Coupled Temporal Scoping of Relational Facts

Partha Pratim Talukdar
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA
ppt@cs.cmu.edu

Derry Wijaya
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA
dwijaya@cs.cmu.edu

Tom Mitchell
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA
tom.mitchell@cs.cmu.edu

ABSTRACT

Recent research has made significant advances in automatically constructing knowledge bases by extracting relational facts (e.g., Bill Clinton-presidentOf-US) from large text corpora. Temporally scoping such relational facts in the knowledge base (i.e., determining that Bill Clinton-presidentOf-US is true only during the period 1993 - 2001) is an important, but relatively unexplored problem. In this paper, we propose a joint inference framework for this task, which leverages fact-specific temporal constraints, and weak supervision in the form of a few labeled examples. Our proposed framework, CoTS (Coupled Temporal Scoping), exploits temporal containment, alignment, succession, and mutual exclusion constraints among facts from within and across relations. Our contribution is multi-fold. Firstly, while most previous research has focused on micro-reading approaches for temporal scoping, we pose it in a macro-reading fashion, as a change detection in a time series of facts' features computed from a large number of documents. Secondly, to the best of our knowledge, there is no other work that has used joint inference for temporal scoping. We show that joint inference is effective compared to doing temporal scoping of individual facts independently. We conduct our experiments on large scale open-domain publicly available time-stamped datasets, such as English Gigaword Corpus and Google Books Ngrams, demonstrating CoTS's effectiveness.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; I.2.6 [**Artificial Intelligence**]: Learning

General Terms

Algorithms,Experimentation

Keywords

Temporal Scoping,Joint Inference,Knowledge Base

WSDM 2012, Seattle, Washington, USA

1. INTRODUCTION

There has been much research on extracting relational facts from both structured and unstructured text. Systems such as YAGO [21], KnowItAll [10], TextRunner [4], and NELL [7] gather entities and factual relations between entities from Web sources. However, much of the effort has been focused on gathering facts without their temporal scope. Facts are treated as time-invariant when in reality they dynamically change with time. New facts arise while others cease to be valid or change over time. Knowledge grows in various dimensions, and completely new entity types, relation types or knowledge structures may arise with time [28]. For example, the fact that Bill Clinton is a US President is only valid from the year 1993 to 2001. Such temporally scoped facts are useful for many reasons. Temporal information can be used as a dimension along which facts can be organized, ranked, or explored. Time can be helpful for relevancy ranking purposes. Presenting facts in a timeline can greatly benefit user experience in their exploration of knowledge evolution [2]. In a search or question answering system, time-sensitive queries such as business-intelligence queries (e.g., when did certain companies acquire other companies?) or medical queries (e.g., when did a certain vaccine become available?) will also benefit from temporally scoped facts. Temporal scoping of facts can also benefit other natural language applications, such as document summarization where the temporal information of facts mentioned in sentences can be used to generate better sentence ordering.

Despite the importance of time in any information space, gathering and distilling temporal knowledge from Web sources remains a major research challenge [28]. To the best of our knowledge, Timely YAGO [27] and PRAVDA [26], two recently proposed techniques, are the only systems which try to harvest temporal facts. Timely YAGO tries to automatically scope facts using regular expressions in Wikipedia infoboxes, and hence is not applicable to widely available free text. PRAVDA, a promising recent approach, uses a combination of textual patterns and graph-based re-ranking techniques to harvest facts and their temporal points at the same time. However, it is not immediately clear how this approach could be used to temporally scope facts in an existing knowledge base. Other works on temporal information extraction have tackled partial aspects of the problem such as temporal relations identification between events [6, 14, 5, 16, 8, 29, 13]. However, these other works are not sufficient to temporally scope facts as they are all focused on micro-reading of time at a single document or sentence level, i.e., temporal expressions and relations are normalized and identified based

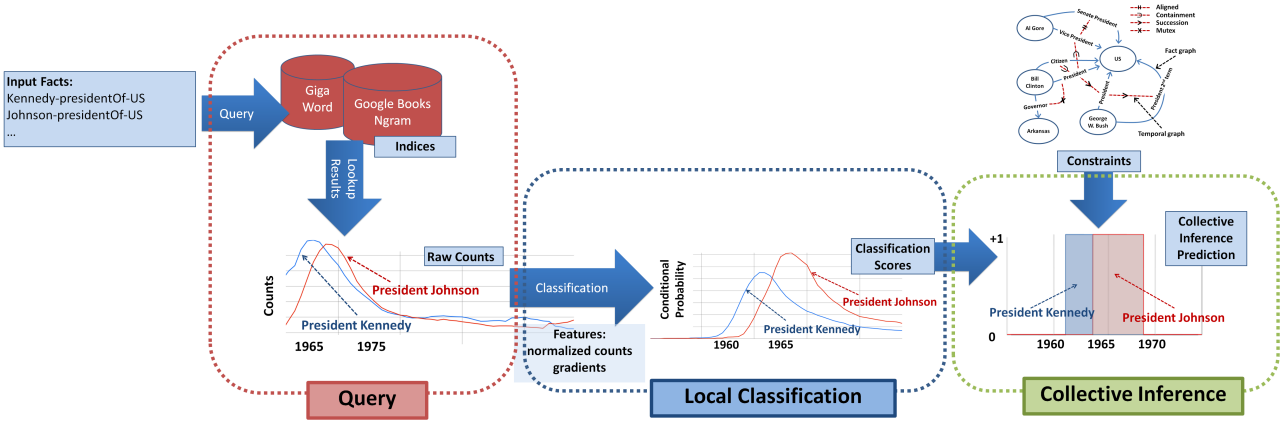

Figure 1: Architecture of the CoTS system (see Section 2 for an overview). Details of the Local Classification and Collective Inference modules are presented in Section 3.1 and Section 3.2, respectively.

only on features derived from the document. However, web is unique in that it typically offers ample redundancy, and hence it only seems natural to aggregate many observed cues for a temporal fact in a statistical manner [28]. This is what we attempt to do in our work, a macro-reading of temporal information from large-scale sources such as Google Books Ngram [15] and newswire text in Gigaword [12] using their document creation times (DCTs) to temporally scope facts.

Our contribution to this important but largely unexplored problem of temporal scoping is to propose a novel system, CoTS (Coupled Temporal Scoping), with multi-fold benefits.

