWarI	type	Event	[1914-07-28,1918-11-11]
AngelaMerkel	type	GermanChancellor	[2005-11-20,now]
AngelaMerkel	type	EnvironmentMinister	[1994-11-17,1998-10-26]
AngelaMerkel	bornIn	Hamburg	[1954-07-17,1954-07-17]
FCBayernMunich	won	UEFAChampionsLeague	[2013-05-25,2013-05-25]
FCBayernMunich	won	UEFAChampionsLeague	[2001-05-23,2001-05-23]

Table 1: Examples of SPOT facts.

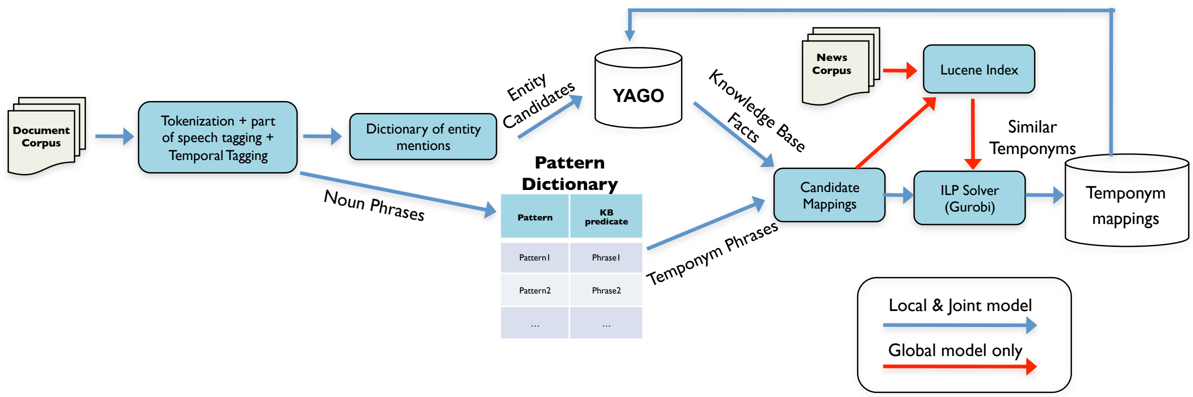

Figure 2: The processing pipeline for temponym detection and resolution.

ponym phrases. These two sets are later fed into integer linear programs (ILP's). The ILP's use different constraints and objective functions to jointly disambiguate mentions to entities and resolve the temponyms. Finally, the temponyms that are mapped to facts are added to the KB for knowledge enrichment. This is the flow for the local and joint ILP models we devised for temponym disambiguation. Additionally, in the global model, we exploit a news corpus to mine more relevant cues to enrich the context of temponym candidates and enhance the efficacy of temponym resolution using ILPs.

Inputs. The following inputs are used by our system.

- Text inputs. We use the Stanford NLP software (`nlp.stanford.edu/software/corenlp.shtml`) for tokenization and part of speech tagging in input documents.
- Knowledge base. We use the Yago knowledge base (`yago-knowledge.org`) to provide us with entities, facts about entities, semantic types of entities, and textual surface forms (alias names) of entities.
- Repository of relational patterns. Based on the PATTY tool [30], we created a dictionary of lexico-syntactic patterns with semantic type signatures for each of the KB relations. We specifically tailor this repository for events and temporal facts. The details of how this repository is constructed are given in Section 4.1.

Significant phrases. We detect several types of textual expressions:

- TempEx's. We identify TempEx's by using the HeidelTime tagger [36]. The TempEx's are later used to construct time histograms for the input documents, which are cues for temponym resolution.
- Mentions. We detect entity mentions by using the Stanford NER tool [15]. These are later mapped to canonical entities in the knowledge base during joint disambiguation of temponyms and entities. A bottleneck is that person, organization, and location entities are addressed by general purpose NER tools. For instance, the phrase "*The United States presidential election of 2016*" is not detected by the NER stage, although it is a named event. Our temponym detector compensates for this limitation, and extracts such mentions for the entities of type event.
- Temponyms. We explain how we detect temponym expressions in Section 4.

Candidate generation. In this step, our system generates entity candidates for mentions and event or fact candidates for temponyms to along with their time scopes. The candidate generation of facts is guided by the repository of relational patterns.

Outputs. Mentions of (non-event-type) entities are disambiguated to canonical entities, and temponyms are mapped to the named events or to facts in the KB. These two tasks are coupled and jointly solved by our ILP methods.

Note that significant phrases might overlap; the disambiguation stage of our system chooses a consistent subset of phrases mapped to targets in the KB. All resolved temponyms are added to the KB. Thus, we provide *knowledge enrichment* in two ways: i) disambiguating the phrases in the text and creating semantic markup, and ii) extending the KB with new "rdfs:label" triples that have a temponym phrase as subject, and an event or a fact as object.

4. TEMPONYM DETECTION

Temponyms typically appear as noun phrases in text. A noun phrase can generally refer to i) the name of an entity (e.g., "president Obama" or "the US president"), ii) a class (type) of entities (e.g., "US presidents" or "football clubs"), iii) a general concept (e.g., "the climate change" or "linear algebra"), iv) textual patterns with temporal scope (e.g., "Greek referendum" or "the FIFA final"), v) miscellaneous cases (e.g., idioms, quotes, etc.).