- Firstly, in the spirit of macro-reading, CoTS uses a statistical approach to the problem, by using simple counts of facts in documents over time as cues to their temporal validity. Without going into document content, CoTS represents a fact by a time series of its counts over time. We believe time series is a natural way of representing a fact as it models directly the dynamic nature of the fact, its rise and fall over time. Evidence of change in time series is used as cues to classify whether the fact is active or not at any given time.

- Secondly, CoTS is novel in that we introduce collective[1] temporal inference over multiple temporally correlated facts to aggregate many observed cues for improved time scoping. Independent activation scores of facts are input to a collective inference framework, and Integer Linear Programming (ILP) is used to infer the temporal scope of facts while respecting various temporal dependencies among those facts, such containment, alignment, succession, and mutual exclusion.

- Thirdly, CoTS is weakly supervised: we need only a few labeled examples to train a local classifier (one for each relation), the only supervised component in CoTS. Moreover, CoTS provides a flexible framework where prior knowledge about the temporal dependencies among facts can be easily specified.

[1]In this paper, we shall use the terms collective inference, coupled inference, and joint inference interchangeably.

- Lastly, through experiments on interesting open-domain datasets such as the Google Books Ngram Corpus and the Gigaword Newswire Corpus, we demonstrate CoTS' effectiveness in improved temporal scoping, highlighting benefits of scoping multiple temporally related facts jointly, rather than scoping each such fact in isolation.

2. CoTS OVERVIEW

2.1 Challenges & Motivation

The problem of macro-reading the temporal scope of a fact from its counts in documents is a difficult one as these counts can be noisy, lagging in time, or sparse. A fact may still be found in documents (i.e., its count is not zero) even after it ceases to be valid. For example, some documents discuss the presidency of Kennedy even after his death. The document creation time may also lag behind a fact's activation time. For example, books about President Clinton may only be published some months after his inauguration (since books often take longer time to print and publish than news). Some facts are not mentioned enough in documents, leading to the sparsity of their counts. For example, unlike the President relation, the US Secretary of State relation may not be mentioned as frequently in documents.

These challenges of the problem motivate our approach. Firstly, to deal with the issue of time lag, especially in books data, we use also the change in the counts over time to capture the activation of a fact. A positive gradient in the time series of a fact's counts indicates that the fact is increasingly being talked about in documents, which may signal its activation time. Using positive gradients as additional information can capture change points based on the moment they start to be increasingly mentioned in data.

Secondly, to deal with the noise and sparsity of counts, we conduct collective inference to temporally scope several facts jointly. The advantage of doing collective inference is multi-fold. By utilizing temporal constraints between correlated facts, collective inference can bind the temporal scope of noisy facts by ensuring temporal consistency between facts, or inform the time scope of a sparse fact from the time scope of other, temporally correlated facts.

For example, in a functional relation (e.g., US presiden-

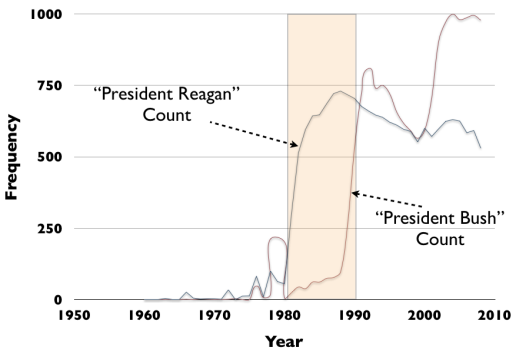

Figure 2: Temporal profiles of two facts demonstrating potential benefit of collective inference: considering 'President Bush' during inference can help determine the end of Reagan's presidency (shaded region).

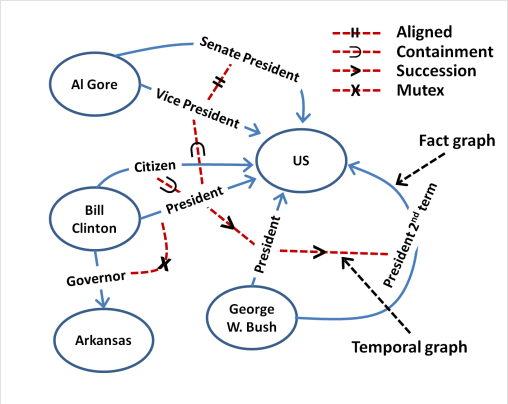

Figure 3: Temporal graph (dotted red edges) imposed over a factual graph (blue solid edges). Edges in the temporal graph correspond to constraints in CoTS, the proposed system.

tial relation), knowing the time scope of one instance of the relation can bind the time scope of another instance of the same relation since no two instances of a functional relation can be true at the same time (i.e., no two persons can be US president at the same time). This constraint is great to bind noisy, frequently mentioned facts. For example, although documents may still refer to Kennedy as 'President Kennedy' even after his death, knowing that Johnson's presidency began in 1963 and that his presidency succeeded Kennedy's can help bind the time scope of Kennedy's presidency. For example, in Figure 2, we can see how knowing the start of Bush's presidency can bind the end of Reagan's presidency, even though 'President Reagan' continues to be mentioned in documents even after his presidency.

Conversely, for facts that are sparse (e.g., those belonging to the US vice president relation), knowing the time scope of a correlated fact can help infer the time scope of the sparse fact. For example, the time scope of Clinton's presidency can be used to infer the time scope of Gore's vice presidency knowing that Gore served under Clinton's presidency.

Allowing users to specify temporal constraints between facts is also a natural way of adding prior knowledge to the task of temporal scoping. In Figure 3, we illustrate various temporal constraints that can be specified for US administration relations. Solid lines indicate factual relations (entity-relation-entity triplets) while dashed lines indicate temporal constraints between these facts. As we can see from Figure 3, temporal alignment between VicePresident and SenatePresident relations indicates that Al Gore must be vice president of the US at the same time that he is the president of US senate. Temporal mutual exclusion between President and Governor relations indicates that Bill Clinton cannot be both President of the US and Governor of Arkansas at the same time. Temporal containment between Citizen and President relations indicates that Bill Clinton must be a President of the US within the period that he is a citizen of the US. Temporal succession between facts indicates that the time scope of Bush-presidentOf-US follows the time scope of Clinton-presidentOf-US.

2.2 System Architecture

Figure 1 summarizes the high level architecture of our work to temporally scope facts, which involves the following pipeline of operations:

1. **Query**: To macro-read the temporal scope of a fact, we first construct a query to represent the fact. For example, the fact: 'Kennedy-presidentOf-US' is represented by the query 'President Kennedy'. Then we do a lookup of the query on the indices we have built for Google Books Ngram and Gigaword datasets. We gather raw counts of the results: i.e., the number of n-grams containing the query (for Google Books) and the number of times a query is found in the news documents (for GigaWord). Each fact is thus represented by a time series of its query counts, normalized over its total query counts.

2. **Local Classification**: Normalized counts and gradients of each fact are then input to a Maximum Entropy classifier which computes the conditional probability that the fact s from relation r is active at a given time t, i.e., $p_r(+1|s,t)$. As discussed previously, gradient information is included as a feature in this classification to capture the informative signal of increasing counts which may indicate the start of the fact's activation. Using only normalized counts and where they peak to classify start time may not be sufficient, as the peak may not always coincide with the start time due to possible time difference (lag) between the actual start time and the document creation time.

3. **Collective Inference**: Individual classification scores of the facts, $p_r(+1|s,t)$ and $p_r(-1|s,t)$, are then input to a collective inference engine (in our case, an Integer Linear Program (ILP)) together with the temporal constraints we have described and illustrated in Figure 3. ILP then predicts which facts are active at which times, based on the input classification scores and the specified temporal constraints. The outputs are then the facts and the times for which they are active.

3. SYSTEM DESCRIPTION

In this section, we present detailed descriptions of the three main modules in the CoTS system.

Notations

- Let $\{S^1, \ldots, S^m\}$ be sets of facts, one set for each of the m relations. We would like to temporally scope each fact $s \in S^r$, $\forall 1 \leq r \leq m$. In this paper, we shall use the relation index, r, also to refer to the relation name itself.
- Let B and E be the beginning and end of the timespan over which the temporal scoping is currently performed. We shall assume that any time instant $t \in \{B, \ldots, E\}$ is discretized at an appropriate granularity, which for the experiments in this paper is at the year level.
- $p_r(+1|s,t)$ is the local classification score specific to relation r, which measures whether the fact $s \in S^r$ is true at time t. Please refer to Section 3.1 for a discussion on estimating such scores.
- $x_{r,s,t} \in \{0,1\}$ is a binary integer variable, where $x_{r,s,t} = 1$ indicates that fact $s \in S^r$ is true at time $t \in \{B \ldots E\}$, and false otherwise.
- $z^b_{r,s,t} \in \{0,1\}$ is a binary integer variable, where $z^b_{r,s,t} = 1$ indicates that fact s started to be true at time t.
- Similarly, $z^e_{r,s,t} \in \{0,1\}$ is a binary integer variable, where $z^e_{r,s,t} = 1$ indicates that fact s ceased to be true at time t.

3.1 Local Classification Scores

In this section, we describe how we estimate the initial score, $p_r(+1|s,t) \in [0,1]$, which measures the probability of fact $s \in S^r$ being true at time t. We obtain these initial scores by training a Maximum Entropy (MaxEnt) classifier, separately for each relation r. For each relation r, we assume that we are given a set of training instances $T_r = \{(\phi_r(s,t,y), y)\}$, where $\phi_r(s,t,y) \in R^d$ is the d-dimensional representation of fact $s \in S^r$ at time t, and $y \in \{-1,+1\}$ is its label, with $y = 1$ suggesting that the fact is true at time t, and -1 otherwise.

For each relation r, these labeled instances are then used to estimate parameters $\mathbf{w}_r$ of corresponding MaxEnt classifier. The classifier can then be applied to classify an unlabeled instance u at time t using Equation 1.

$$p_r(+1|u,t) = \frac{\exp(\mathbf{w}_r^T \cdot \phi_r(u,t,+1))}{\sum_y \exp(\mathbf{w}_r^T \cdot \phi_r(u,t,y))} \quad (1)$$

This score gives an estimate of how likely it is that fact u from relation r is true at time t. We note that $p_r(+1|u,t)$ is a probability distribution and hence it is bounded in $[0,1]$. We use both positive and negative real numbers as the values for our features. The actual features that we use for the experiments in this paper are described below.

Features:

Using the Query module described in Section 2.2, we obtain a timeline L_u for each fact u, where $L_u(t)$ returns the (normalized) count of fact u at time t. We use $L_u(t)$ as the **count feature**. We also compute the gradient over L_u at t, and use the gradient value as the **gradient feature**. This feature helps in adapting the classifier to adapt to time-lag issues as described in Section 2.1. We use only these two features in the local classifier. Since the local classifier estimates its parameters from a very small number of labeled instances, we wanted to make sure the the local classifier doesn't overfit severely which is possible in the presence of large numbers of features.

3.2 Collective Inference

In our system, time scoping each fact in isolation corresponds to making the final scoping decisions based purely on the scores $p_r(+1|u,t)$ as described in previous section. While such scores provide useful discriminating information, they are often noisy, leading to incorrect time scoping decisions. Lack of discriminating features, paucity of labeled training data, and inherent hardness of the classification task itself could be one of several reasons leading to these noisy predictions. While obtaining large amounts of labeled data may not be practical, incremental improvements may be possible by employing more sophisticated features. We take a different approach and as motivated in previous sections, we introduce here a way to use these prediction scores, and at the same time exploit available domain-specific constraints. To this effect, in this section, we present the Integer Linear Program (ILP)-based collective inference module of our system. First, we present the constraints that can be specified in our system, and then present the objective which is ultimately optimized subject to the constraints.

3.2.1 Constraints

Constraints in the CoTS system can be categorized into the following two classes:

1. **Intra Relation Constraints**: These constraints regulate the temporal scoping of one or more facts from a single relation. For example, Functional constraints (described below) belong to this category, which can enforce the requirement that at most one fact from the relation can be true at any given time. For example, there is only one US President at any given time. Please note that such requirements can be enforced only if multiple facts from the same relation are considered jointly during temporal scoping, which is strictly not possible in case of temporal scoping of each fact in isolation.
2. **Cross Relation Constraints**: Constraints in this category couple temporal scoping of facts from multiple relations. For example, we may want to enforce the requirement that Al Gore's Vice Presidency aligned exactly with Bill Clinton's Presidency. Aligned constraints (described below) operating over the Vice President and President relations can be used to enforce such requirement.

Please note that all our constraints are specified at the fact level, even though they may affect one or more facts from the same or multiple relations. Below, we present examples of different constraints exploited by the CoTS system for the experiments in this paper.