In order to gather noun phrases for case (iv) from a given input text, we use a small number of handcrafted regular expressions over word sequences and their part-of-speech

(POS) tags (i.e., word categories). For example, the regular expression $[DT]^*$ $[JJ]^*$ $[NN]^+$ matches all phrases optionally start with an article (POS tag DT), optionally have one or more adjectives (POS tag JJ), and are followed by one or more nouns (POS tag NN). Examples are "the next presidential election" or "a memorable champions league final".

These regular expressions are fairly liberal so as not to miss any candidate phrases. Only a small portion of the captured noun phrases are indeed temponyms. The temponyms are differentiated from the other cases by harnessing lexico-syntactic patterns for binary relations, as discussed in Subsection 4.1 below. These patterns are chosen conservatively to eliminate false positives.

4.1 Lexico-Syntactic Patterns

To detect temponyms, we leverage known temporal facts and events in the knowledge base that have associated temporal scopes.

- Temporal facts. These are the facts about entities that the relationship between entities holds for a certain time period. Such relationships are represented through temporal predicates e.g., `spouse, hasAcademicAdvisor, isCeoOf, playsForFootballClub`, etc.
- Events. These are the named event entities in the knowledge base that are associated with semantic event types e.g., `election, cup final, music album release,` etc.

We create a dictionary of lexico-syntactic patterns with type signatures, and these patterns serve as cues detecting the above two cases and distinguishing them.

Patterns for temporal facts. We harness the PATTY repository [30] for the subset of predicates that have temporal scopes, focusing on noun phrases. PATTY contains a total of 160 patterns for temporal predicates. We keep only patterns above a certain confidence as computed by PATTY. These confidence scores are later used during temponym resolution as the semantic similarity measure.

Patterns for events. Starting with a large pool of noun phrases, we run them through a noun group parser that determines their semantic head words (e.g., "referendum" in "Greek bailout referendum" or "victory" in "Germany's victory in Maracana"). These head words are used to test whether a phrase falls into a class of event types. We considered doing this by disambiguating the head words onto WordNet [14], but WordNet has some misleading information in its type hierarchy, for example, placing movies under the type of events.

To overcome this problem, we harness the knowledge base instead, specifically Yago2. There are three relations in Yago2 that only accept an event as their domain and a date as their range. These predicates are `startedOn, endedOn, happenedOn`. Since Yago2 has strong type checking constraints, the instances of these predicates are very precise, above 95% [17]. We use the names of the left-hand arguments of the facts in these three predicates to generate a list of noun-phrase head words denoting events with high probability, for example, "*election, festival, final*", etc. However, this approach still yields false positives. For example, we derive "*woodstock*" from "Woodstock" which is a festival, and "*concert*" from "Concert of Europe" which is a conference.

We cope with such false positives as follows: We observe that noun-phrase head words that denote events mostly agree with the Yago2 event categories. The Yago2 event categories in turn yield additional cues for deciding whether a noun phrase is an event or not. Examples are "Elections in 1981, Electronic music festivals in Turkey, political scandals", etc. The above case of "Woodstock" is in categories like "rock festivals", "music festivals", etc., and these are the cues to pick up only "*festival*" as an event type.

This observation is leveraged as follows:

- The Yago2 event category names are parsed, and the names of their instances (compiled from the predicates `startedOn, endedOn, happenedOn`) are parsed as well, giving us head words for both categories and instances.
- A mutual information measure between category-level and instance-level head words is computed as below:
$$P[eventHead|catHead] = \frac{P[eventHead, catHead]}{P[catHead]}$$
The mutual information scores are later used during temponym resolution as semantic similarity measure.
- We estimate the instance-level head words denoting events with mutual information above a threshold. Thus, false positives coming from specific event names are removed.

Using these techniques, we generate a list of noun-phrase head words that denote events with very high probability. This method resulted in a dictionary for type predicates that maps nouns to event types. This dictionary has 210 patterns for event detection. In total, we have 370 patterns for fact and event detection. Examples of event and fact patterns are shown in Table 2.

4.2 Temponym Features

Our system applies lexico-syntactic patterns on the extracted noun phrases from the input text, and extracts the temponym phrases with the following features:

1. the facts (i.e., predicate instances) matching the temponym based on the paraphrase dictionary, along with confidence scores;
2. the TempEx's appearing in the context of the temponym;
3. the entity mentions in the context of the temponym;
4. the sentence that the temponym appears in;
5. the provenance info about the temponym: document URL, (publication) timestamp of document, etc.

5. MAPPING TEMPONYMS TO THE KNOWLEDGE BASE

Given a set of temponyms together with their contextual cues, the final – and most difficult – task in temponym resolution is to disambiguate them onto the KB facts and events. This difficulty is reflected in the limitations of using existing NERD techniques to resolve temponyms. We make a vital observation that NERD techniques typically fail to capture named events. This is due to the fact that unlike people, organizations, and places, the temponyms are not always capitalized. For example, the expressions like "the first Winter Olympics hosted by Russia", are typically not resolved by NERD techniques. As we experimentally show in Section 6, it is evident that, NED tools such as AIDA miss significant fraction of temponyms. Moreover, the previous NERD techniques cannot cope with temponyms that are facts such as "Obama's presidency". To this end, we posit that considering entity coherence and temporal coherence jointly is crucial in resolving temponyms that are otherwise ignored.

Nominal pattern	KB Predicate	Domain	Range
"receiving"	hasWonPrize	wn_person	wn_award
"nomination"	hasWonPrize	wn_person	wn_award
"inauguration"	holdsPosition	wn_person	wn_politicalPost
"inauguration"	rdf:type	rdfs:resource	wn_inauguration
"presidency"	holdsPosition	wn_person	wn_politicalPost
"presidency"	isLeaderOf	wn_person	yagoLegalActorGeo
"death"	rdf:type	rdfs:resource	wn_death
"death"	diedIn	wn_person	wn_city

Table 2: Examples of lexico-syntactic patterns for temponym detection.