- Consistency: The following constraints make sure that the begin ($z^b_{r,s,t}$) and end ($z^e_{r,s,t}$) variables are consistent with the $x_{r,s,t}$ variables.
  - BeginConsistency1: For a fact to start at a given time, the fact should also be true at that time.

$$z^b_{r,s,t} \leq x_{r,s,t},\ \forall s \in S^r,\ t \in \{B, \ldots, E\}$$

– BeginConsistency2: For a fact to begin at time t, the fact should not be true at time $t-1$ and become true at t. We use the boundary condition $x_{r,s,(B-1)} = 0$.

$$z^b_{r,s,t} \geq x_{r,s,t} - x_{r,s,(t-1)},\ \forall s, t \in \{B \ldots E\}$$

– EndConsistency1: For a fact to end at a given time, the fact should also be true until that time.

$$z^e_{r,s,t} \leq x_{r,s,t},\ \forall s \in S^r, \text{and } t$$

– EndConsistency2: For a fact to end at time t, the fact should be true at time t and not true at $t+1$. We use the boundary condition $x_{r,s,(E+1)} = 0$.

$$z^e_{r,s,t} \geq x_{r,s,t} - x_{r,s,(t+1)},\ \forall t \in \{B \ldots E\}$$

- Functional: For a given relation r, these constraints enforce the requirement that no two facts from r be true at the same time.

$$\textstyle\sum_s x_{r,s,t} \leq 1,\ \ \forall t \in B \ldots E$$

As an example, Functional constraints can be used to enforce the fact that there can be at most one Vice President in USA at any given time. Functional constraints are Intra Relation in nature.

- SingleSpan: These constraints make sure that any fact from a relation r is true continuously for a single span of time, without any interruption in between. For example, US presidencies tend to be a single continuous span of time. SingleSpan, another member of the Intra Relation constraint class, is actually a set of constraints, which we describe below. Below, we shall assume that the fact belongs to relations r.

  – SingleBegin: For each fact, there is at most one beginning. This is "at most" and not "equal" as the fact may not be activated during the time interval $B \ldots E$.

  $$\textstyle\sum_t z^b_{r,s,t} \leq 1,\ \forall s \in S^r$$

  – SingleEnd: For each facts, there is at most one end.

  $$\textstyle\sum_t z^e_{r,s,t} \leq 1,\ \forall s \in S^r$$

  – EndAfterBegin: End of the fact should happen after its beginning.

  $$\textstyle\sum_t t * z^e_{r,s,t} - \sum_t t * z^b_{r,s,t} \geq 0,\ \forall s \in S^r$$

- Point: This constraint is useful when the fact is true only at one instant of time, i.e., the temporal span has unit length. For example, even though Steven Spielberg has multiple Academy Award for Best Director, length of temporal scope of each win is of unit length, at the year granularity.

$$\textstyle\sum_t x_{r,u,t} \leq 1,\ \forall u \in S^r$$

Please note that this is different from the Functional constraint as the sum here is over all $t \in \{B \ldots E\}$, as opposed to all facts s in case of Functional. Point constraints are Intra Class.

- Aligned: This constraint is useful whenever two facts (from same or different relations) have exactly same temporal span. For example, George H.W. Bush was the Vice President throughout Ronald Reagan's presidency. In other words, if facts $u \in S^a$ and $v \in S^c$ are temporally aligned, then $x_{a,u,t} = x_{c,v,t},\ \forall t$. We enforce this through two inequality constraints.

$$x_{a,u,t} \leq x_{c,v,t}, \text{ and } x_{c,v,t} \leq x_{a,u,t},\ \forall t \in \{B \ldots E\}$$

- Containment: Suppose, we want to express the fact that Al Gore's Vice Presidency was contained within Bill Clinton's Presidency, even though we don't exactly know the time span of either fact. The Containment temporal constraint can be used to achieve this. If the timespan of fact $u \in S^a$ is contained within the time span of fact $v \in S^c$, then we the Containment constraint is of the form:

$$x_{a,u,t} \leq x_{c,v,t},\ \forall t \in \{B \ldots E\}$$

We can verify that this constraint will be violated only when $x_{a,u,t} = 1$ and $x_{c,v,t} = 0$, the only undesirable case. Containment can be both Intra Relation as well as Cross Relation.

- Succession: This constraint is useful when we want to express the requirement that one fact ($v \in S^c$) from relation c happened after another fact ($u \in S^a$) from relation a, e.g., Ronald Reagan became president after Henry Kissinger was the Secretary of State.

$$\textstyle\sum_t t * z^e_{a,u,t} - \sum_t t * z^b_{c,v,t} \leq 0$$

Please note that the above constraint is effective only in conjunction with SingleSpan constraint.

- Mutex: Suppose, we want to enforce the requirement that George H. W. Bush can't be US President and Vice President at the same time. This can be achieved through the Mutex constraint. This constraint ensures that two facts, $u \in S^a$ and $v \in S^c$, are not true at the same time.

$$x_{a,u,t} + x_{c,v,t} \leq 1,\ \forall t \in \{B \ldots E\}$$

Mutex can be both Intra ($a = c$) as well as Cross Relational ($a \neq c$).

Please note that choice of constraints finally used in any given inference is dependent on the type of relation(s) involved. Moreover, the temporal constraints presented above are not meant to be exhaustive, as depending on the domain and type of relation(s), new constrains may have to introduced. We hope that CoTS' linear constraint specification is flexible enough to support a majority of such extensions.