Therefore, when resolving a temponym, our methods also jointly resolve the following:

i the *entity mentions* in the context of the temponym are disambiguated;

ii the *time point or period* attached to the KB fact or event is propagated to the temponym as additional markup for the input text;

iii the *semantic type* of the temponym is determined (e.g., marriage, election, concert, etc.).

In the rest of this Section, we define several measures that capture the similarities between temponym candidates and KB facts, coherence of entities, temporal expressions and events. Then, we formulate the problem of resolving temponyms as three different Integer Linear Programs (ILP) with an objective to maximize the similarity and coherence. We also introduce several constraints to ensure that the selected mapping is meaningful.

5.1 Candidate Mappings Generation

One of the challenges of resolving temponyms is the large number of entities from the candidate temponyms obtained from Section 4.2. To address this problem we prune the candidates that have low potential of being resolved as temponyms. In this regard, we refer to the named entity dictionary from [18] to obtain a score for the candidates. Using this scoring we rank and select top-k candidates.

Consequently, we derive the candidate facts and events from Yago2. A fact from Yago2 is a candidate fact if it contains an entity from the above mentioned top-k selected candidates as a subject or an object. In addition, a Yago2 fact that is of type `event` is also a candidate mapping if the subject is one of the top-k entities. This gives us a set of candidate mappings for a temponym.

The final goal is to select the best matching mapping among the chosen candidates. For this purpose, given a temponym t and a fact f, we derive measures of relatedness for the mapping between t and f using diverse features such as textual, temporal, and semantic similarity as below:

$w\text{-}text_{tf}$**:** The jaccard string similarity between the tokens of t and f.

$w\text{-}sem_{tf}$**:** The semantic similarity score for the head noun of t and the predicate of f obtained from the pattern dictionary (as explained in Section 4.1)

$w\text{-}temp_{tf}$**:** The temporal similarity of f and the normalized dates in the context of t.

The temporal similarity $w\text{-}temp_{tf}$, between a temponym and a fact is estimated from the divergence between the distribution of the normalized dates in the context of the temponym, and the time scope of the fact. The time scope of a fact is converted to a yearly uniform distribution between the beginning and the end of the time scope. We implement these distributions in the form of histograms.

Definition 1 (**Temporal similarity**). *The temporal similarity between temponym t and fact f is defined as:*

$$w\text{-}temp_{tf} = 1 - JSD(H_t || H_f)$$

JSD (Jensen-Shannon Divergence) is the symmetric extension of the Kullback-Leibler Divergence, and H_t and H_f are the distributions of temporal information in the contexts of t and in the time scope of f. The Jensen-Shannon Divergence between H_t and H_f is defined as:

$$JSD(H_t||H_f) = \frac{1}{2} KL(H_t||M) + \frac{1}{2} KL(H_f||M)$$

Here, $M = \frac{1}{2}(H_t + H_f)$ and KL is the Kullback-Leibler divergence calculated as $KL(H_t||H_f) = \sum_i H_t(i) \log \frac{H_t(i)}{H_f(i)}$.

Using a linear combinations of the three measures described above we compute the relatedness score for fact-temponym mappings as below:

Definition 2 (**Fact-temponym relatedness**). *$w\text{-}rel_{tf}$ is a relatedness measure for temponym t and fact f:*

$$w\text{-}rel_{tf} = w\text{-}text_{tf} + w\text{-}sem_{tf} + w\text{-}temp_{tf}$$

In addition to fact-temponym relatedness, we also consider a probablistic prior coined *Mention-entity prior* denoted as $w\text{-}ned_{me}$ quantifying the similarity of an entity mention m and an existing canonical Wikipedia entity e. $w\text{-}ned_{me}$ is computed based on the frequency of a particular mention m appearing in the inlink anchor texts referring to specific entity e in Wikipedia as described in [18].

The features defined above are derived from a given single temponym and fact pair locally. We further make an important observation that a coherent text from a document contains entities, explicit TempEx's and temponyms that have high mutual relatedness in terms of their semantic and temporal properties. To exploit this, we introduce measures for semantic coherence between entities and temporal coherence between facts.

Definition 3 (**Entity-entity coherence**). *$w\text{-}coh_{ee'}$ is the precomputed Jaccard coefficient of two entities e and e':*

$$w\text{-}coh_{ee'} = \frac{|inlinks(e) \cap inlinks(e')|}{|inlinks(e) \cup inlinks(e')|}$$

where inlinks are the incoming links in the Wikipedia articles for the respective entity.

The semantic coherence enhances the coherent mapping of mentions to semantically related entities. For example, in the example text in Figure 1, the semantic coherence encourages to disambiguate the phrase "Ronaldo" as Portuguese footballer Cristiano Ronaldo rather than the famous Brazilian footballer Ronaldo.

Definition 4 (**Temporal coherence**). *$w\text{-}temp_{ff'}$ is the Jensen-Shannon Divergence between the histograms for* *the temporal scopes of two facts f and f':*

$$w\text{-}temp_{ff'} = 1 - JSD(H_f || H_{f'})$$

The temporal coherence enhances mapping of temponyms to facts that their time scopes are temporally coherent. For example, in the example text in Figure 1, the temporal coherence encourages to disambiguate the phrase "World Cup" as `FIFA_World_Cup_2014` rather than `FIFA_World_Cup_2010`, since the latter is temporally incoherent with other facts.

5.2 Joint Disambiguation and Resolution

Having defined several features that give weights to the temponym fact pairs, the final step is to choose the pair that optimally match. For this purpose, we developed three Integer Linear Program (ILP) models that differ in their scopes for mapping temponyms to the KB.

5.2.1 Local Model

The local model considers a single temponym as input, and uses the mention-entity prior and the relatedness measure to score candidate mappings. We define the ILP variables, the objective function, and the constraints as follows.

Variables
X_{tf} : 1 if temponym t is mapped to fact f, 0 else.
Y_{me} : 1 if mention m is disambiguated as entity e, 0 else.

Objective
maximize
$\sum_{t \in T} X_{tf} \times w\text{-}rel_{tf} + \sum_{m \in M} Y_{me} \times w\text{-}ned_{me}$

Constraints
1. $\sum_f X_{tf} \leq 1, \forall t$
2. $\sum_e Y_{me} \leq 1, \forall m$
3. $X_{tf} \leq \sum_{m, e \in args(f)} Y_{me}, \forall t, f$

Table 3: ILP for the local model.

The local model jointly disambiguates entities and the mappings of temponyms by maximizing the total sum of the fact-temponym relatedness and mention-entity prior. It enforces hard constraints to ensure a consistent set of mappings and disambiguations. Constraint 1 ensures that a temponym is mapped to at most one fact. Constraint 2 ensures that a mention is disambiguated to at most one entity. Finally, Constraint 3 ensures that if a temponym is mapped to a fact, then the fact should contain a disambiguated entity as its subject or object. A better disambiguation of mentions yields a better mappings of temponyms, since a temponym can be mapped to only a fact of a disambiguated entity (Constraint 3).

5.2.2 Joint Model

The joint model extends the local model to jointly resolve all temponyms and disambiguate entities in a given document together by considering semantic coherence between entities and temporal coherence between facts.

The objective, variables and constraints from the local model are borrowed for the joint model. In addition, new objectives to maximize entity coherence and temporal coherence, corresponding variables and constraints are introduced as in Table 4:

The joint model disambiguates entities and selects the mappings of temponyms by maximizing the total sum of the fact-temponym relatedness, mention-entity prior, entity-entity coherence, and temporal coherence of facts that are chosen. It enforces hard constraints to ensure a consistent set of mappings. Constraints 1, 2, 3 are the same as for the local model. Constraints 4, 5, 6 ensure that for any selected pair of entities, the respective mention-entity disambiguations should be selected, too. Constraints 7, 8, 9 ensure that for any selected pair of facts, their respective temponym-fact mappings should be selected, too.

Variables
X_{tf} : 1 if temponym t is mapped to fact f, 0 else.
Y_{me} : 1 if mention m is disambiguated as entity e, 0 else.
$Z_{ee'}$: 1 if both disambiguations $m \to e$ and $m' \to e'$ are selected, 0 else.
$C_{ff'}$: 1 if both mappings $t \to f$ and $t' \to f'$ are selected, 0 else.

Objective
maximize
$\sum_{t \in T, f} X_{t_f} \times w\text{-}rel_{t_f} + \sum_{m \in M, e \in E} Y_{m_e} \times w\text{-}ned_{m_e} + \sum_{ee'} Z_{ee'} \times w\text{-}coh_{ee'} + \sum_{ff'} C_{ff'} \times w\text{-}temp_{ff'}$

Constraints
1. $\sum_f X_{tf} \leq 1, \forall t$
2. $\sum_e Y_{me} \leq 1, \forall m$
3. $X_{tf} \leq \sum_{m, e \in args(f)} Y_{me}, \forall t, f$
4. $Z_{ee'} \leq Y_{me}, \forall me, m'e'$
5. $Z_{ee'} \leq Y_{m'e'}, \forall me, m'e'$
6. $Z_{ee'} + 1 \geq Y_{me} + Y_{m'e'} \forall me, m'e'$
7. $C_{ff'} \leq X_{tf}, \forall tf, t'f'$
8. $C_{ff'} \leq X_{t'f'}, \forall tf, t'f'$
9. $C_{ff'} + 1 \geq X_{tf} + X_{t'f'}, \forall tf, t'f'$

Table 4: ILP for the joint model.

5.2.3 Global Model

Temponyms often occur (possibly in different forms) in multiple documents of a corpus, e.g., news articles in a news corpus. The goal here is to process all relevant temponyms and surrounding entity mentions from different documents. The cues obtained this way enrich the context of temponyms from cues accross different documents. For example, we can combine cues about several temponyms such as *German triumph in Maracana*, *2014 FIFA World Cup Final* and *Argentina vs Germany (2014 FIFA World Cup)* that refer to same event, from different documents, to enrich the context for resolving each of these temponyms.

However, finding relevant temponyms is expensive if every pair of temponyms are considered. Moreover, it is computationally expensive to feed all potential cues to the ILP solver. Therefore, we need to restrict the search space to highly similar temponym candidates. To achieve this, we index all temponyms phrases obtained from a given news corpus[1] with their contextual features.

The global model employs a grouping function that takes a temponym as input and returns a group of temponyms with high similarity. For grouping we use the following features:

i the surface strings of temponyms,
ii the entity mentions in their contexts,

[1]We used GDELT (`http://gdeltproject.org/`) news dataset.

iii the normalized TempEx's in their contexts,
iv the sentences containing the temponyms.