3.2.2 Objective

Subject to the constraints mentioned above, CoTS optimizes the following objective,

$$\max_{\{x_{r,s,t}\}} \sum_{r,s,t} p_r(+1|s,t) * x_{r,s,t} + \lambda * p_r(-1|s,t) * (1 - x_{r,s,t})$$

where $\lambda \in [0, 1]$ is the tradeoff weight which controls the relative importance of the two terms in the objective. The first

term in the objective encourages the optimization to respect the classification scores $p_r(+1|s,t)$; while the second term encourages the inference to abstain from over-predicting in case the local MaxEnt classifier is not confident enough. This become more apparent after the following analysis.

$$\begin{aligned} & p_r(+1|s,t) * x_{r,s,t} + \lambda * p_r(-1|s,t) * (1 - x_{r,s,t}) \\ = \;& p_r(+1|s,t) * x_{r,s,t} + \lambda * (1 - p_r(+1|s,t) * (1 - x_{r,s,t}) \\ = \;& ((1+\lambda) * p_r(+1|s,t) - \lambda) * x_{r,s,t} + \lambda * (1 - p_r(+1|s,t)) \\ = \;& ((1+\lambda) * p_r(+1|s,t) - \lambda) * x_{r,s,t} + \text{Constant} \end{aligned}$$

The optimization can ignore the second constant term as it is not dependent on $x_{r,s,t}$. Since the objective is one of maximization sense, there is incentive to set $x_{r,s,t} = 1$ (subject to constraint satisfaction) iff

$$\begin{aligned} (1+\lambda) * p_r(+1|s,t) - \lambda &> 0 \\ p_r(+1|s,t) &> \frac{\lambda}{1+\lambda} \end{aligned}$$

With $\lambda = 1$, we observe that $x_{r,s,t} = 1$ is a candidate for prediction if $p_r(+1|s,t) > 0.5$, i.e., $p_r(+1|s,t) > p_r(-1|s,t)$. By varying λ, we can control how much confidence collective inference rests on the local classification scores. It is interesting to note that all these fall out naturally from the formulation of the objective. This built-in mechanism against over-prediction is a desirable property of CoTS, which is lacking in the other systems we compare against, and as we shall see in Section 4, it results in better temporal scoping.

4. EXPERIMENTS

In this section, we investigate the following:

- Does adding gradient-based features in the local classifier lead to improved temporal scoping? (Section 4.2)
- Does coupled temporal scoping (i.e., joint inference involving multiple instances from same as well different relations) help improve performance? (Section 4.3)
- What are the effects of different types of constraints on CoTS's performance? (Section 4.4)
- Is CoTS's Collective inference module fast enough? (Section 4.5)

4.1 Experimental Setup

4.1.1 Relations and Facts

Relation names and the number of facts from these relations which were used in the experiments in this section are presented in Table 1. Please note that CoTS assumes that these facts have already been extracted, and it instead focuses only on temporally scoping them. Facts from the US Administration domain (i.e., facts from the relations US President, US Vice President, and US Secretary of State) were temporally scoped together, while facts from the Academy Awards domain (i.e., Best Director and Best Movie) were scoped jointly. True temporal scope of each fact was determined manually, which in turn was used for training and evaluation purposes. Facts from the US Administration domain spanned the period 1961 - 2008, while facts from the Academy Awards domain spanned the period 1995 - 2008.

Relation Name	Number of Facts
US President	9
US Vice President	12
US Secretary of State	27
Best Director	14
Best Movie	14

Table 1: Relations and the number of facts from these relations used in the experiments in Section 4.

4.1.2 Query Module

Queries: As described in Section 2, in order to get the temporal profile of a fact (see Figure 2), we first construct a query to represent the fact. In the following description, we shall use the notation {q} to denote a query, whose keywords and operators are stored in q as per Lucene's query syntax [11]. For the three US Administration office relations in Table 1, we use the query template: Office LastName. So, the fact Bill Clinton-presidentOf-US is represented by the query {"president clinton"}. Similarly, {"vice president gore"}, and {"secretary albright"}. For the Academy Awards Best Movie relation, we used the query template {"academy award" AND movieName}, which resulted in queries of the form {"academy award" AND "a beautiful mind"}. The goal behind using this query is to retrieve documents containing both phrases: "academy award" and "a beautiful mind". Similarly, for the Academy Award Best Directory relation, the query template {"academy award" AND directorLastName} was used, which resulted in queries of the form {"academy award" AND "spielberg"}.

Time-Stamped Corpus: Once a query is generated from a fact as described above, we construct a temporal profile of the fact from the document creation times (DCTs) of documents retrieved by its query. In order to retrieve such documents for a given query, we use Lucene [11] to index time-stamped documents and look the query up against that index. We use the following two sources of time-stamped documents:

- **Google Books Ngram**[2] [15] is a corpus of about 5 million digitized books. The resulting corpus contains over 500 billion words in several languages. The data is released in the form of n-gram (n = 1 to 5) and three different counts of each of these n-grams are available per year: (1) the number of times the n-gram is found in books published in the year; (2) the number of pages that contain the n-gram in books published in the year; and (3) the number of books published in the year that contain the n-gram. An interesting feature of this data set is its extended time coverage spanning from 1500s to 2008.
- **Gigaword**[3] [12] contains English newswire text data acquired from four international newswire services, spanning the period from 1994 to 2008. Unlike Google Books Ngrams, Gigaword contains full time-stamped newswire documents and not just ngrams extracted from them. However, compared to the Google Books Ngrams dataset, it is much smaller in size and has smaller time coverage.

[2]http://books.google.com/ngrams/datasets
[3]LDC Gigaword: http://bit.ly/vZFoJ6

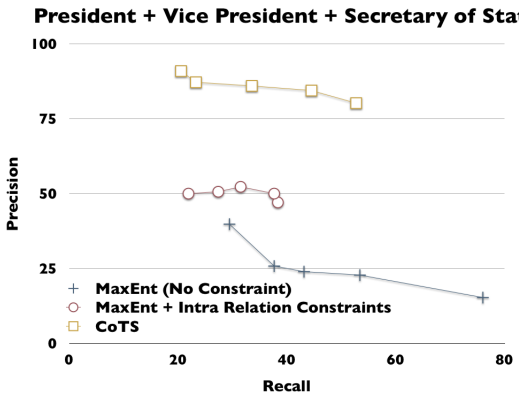

Figure 4: Precision-Recall plot for temporal scoping of US President, Vice President and Secretary of State relations. CoTS is the system proposed in this paper. (see Section 4.3)

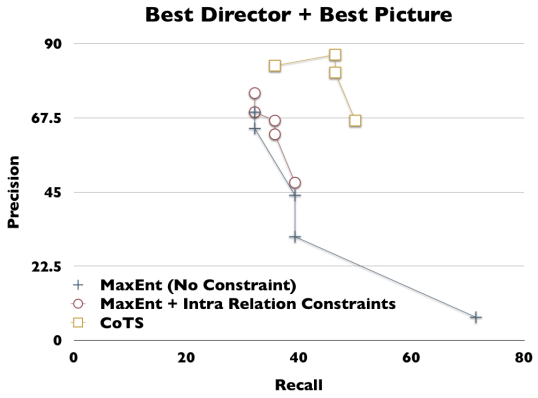

Figure 5: Precision-Recall plot for temporal scoping of Academy Award Best Director and Best Picture relations. CoTS is the system proposed in this paper. (see Section 4.3)

In case of Google Books Ngram dataset, each (unique) ngram is considered a document. Per year data from these two datasets are (separately) indexed using Lucene as mentioned above. For Google Books Ngrams dataset, we index only the English 5-grams published during the period 1960 - 2008. We consider 5-grams as they can accommodate longer queries. We use the Google Books Ngram dataset for the three US Administration relations in Table 1, as temporal scope of facts from these three relations span 1961 - 2008, which is beyond the temporal coverage of Gigaword (1994 - 2008). We use the Gigaword corpus for the two Academy Awards relations in Table 1, as the combined temporal span of facts from these two relations (as considered in this section) are aligned with that of Gigaword.