These features one-to-one correspond to the features we considered for computing the candidate mappings and their weights described in Section 5.1.

This task is realized by indexing all the features mentioned above using the Lucene search engine. For each temponym t of interest, we run a multi-field boolean search over the different features of the temponym, retrieving a set S_t of similar temponyms:

$$S_t = \{t' : sim_{Lucene}(t, t') \geq \tau\}$$

where sim_{Lucene} is the similarity score of the boolean vector space model provided by Lucene and τ is a specified threshold. Specifically, the similarity score is computed as:

$$sim_{Lucene}(t, t') = \sum_i \frac{v(t_i) \cdot v(t'_i)}{|v(t_i)||v(t'_i)|}$$

where t_i is the vector for feature group i (string, mentions, TempEx's, sentence) of temponym t.

For each temponym t the context features from the temponyms in S_t are merged. Thus, each temponym is enriched with the contextual information taken from highly similar temponyms in the corpus. Then, the ILP for the joint model is used to compute a solution for the global model. The data flow for the global model is illustrated in Figure 2.

6. EXPERIMENTS

In order to extensively evaluate our methods, we composed four hypotheses:

1. **Detection quality:** Our methods detect temponyms that are either events or facts with a significant coverage.
2. **Disambiguation quality:** Our methods significantly resolve the temponyms for different kinds of text.
3. **Temporal enrichment:** Our methods substantially add temporal information to documents by finding the temporal scopes of temponyms.
4. **Knowledge enrichment:** Our methods substantially add new knowledge to a knowledge base i) by finding new alias names for events and facts, and ii) by finding the time scope of knowledge base facts via anchoring them to resolved temponyms.

Since each hypothesis aims a different research goal, we developed different experimental settings to effectively assess each hypothesis independently. We first introduce the different datasets we used in the experiments.

6.1 Datasets

To evaluate the quality of our methods for temponym resolution, we performed experiments with three datasets with different characteristics: WikiWars, Biographies, and News.
WikiWars. The WikiWars corpus [28] has been popular in benchmarks for temporal tagging (i.e., resolving explicit, relative and implicit TempEx's). It contains 22 long Wikipedia articles about major wars in history. These articles are specifically rich in terms of TempEx's and named events. Thus, the temponyms detected in these articles are mostly of the event type. Note that WikiWars articles are plain text documents that do not contain any structured elements of Wikipedia such as entity links, categories, etc.

WikiBios. These are Wikipedia articles on the biographies of 30 prominent politicians (e.g., Barack Obama, Hugo Chávez, Vladimir Putin). We refer to this dataset as *WikiBios.* In contrast to the WikiWars, this corpus contains fewer event temponyms but features many temponyms that refer to temporal facts (awards, spouses, positions held, etc.). This makes it particularly challenging, since spotting facts is harder than spotting events which is a specific case of named entity disambiguation task. As WikiWars articles, WikiBios articles are plain text documents that do not contain any structured elements of Wikipedia. **News articles.** We show that our methods can perform well not only on properly edited texts that are rich in terms of events and facts (i.e., WikiWars, WikiBios) but also on the news that are compiled from a large source of news channels. We used GDELT (`http://gdeltproject.org/`) news dataset for our experiments. GDELT contains a set of entities for each article; however, we ignored these annotations and solely relied on our own methods to extract and disambiguate entities. In total, this test corpus contains 1,5 million news articles.

6.2 Evaluation Tasks and Metrics

To validate each hypothesis explained above we define an evaluation task.
Detection quality. We evaluated the quality of temponym detection by checking whether a detected noun phrase is indeed a temponym. We divided this task into two separate tasks: Event detection quality, and fact detection quality. For the event detection task, we manually annotated the named events appearing in WikiWars. In total, we annotated 1,154 events. We compare our method's coverage to the state-of-the-art entity disambiguation tool AIDA. In order to make a fair comparison in favor of AIDA, we only considered the named events that are linked to particular Wikipedia event articles by Wikipedia editors. Thus, we ended up 646 named event phrases with the respective sentences that they appear in.

For the fact detection task we manually annotated the facts appearing in WikiBios dataset. We only considered the first three paragraphs of each article during annotation. We annotated 589 temporal facts. The previous works [37, 40] consider only subject-verb-object style phrases for fact extraction. Since temponyms are of the noun phrase nature, we do not compare our method's coverage to previous work. Thus, we just report the recall values.
Disambiguation quality. The evaluation of mapping of temponyms is a human intelligence task. We evaluated the quality of temponym disambiguation by checking whether a temponym is mapped to the correct event or fact in the KB. This implies that the temporal scoping for the temponym is correct, too. We additionally checked whether the mentions in the temponym context are correctly disambiguated as well. There is no prior ground-truth for these corpora and creating such a dataset is a big amount of human work. Thus, we manually judged the quality of the computed mappings. We randomly selected 100 temponyms per model per dataset. In other words, 200 temponyms from WikiWars mappings, 300 from WikiBios mappings, and 300 from News mappings, a total of 800 temponym mappings. For statistical significance, we calculated Wilson confidence intervals [7].

We ran the local model, the joint model, and the global model on each corpus with the exception of WikiWars. The

Dataset	Strict			Relaxed		
	Local	Joint	Global	Local	Joint	Global
WikiBios	.54 ±.09	.60 ±.09	.68 ±.09	.61 ±.09	.66±.09	.76±.08
WikiWars	.75 ±.08	.82 ±.07	n/a	.84±.07	.86 ±.06	n/a
News	.58 ±.09	.64 ±.09	.67 ±.09	.69 ±.09	.75 ±.08	.79 ±.08

Table 5: Precision at 95% Wilson interval for different methods.

temponym	Yago	Our model	Time scope	Eval
the Great Recession	`GreatRecession`	`GreatRecession`	`[2007, 2009]`	Correct
the second term of Merkel	–	⟨`AngelaMerkel, holdsPosition, ChancellorOfGermany`⟩	`[2005, now]`	Okay
Obama's graduation	–	⟨`BarackObama, graduatedFrom, HarvardLawSchool`⟩	`[1991, 1991]`	Correct
the first Winter Olympics to be hosted by Russia	–	`2014WinterOlympics`	`[2014,2014]`	Correct
Putin's presidency	–	⟨`VladimirPutin, holdsPosition, PrimeMinisterOfRussia`⟩	`[2008, 2012]`	Wrong

Table 6: Example of temponyms mapped by our system vs Yago.

global model is not applicable here, as it requires multiple documents on the same or overlapping topics. In contrast, the 22 WikiWars articles are fairly disjoint in their contents and are not mentioned in GDELT news corpus much.

The evaluation is done by marking a mapping with three different scores; Correct, Okay, Wrong. Table 6 shows some examples of Correct, Okay, and Wrong matches. A mapping is considered "Okay" if it has partially correct match. For example, the temponym *the second term of Merkel* is mapped to the correct fact ⟨`AngelaMerkel, holdsPosition, ChancellorOfGermany`⟩ but it is marked as "Okay". The reason is that the second term of Angela Merkel is actually from 2009 to 2013 rather than from 2005 to now.

Precision is calculated in two different ways:

- For *strict* precision, we count the *Okay* mappings as wrong:

$$Precision_{strict} = \frac{\#Correct}{\#Total\ mappings}$$

- For *relaxed* precision, we count the *Okay* mappings as true:

$$Precision_{relaxed} = \frac{\#Correct\ +\ \#Okay}{\#Total\ mappings}$$

Temporal enrichment. To show our methods can substantially add extra temporal information to documents, we compare our methods to well known HeidelTime tagger by running the both methods on WikiWars and WikiBios datasets. We compare the number of normalized TempEx's by HeidelTime tagger to the number of normalized temponyms by our methods.

Knowledge enrichment. The temponym resolution task has two important outcomes in terms of knowledge enrichment: First, temponym resolution enriches the KB by providing additional paraphrases for known events and facts. For example, our methods can add the temponym *"the largest naval battle in history"* as an alias for the event `BattleOfLeyteGulf`, or *"Obama's presidency"* as an alias for the fact `BarackObama, holdsPosition, presidentOfUS`. We add this new knowledge to the KB through "rdfs:label" triples that have a temponym phrase as subject, and an event or a fact identifier as object. We call this task *Knowledge paraphrasing.*

We assess the knowledge paraphrasing, by comparing outcome of our methods to Yago2 knowledge base in terms of paraphrase coverage. Therefore, we randomly chose 100 correctly mapped temponyms and checked how many temponyms are already known to Yago2, either as an event entity or as a fact. We built a text index over all events and facts in Yago2 and their alias names. For the randomly chosen 100 temponyms, we queried this index for each temponym and took the top-10 most relevant results for each query. We manually checked all these returned answers, thus considering also approximate matches for a fair comparison in favor of Yago2.

Second, temponym resolution also enhances the fact extraction tools for knowledge bases by providing them additional temporal and semantic clues. For example, in the sentence "Ronaldo joined Real Madrid during second term of Florentino Pèrez" a fact extraction tool can extract the fact ⟨`f1:CristianoRonaldo, playsFor, RealMadrid`⟩ but no time scope attached. Temponym resolution would normalize the phrase *second term of Florentino Pèrez* to time [2009, now] by mapping it to the fact ⟨`f2:FlorentinoPèrez, isPresidentOf, RealMadrid, [2009, now]`⟩. Thus, a fact extraction tool can temporally link two facts as a new fact ⟨`f3:f1, validDuring, f2`⟩. We call this task *Knowledge linking.*

For the knowledge linking task, we carried out an extrinsic case study. We modified the PATTY's binary fact extraction patterns to ternary patterns so that they can take a temponym as an argument. For example, the PATTY pattern ⟨*subject, verb, object*⟩ is modified to ⟨*subject, verb, object, preposition, temponym*⟩. Thus, a fact extracted from ⟨*subject, verb, object*⟩ triple can be linked to the particular temponym through a particular preposition such as "during, before, after". For this task, we ran PATTY tool on its extraction corpus. We report the number of facts that

	WikiWars	WikiBios
# Gold annotations	646	589
# AIDA's extractions	186	–
# Our extractions	338	194
AIDA's recall	29%	–
Our recall	52%	33%

Table 7: Recall values for AIDA and for our method.

are linked to temponyms through three prepositions "during, before, after".

6.3 Results

Detection quality. Our methods detected 233 265 temponyms from the three corpora. Specifically, 2 504 temponyms from WikiWars, 5 390 from WikiBios, and 225 371 from the News dataset are extracted. The recall values we calculated specifically for events in WikiWars and for facts in WikiBios datasets are shown in Table 7.

i) Event detection. Among the 646 annotated named events in WikiWars dataset, AIDA detected 186 of them, which results in 29% coverage. On the contrary, our methods detected 338 of the events, which resulted in 52% coverage. It is obvious that general ail NED tools such as AIDA are not well suited for event detection. Therefore, specialized solutions such as our methods should be pursued.

ii) Fact detection. Among the 589 annotated temporal facts in WikiBios dataset, our method detected 194 of them, which yields a 33% coverage. It might seem a low coverage. However, considering that temporal facts can be phrased in text in many different ways, our results are encouraging. Our empirical observations show that the main cause of the low coverage is the deficacy of the KB. Using a larger knowledge base may improve the results. Secondly, enlarging the pattern dictionary might have a direct impact on the coverage.