4.1.3 *Training Local Classifier*

Once the temporal profiles of facts are generated as described above, we next train a relation-specific local classifier by deriving features from these temporal profiles (Section 3.1). For each relation experimented with in this paper (Table 1), we use a single temporally scoped fact from this relation to generate all training data needed to train the corresponding relation-specific local classifier (MaxEnt, see Section 3.1). For example, for the US President relation, we derive training data from the temporally scoped fact: President Kennedy (1961 - 1963). In this case, all instances in the span [1961,1963] correspond to positive instances ($y = +1$), while everything outside this range correspond to negative instances ($y = +1$). Please note that all the classifiers in these experiments are trained from such limited amount of training data, demonstrating the real-world applicability of CoTS, the proposed method.

4.1.4 *Metrics*

As mentioned in Section 4.1.1, true temporal scopes of all facts in Table 1 were determined by a human annotator. Prediction from CoTS (or any of the baselines) that a fact is true at a given time is matched against this gold-standard to determine prediction correctness. Based on this, Precision (P), Recall (R), and F1 ($\frac{2*P*R}{(P+R)}$) are computed, which are used as the final evaluation metrics. Also, all evaluations are performed at the year level, as that is the finest granularity common to the two time-stamped datasets (Section 4.1.2).

In order to generate different Precision-Recall plots, we vary the λ parameter over the local classifier's scores, as described in Section 3.2.2.

4.2 Effect of Gradient Feature

Relations	F1 (No Gradient)	F1 (With Gradient)
President, V. President, Sec. of State	42.8	**63.63**
Best Director, Best Picture	4.26	**60.47**

Table 2: Effect of using gradient information during temporal scoping of different relations from two domains.

In this section, we evaluate the effect of gradient-based features in the local classifier (Section 3.1) can have on collective inference. We use the full CoTS with and without gradient features in the local classifier, and apply it on relations from two domains. F1 results are presented in Table 2. We observe that gradient features can significantly improve temporal scoping performance. This validates our motivation behind using gradient-based features in the local classifier, as it can help get rid of the publication lag issues in the books data and for certain domains in the newswire.

Based on these results, in all subsequent experiments, we include gradient-based features in CoTS's local classifier.

4.3 Effect of Coupling Constraints

In this section, we evaluate the effect of Cross Relational coupling constraints on temporal scoping. We compare the performance of CoTS against other systems exploiting either no constraint (MaxEnt, the local classifier in Section 3.1), or a subset of constraints (Intra Relation, Section 3.2.1) but without any cross relation coupling constraints. Examples of a few manually-specified coupling constraints used by CoTS for these experiments are presented in Table 3.

President, Vice President, Secretary of State		
Fact	Temporal Constraint	Fact
President Clinton	Containment	Vice President Gore
President Reagan	Containment	Vice President Bush
President Reagan	Succession	Secretary Kissinger
Best Director, Best Picture		
Director Cameron	Aligned	Titanic
Director Howard	Aligned	A Beautiful Mind

Table 3: Examples of coupling constraints used by CoTS for the experiments in Section 4.3. For brevity, we represent the facts by the queries used to compute their counts from the timestamped datasets (Section 4.1.2).

The Precision-Recall plots for these comparisons over facts from the two sets of relations in Table 1 are reported in Figure 4 and Figure 5, respectively. From these plots, we observe that CoTS, which exploits cross relational coupling constraints, significantly outperforms other baselines which either don't use any constraint (MaxEnt), or use only a subset of constraints exploited by CoTS.

4.4 Constraint Ablation Study

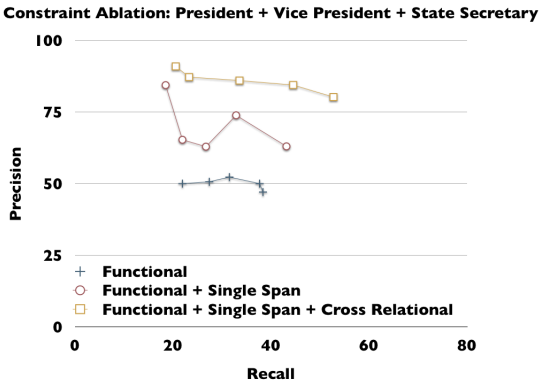

Figure 6: Constraint ablation results for temporal scoping over US President, Vice President and Secretary of State relations.

In this section, we report results from constraint ablation studies involving CoTS. The goal is to study the effect of different constraint types on CoTS's performance. Results from two different sets of relations are reported in Figure 6 and Figure 7. For the politics domain (President, Vice President, Sec. of State), we use the SingleSpan constraint as facts in that domain tend to be true for a contiguous single span (e.g., presidency usually lasts multiple consecutive years, and one is president for a single span). This is in contrast to the temporal spans of the two relations in the movies domain whose spans are usually a unit length, and where one could win the same award multiple times in one's lifetime, and hence the choice of the Point constraint.

From the results in Figure 6 and Figure 7, we observe that CoTS's performance improves as more prior knowledge is injected through additional coupling constraints. This

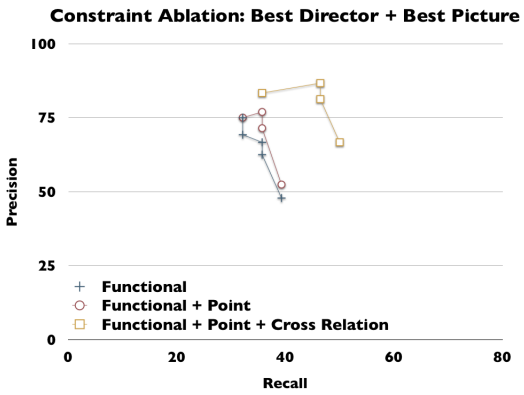

Figure 7: Constraint ablation results for temporal scoping over Academy Award winner Best Director and Best Picture relations.

justifies our design choice of using coupled (joint) inference for more accurate temporal scoping.

4.5 Running time of ILP

For all the problems in our experiments, CoTS's collective inference took on average 0.25 sec, with inference over the largest problem involving 2334 variables and 3804 constraints taking only 2.6 sec. This demonstrates the practicality of CoTS's inference scheme.

5. RELATED WORK

Even though time is an important dimension in any information space, and knowledge of temporal scopes of facts can be useful for better information retrieval systems, and user experience [2], temporal scoping of knowledge base facts is an area that is still largely unexplored. Only recently, a few research papers have started to address this problem [13, 27, 26]. Otherwise, most previous work on temporal information extraction has been focused largely on temporal representations [17], temporal relation identification [6, 14, 5, 16], and ensuring temporal consistency between extracted temporal relations [8, 29, 13]. We shall review some of the more recently proposed (and more relevant) approaches first, and then compare CoTS against the other previous work on temporal relation identification.

Timely YAGO [27] is similar in spirit to CoTS in that its objective is also temporal grounding of facts. As in CoTS, a fact in Timely YAGO refers to an instance of a binary relation between entities (entity-relation-entity triplet). Facts and their temporal information in Timely YAGO are extracted from semi-structured text in the infoboxes and category information in Wikipedia using regular expression matching. In contrast, CoTS is applicable more widely and is not limited to Wikipedia texts alone. Moreover, instead of using any regular expression based extractors, CoTS aggregates document-level metadata (viz., document creation time) based evidences to temporally scope multiple temporally related facts at the same time.

A method to temporally scope facts and reason over such scoped facts is presented in [25]. Given a target fact, the method in [25] attempts to gather count-based evidence for the begin, active and end time of this fact. These possi-

bly inconsistent evidences are then aggregated using a set of heuristics to determine the final interval over which the target fact is likely to be true. Instead of temporally scoping each fact in isolation as in [25], CoTS performs joint inference to temporally scope multiple facts at the same time while exploiting temporal dependencies among those facts, resulting in more accurate temporal scoping as demonstrated through the experiments in this paper.

The Temporal Information Extraction (TIE) system [13] attempts to output a maximal set of events and their temporal relations as directly implied in a given sentence. TIE uses joint transitivity inference to bind the start and end times for each event. As in TIE, and instead of Allen-style intervals [1], CoTS uses a real-valued point-based temporal representation. While TIE is a micro-reading system which processes a single (or a set of) sentences at once, CoTS works at the macro-reading level, aggregating evidences from a large number of documents to temporally scope a set of facts.

PRAVDA is a recently proposed method to harvest temporal facts from free text [26]. PRAVDA uses textual patterns to generate candidate temporal facts, which are then re-ranked using a graph-based label propagation algorithm adapted from a recently proposed graph transduction algorithm [22]. PRAVDA is probably the prior work which is closest in spirit to CoTS. However, unlike PRAVDA, CoTS exploits temporal dependencies among facts to scope them jointly, a strategy that we have found to be quite effective as demonstrated in Section 4.

We note that Timely Yago [27], TIE [13], and PRAVDA [26] are complimentary to CoTS, as extractions from these three systems can be used as additional evidence (features) in CoTS's local classifier, with potential for more accurate temporal scoping.

Learning and inference in CoTS is similar in spirit to the CCM framework [9, 20]. However, unlike existing CCM models, CoTS uses additional variables to define more expressive constraints, and applies them to the novel problem of temporally scoping relation facts, which is beyond the scope of current CCM-based models. Similar to CoTS, the SCAD system [3] also uses ILP to improve on the predictions from a weakly-supervised local classifier. However, the two systems address very different problems with completely different motivations.

Previous works on temporal relation identification have been largely spurred by the release of TimeML [17], a notable markup language for events and temporal expressions in natural language, the availability of tools such as TARSQI [24] to automatically annotate events and times in TimeML, and the release of TimeBank [18], a TimeML annotated corpora for training and testing. The TimeBank corpus consists of 186 news articles that are annotated for events, time expressions and temporal relations between events and times. Most previous work on temporal relation identification is based on the TimeBank corpus [6, 14, 8]. These approaches build temporal relation classifier over the corpus relations and making either local pairwise decisions of temporal relations between events [6, 14], or jointly informed decision to impose transitivity constraints (A before B and B before C implies A before C) between locally discovered temporal relations [8]. Aside from transitivity constraint, temporal expression normalization (e.g., "last year" is before "last month") are also used to discover implicit time-time relation in the document to enrich temporal network between events and times in TimeBank.

Other works on temporal relation identification [5, 16, 29] have been conducted for the tasks of TempEval Challenge [23]. TempEval challenge uses training and test sets encoded in a subset of TimeML, and is divided into three subtasks: to identify temporal relations 1) between a specific event and a time expression in the same sentence, 2) between a specific event and the Document Creation Time (DCT), and 3) between the provided main events in two adjacent sentences. The event is specified per sentence by the main verb in the sentence. Some approaches use relation classification on each subtask independently by using either a pure machine learning approach [5], or a combination of rule based and machine learning [16]. Another approach attempts to solve the three subtasks jointly [29] by learning a single probabilistic model for all three tasks, incorporating formulas of temporal transitivity that should hold across tasks.

Please note that there are fundamental differences between these temporal relation identification approaches and CoTS. Firstly, an event in these approaches is usually a tensed verb, which is different from the notion of relational facts in CoTS. Moreover, unlike these previous approaches, CoTS attempts to temporally ground facts on the timeline, and not just infer temporal dependencies among events (and time). However, these approaches could be complimentary to CoTS in the sense that the temporal relations inferred by these approaches could potentially be used as additional constraints in CoTS's collective inference engine.

6. CONCLUSION

Despite the benefits of temporally scoped facts in knowledge bases, temporal scoping of facts is a research area that has been largely unexplored, with the exception of a few recent proposals [27, 26]. In this paper we propose CoTS (Coupled Temporal Scoping), a novel way of temporally scoping facts by exploiting a variety of signals: counts of mentions of the fact in large open domain data sources such as Google Books Ngram corpus and Gigaword Corpus; and by exploiting temporal relationships between the fact and other facts from the same or different relations.