Disambiguation quality. We evaluated the overall disambiguation quality over randomly selected 800 temponym mappings. We computed 95% Wilson confidence intervals for strict precision and for relaxed precision, on all three datasets. The strict matching evaluation gives us a $65\% \pm 0.03$ precision. The relaxed matching evaluation gives us a $73\% \pm 0.03$. The detailed precision results for each dataset and for each method are shown in Table 5.

We see that the joint and global models boost the precision by a large margin. For the relaxed precision measure, the global models achieved substantial gains over the joint models. The precision numbers are particularly good for the News and the WikiWars corpora, thus achieving high value for semantic markup and knowledge enrichment. For WikiBios, the results are somewhat worse. Here we faced the challenge that many temponyms refer to SPOT facts (e.g., awards, spouses, children, held positions, etc.) rather than typed events, which is much harder to deal with. Nevertheless, the results are very encouraging, given that temponym resolution is more demanding than TempEx resolution and the state-of-the-art results for TempEx's are 80 to 90% [38].

Temporal enrichment. We compared our best performing model, global model, to HeidelTime tagger to see how much additional temporal information is added to documents. HeidelTime normalized 5 533 TempEx's from WikiBios dataset, and 2 047 from WikiWars dataset to date values. Whereas, our methods normalized 885 temponyms from WikiBios dataset, and 558 from WikiWars dataset to date values by disambiguating these temponyms to KB facts or events. Note that these temponyms are not detected by HeidelTime tagger at all. Thus, our methods add 16% additional temporal information to WikiBios dataset and 27% to WikiWars dataset.

Knowledge enrichment. For the knowledge paraphrasing task, the manual assessment over randomly selected 100 temponyms showed that Yago2 knows alias names for only 52 of the events given by the 100 temponyms. On the remaining 48, Yago2 does not even have any approximate matches. Yago2's coverage is great for canonicalized event names such as "the Great Recession", "Second World War", etc. However, it is largely agnostic to phrases for less standardized events such as "the second term of Merkel", "Obama's graduation", "the last presidential election in France", etc. Our methods do not only detect these temponyms but also disambiguate them correctly onto events or facts. Examples from this comparison are shown in Table 6.

For the knowledge linking task, our methods disambiguated 65 625 temponyms surrounding the facts that are extracted by ternary patterns. 12 803 (20%) of these temponyms are temporally linked to the extracted facts through prepositional links. For example, the base facts extracted from the sentence "Hillary was First Lady of the United States during Clinton's tenure." by this method are

$\langle$`f1:HillaryClinton, holdsPosition, FirstLadyOfUS`$\rangle$,
$\langle$`f2:BillClinton, holdsPosition, PresidentOfUS`$\rangle$.

These two base facts are linked through the reification mechanism of RDFS. Thus, `f1` and `f2` are linked as

$\langle$`f3:f1, validDuring, f2`$\rangle$.

6.4 Data and Software

The data used for our experiments is publicly available.[2] Moreover, we incorporated some of our methods into the well known temporal tagger HeidelTime. Further information how to use this new version of HeidelTime can be obtained from the same URL.

7. CONCLUSION

We have presented a viable solution for temponym resolution – an important problem for search, text analytics and KB curation that has received little attention in the literature so far. Our experiments demonstrate that we can resolve temponyms onto events or facts in a KB with fairly good precision, and that we can enrich the KB itself with additional names for known events and with newly emerging events. Our future work includes scaling our system up for processing very large text corpora, testing our methods with different knowledge bases and with a larger pattern dictionary. We expect that the semantic markup of temponyms in news articles and social media will boost next-generation deep analytics of unstructured data.

8. REFERENCES

[1] O. Alonso, M. Gertz, and R. Baeza-Yates. Enhancing Document Snippets Using Temporal Information. In *SPIRE*, 2011.

[2] `http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/evin/`

[2] O. Alonso and K. Khandelwal. Kondenzer: Exploration and Visualization of Archived Social Media. In *ICDE*, 2014.

[3] O. Alonso, J. Strötgen, R. Baeza-Yates, and M. Gertz. Temporal Information Retrieval: Challenges and Opportunities. In *TWAW*, 2011.

[4] A. Angel, N. Sarkas, N. Koudas, and D. Srivastava. Dense Subgraph Maintenance Under Streaming Edge Weight Updates for Real-time Story Identification. *Proc. VLDB Endow.*, 5(6):574–585, 2012.

[5] K. Berberich, S. J. Bedathur, O. Alonso, and G. Weikum. A Language Modeling Approach for Temporal Information Needs. In *ECIR*, 2010.

[6] S. Bethard and J. H. Martin. Identification of Event Mentions and Their Semantic Class. In *EMNLP*, 2006.

[7] L. D. Brown, T. T. Cai, and A. Dasgupta. Interval Estimation for a Binomial Proportion. *Statistical Science*, 16:101–133, 2001.

[8] D. Carmel, M.-W. Chang, E. Gabrilovich, B.-J. P. Hsu, and K. Wang. First International Workshop on Entity Recognition & Disambiguation. 2014.