CoTS poses the task of temporal scoping as a change detection in the temporal profile of the fact. To aggregate redundant observed cues (e.g., gradients and counts from the temporal profile) for a fact, CoTS uses Integer Linear Program (ILP) to jointly infer the time scopes of multiple temporally related facts while respecting any temporal constraints among them. CoTS is a weakly supervised system in that it only needs a few labeled examples per relation to make decision on the temporal scope of facts.

There are several avenues for improving CoTS, and overcoming fact count sparseness is one of them. Some queries such as those related to the 'defenseSecretaryOf' relation return very few results in both Google Books and Gigaword datasets. Although temporal scope of sparse facts can be inferred from the temporal scope of other related facts via coupling, sparsity of counts may still hamper temporal scoping of facts that have little or no other correlated facts. To overcome this problem, we are planning to (1) derive counts from a larger number of time-stamped sources; and (2) use semi-supervised learning algorithms to gather fact counts even from non time-stamped documents.

Furthermore, in the current CoTS system, temporal con-

straints among facts are specified manually based on users' prior knowledge. In the future, we would like to automatically acquire such relation and fact specific constraints, for example, by learning temporal relationships among facts in a semi-supervised fashion, such as in NELL [7].

Acknowledgments

This research has been supported in part by DARPA (under contract number FA8750-09-C-0179), Google, and Fulbright. Any opinions, findings, conclusions and recommendations expressed in this paper are the authors' and do not necessarily reflect those of the sponsors. We are thankful to Lei Li (CMU) for useful discussions, and to the four anonymous reviewers for their constructive comments.

7. REFERENCES

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

[2] O. Alonso, M. Gertz, and R. Baeza-Yates. On the value of temporal information in information retrieval. In *ACM SIGIR Forum*, volume 41. ACM, 2007.

[3] A. Bakalov, A. Fuxman, P.P. Talukdar, and S. Chakrabarti. Scad: Collective discovery of attribute values. In *Proceedings of WWW*, 2011.

[4] M. Banko, M.J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the web. *Proceedings of IJCAI*, 2007.

[5] S. Bethard and J.H. Martin. Cu-tmp: Temporal relation classification using syntactic and semantic features. In *In SemEval-2007*, 2007.

[6] B. Boguraev and R.K. Ando. Timeml-compliant text analysis for temporal reasoning. In *Proceedings of IJCAI*, 2005.

[7] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E.R. Hruschka Jr, and T.M. Mitchell. Toward an architecture for never-ending language learning. In *Proceedings of AAAI*, 2010.

[8] N. Chambers and D. Jurafsky. Jointly combining implicit constraints improves temporal ordering. In *Proceedings of EMNLP*, 2008.

[9] M.W. Chang, L. Ratinov, N. Rizzolo, and D. Roth. Learning and inference with constraints. In *Proceedings of the 23rd National Conference on Artificial intelligence*, 2008.

[10] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A.M. Popescu, T. Shaked, S. Soderland, D.S. Weld, and A. Yates. Web-scale information extraction in knowitall:(preliminary results). In *Proceedings of WWW*, 2004.

[11] O. Gospodnetic, E. Hatcher, et al. *Lucene in action*. Manning, 2005.

[12] D. Graff, J. Kong, K. Chen, and K. Maeda. English gigaword. *Linguistic Data Consortium, Philadelphia*, 2003.

[13] X. Ling and D.S. Weld. Temporal information extraction. In *Proceedings of AAAI*, 2010.

[14] I. Mani, M. Verhagen, B. Wellner, C.M. Lee, and J. Pustejovsky. Machine learning of temporal relations. In *Proceedings of the ACL*, 2006.

[15] J.B. Michel, Y.K. Shen, A.P. Aiden, A. Veres, M.K. Gray, J.P. Pickett, D. Hoiberg, D. Clancy, P. Norvig, J. Orwant, et al. Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014), 2011.

[16] G. Puscasu. Wvali: Temporal relation identification by syntactico-semantic analysis. In *Proceedings of the 4th International Workshop on SemEval*, 2007.

[17] J. Pustejovsky, J. Castano, R. Ingria, R. Saurí, R. Gaizauskas, A. Setzer, G. Katz, and D. Radev. Timeml: Robust specification of event and temporal expressions in text. In *Fifth International Workshop on Computational Semantics*, 2003.

[18] J. Pustejovsky, P. Hanks, R. Sauri, A. See, R. Gaizauskas, A. Setzer, D. Radev, B. Sundheim, D. Day, L. Ferro, et al. The timebank corpus. In *Corpus Linguistics*, 2003.

[19] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62(1), 2006.

[20] D. Roth and W. Yih. A linear programming formulation for global inference in natural language tasks.

[21] F.M. Suchanek, G. Kasneci, and G. Weikum. Yago: a core of semantic knowledge. In *Proceedings of WWW*, 2007.

[22] P. Talukdar and K. Crammer. New regularized algorithms for transductive learning. In *Proceedings of ECML*, 2009.

[23] M. Verhagen, R. Gaizauskas, F. Schilder, M. Hepple, G. Katz, and J. Pustejovsky. Semeval-2007 task 15: Tempeval temporal relation identification. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, 2007.

[24] M. Verhagen, I. Mani, R. Sauri, R. Knippen, S.B. Jang, J. Littman, A. Rumshisky, J. Phillips, and J. Pustejovsky. Automating temporal annotation with tarsqi. In *Proceedings of the ACL Session on Interactive poster and demonstration sessions*, 2005.

[25] Y. Wang, M. Yahya, and M. Theobald. Time-aware reasoning in uncertain knowledge bases. In *MUD Workshop*, 2010.

[26] Y. Wang, B. Yang, L. Qu, M. Spaniol, and G. Weikum. Harvesting facts from textual web sources by constrained label propagation. In *Proceedings of CIKM*, 2011.

[27] Y. Wang, M. Zhu, L. Qu, M. Spaniol, and G. Weikum. Timely yago: harvesting, querying, and visualizing temporal knowledge from wikipedia. In *Proceedings of the 13th International Conference on Extending Database Technology*, 2010.

[28] G. Weikum, S. Bedathur, and R. Schenkel. Temporal knowledge for timely intelligence. *Enabling Real-Time Business Intelligence*, 2011.

[29] K. Yoshikawa, S. Riedel, M. Asahara, and Y. Matsumoto. Jointly identifying temporal relations with markov logic. In *Proceedings of ACL*, 2009.