[9] A. Chang and C. Manning. SUTime: A Library for Recognizing and Normalizing Time Expressions. In *LREC*, 2012.

[10] P.-T. Chang, Y.-C. Huang, C.-L. Yang, S.-D. Lin, and P.-J. Cheng. Learning-based Time-sensitive Re-ranking for Web Search. In *SIGIR*, 2012.

[11] M. Cornolti, P. Ferragina, and M. Ciaramita. A Framework for Benchmarking Entity-annotation Systems. In *WWW*, 2013.

[12] A. Das Sarma, A. Jain, and C. Yu. Dynamic Relationship and Event Discovery. In *WSDM*, 2011.

[13] Q. Do, W. Lu, and D. Roth. Joint Inference for Event Timeline Construction. In *EMNLP-CoNLL*, 2012.

[14] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[15] J. R. Finkel, T. Grenager, and C. Manning. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *ACL*, 2005.

[16] D. Gupta and K. Berberich. Identifying Time Intervals of Interest to Queries. In *CIKM*, 2014.

[17] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum. YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia. *Artif. Intell.*, 194:28–61, 2013.

[18] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum. Robust Disambiguation of Named Entities in Text. In *EMNLP*, 2011.

[19] P. Jindal and D. Roth. Extraction of Events and Temporal Expressions from Clinical Narratives. *Journal of Biomedical Information*, 46:S13–S19, 2013.

[20] N. Kanhabua, T. Ngoc Nguyen, and W. Nejdl. Learning to Detect Event-Related Queries for Web Search. In *WWW*, 2015.

[21] Z. Kozareva and E. H. Hovy. Learning Temporal Information for States and Events. In *ICSC*, 2011.

[22] E. Kuzey and G. Weikum. Extraction of Temporal Facts and Events from Wikipedia. In *TempWeb*, 2012.

[23] K. Lee, Y. Artzi, J. Dodge, and L. Zettlemoyer. Context-dependent Semantic Parsing for Time Expressions. In *ACL*, 2014.

[24] J. L. Leidner. Toponym Resolution in Text. *SIGIR Forum*, 41(2):124–126, 2007.

[25] X. Li and W. B. Croft. Time-based Language Models. In *CIKM*, 2003.

[26] M. D. Lieberman and H. Samet. Adaptive Context Features for Toponym Resolution in Streaming News. In *SIGIR*, 2012.

[27] X. Ling and D. S. Weld. Temporal Information Extraction. In *AAAI*, 2010.

[28] P. Mazur and R. Dale. WikiWars: A New Corpus for Research on Temporal Expressions. In *EMNLP*, 2010.

[29] D. Metzler, R. Jones, F. Peng, and R. Zhang. Improving Search Relevance for Implicitly Temporal Queries. In *SIGIR*, 2009.

[30] N. Nakashole, G. Weikum, and F. Suchanek. PATTY: A Taxonomy of Relational Patterns with Semantic Types. In *EMNLP-CoNLL*, 2012.

[31] J. Piskorski, J. Belayeva, and M. Atkinson. Exploring the Usefulness of Cross-lingual Information Fusion for Refining Real-time News Event Extraction: A Preliminary Study. In *RANLP*, 2011.

[32] J. Pustejovsky, K. Lee, H. Bunt, and L. Romary. ISO-TimeML: An International Standard for Semantic Annotation. In *LREC*, 2010.

[33] D. Shahaf and C. Guestrin. Connecting the Dots Between News Articles. In *KDD*, 2010.

[34] W. Shen, J. Wang, and J. Han. Entity Linking with a Knowledge Base: Issues, Techniques, and Solutions. *Trans. on Knowl. and Data Eng.*, 27(2):443–460, 2015.

[35] M. Spaniol, J. Masanès, and R. Baeza-Yates. The 4th Temporal Web Analytics Workshop. 2014.

[36] J. Strötgen and M. Gertz. Multilingual and Cross-domain Temporal Tagging. *Language Resources and Evaluation*, 47(2):269–298, 2013.

[37] P. P. Talukdar, D. Wijaya, and T. Mitchell. Coupled Temporal Scoping of Relational Facts. In *WSDM*, 2012.

[38] N. UzZaman, H. Llorens, L. Derczynski, J. F. Allen, M. Verhagen, and J. Pustejovsky. SemEval-2013 Task 1: TempEval-3: Evaluating Time Expressions, Events, and Temporal Relations. In *SemEval*, 2013.

[39] M. Verhagen and J. Pustejovsky. The TARSQI Toolkit. In *LREC*, 2012.

[40] Y. Wang, M. Dylla, M. Spaniol, and G. Weikum. Coupling Label Propagation and Constraints for Temporal Fact Extraction. In *ACL*, 2012.

[41] Y. Wang, B. Yang, L. Qu, M. Spaniol, and G. Weikum. Harvesting Facts from Textual Web Sources by Constrained Label Propagation. In *CIKM*, 2011.

[42] S. Whiting, J. Jose, and O. Alonso. Wikipedia As a Time Machine. In *WWW*, 2014.

[43] R. Yan, X. Wan, J. Otterbacher, L. Kong, X. Li, and Y. Zhang. Evolutionary Timeline Summarization: A Balanced Optimization Framework via Iterative Substitution. In *SIGIR*, 2011.

[44] X. Zhao, Y. Guo, R. Yan, Y. He, and X. Li. Timeline Generation with Social Attention. In *SIGIR*, 2013.