Temporal Scoping of Relational Facts based on Wikipedia Data

Avirup Sil [*]
Computer and Information Sciences
Temple University
Philadelphia, PA 19122
avi@temple.edu

Silviu Cucerzan
Microsoft Research
One Microsoft Way
Redmond, WA 98052
silviu@microsoft.com

Abstract

Most previous work in information extraction from text has focused on named-entity recognition, entity linking, and relation extraction. Less attention has been paid given to extracting the temporal scope for relations between named entities; for example, the relation `president-Of`(John F. Kennedy, USA) is true only in the time-frame (January 20, 1961 - November 22, 1963). In this paper we present a system for temporal scoping of relational facts, which is trained on distant supervision based on the largest semi-structured resource available: Wikipedia. The system employs language models consisting of patterns automatically bootstrapped from Wikipedia sentences that contain the main entity of a page and slot-fillers extracted from the corresponding infoboxes. This proposed system achieves state-of-the-art results on 6 out of 7 relations on the benchmark Text Analysis Conference 2013 dataset for temporal slot filling (TSF), and outperforms the next best system in the TAC 2013 evaluation by more than 10 points.

1 Introduction

Previous work on relation extraction (Agichtein and Gravano, 2000; Etzioni et al., 2004) by systems such as NELL (Carlson et al., 2010), KnowItAll (Etzioni et al., 2004) and YAGO (Suchanek et al., 2007) have targeted the extraction of entity tuples, such as `president-Of`(George W. Bush, USA), in order to build large knowledge bases of facts. These systems assume that relational facts are time-invariant. However, this assumption is not always true, for example `president-Of`(George W. Bush, USA) holds within the time-frame (2001-2009) only. In this paper, we focus on the relatively less explored problem of attaching temporal scope to relation between entities. The Text Analysis Conference (TAC) introduced temporal slot filling (TSF) as one of the knowledge base population (KBP) tasks in 2013 (Dang and Surdeanu, 2013). The input to a TAC-TSF system is a binary relation *e.g.* `per:spouse`(Brad Pitt, Jennifer Aniston) and a document assumed to contain supporting evidence for the relation. The required output is a 4-tuple timestamp [T1, T2, T3, T4], where T1 and T2 are normalized dates that provide a range for the start date of the relation, and T3 and T4 provide the range for the end of the relationship. Systems must also output the offsets of the text mentions that support the temporal information extracted. For example, from a text such as *"Pitt married Jennifer Aniston on July 29, 2000 [...] the couple divorced five years later in 2005."*, a system must extract the normalized timestamp [2000-07-29, 2000-07-29, 2005-01-01, 2005-12-31], together with the entity and date offsets that support the timestamp.

In this paper, we describe TSRF, a system for **t**emporal **s**coping of **r**elational **f**acts. For every relation type, TSRF uses distant supervision from Wikipedia infobox tuples to learn a language model consisting of patterns of entity types, categories, and word n-grams. Then it uses this trained relation-specific language model to extract the top k sentences that support the given relation between the query entity and the slot filler. In a second stage, TSRF performs timestamp classification by employing models which learn "Start", "End" and "In" predictors of entities in a relationship; it computes the best 4-tuple timestamp [T1, T2, T3, T4] based on the confidence values associated to the top sentences extracted. Following the TAC-TSF task for 2013, TSRF is trained and evaluated for seven relation types, as shown in Table 1.

[*] This research was carried out during an internship at Microsoft Research.

Proceedings of the Eighteenth Conference on Computational Language Learning, pages 109–118,
Baltimore, Maryland USA, June 26-27 2014. ©2014 Association for Computational Linguistics

per:spouse
per:title
per:employee_or_member_of
org:top_employees/members
per:cities_of_residence
per:statesorprovinces_of_residence
per:countries_of_residence

Table 1: Types of relations in the TAC-TSF.

Col.1: TEMP72211	Col.7: 1492
Col.2: per:spouse	Col.8: 1311
Col.3: Brad Pitt	Col.9: 1.0
Col.4: AFP_ENG_20081208.0592	Col.10: E0566375
Col.5: Jennifer Aniston	Col.11: E0082980
Col.6: 1098	

Table 2: Input to a TSF System.

The remainder of the paper is organized as follows: The next section describes related work. Section 3 introduces the TAC-TSF input and output formats. Section 4 discusses the main challenges, and Section 5 details our method for temporal scoping of relations. Section 6 describes our experiments and results, and it is followed by concluding remarks.

2 Related Work

To our knowledge, there are only a small number of systems that have tackled the temporal scoping of relations task. YAGO (Wang et al., 2010) extracts temporal facts using regular expressions from Wikipedia infoboxes, while PRAVDA (Wang et al., 2011) uses a combination of textual patterns and graph-based re-ranking techniques to extract facts and their temporal scopes simultaneously. Both systems augment an existing KB with temporal facts similarly to the CoTS system by Talukdar et al. (2012a; 2012b). However, their underlying techniques are not applicable to arbitrary text. In contrast, TSRF automatically bootstraps patterns to learn relation-specific language models, which can be used then for processing any text. CoTS, a recent system that is part of CMU's NELL (Carlson et al., 2010) project, performs temporal scoping of relational facts by using manually edited temporal order constraints. While manual ordering is appealing and can lead to high accuracy, it is impractical from a scalability perspective. Moreover, the main goal of CoTS is to predict temporal ordering of relations rather than to scope temporally individual facts. Conversely, our system automatically extracts text patterns, and then uses them to perform temporal classification based on gradient boosted decision trees (Friedman, 2001).

The TempEval task (Pustejovsky and Verhagen, 2009) focused mainly on temporal event ordering. Systems such as (Chambers et al., 2007) and (Bethard and Martin, 2007) have been successful in extracting temporally related events. Sil et al. (2011a) automatically extract STRIPS representations (Fikes and Nilsson, 1971) from web text, which are defined as states of the world before and after an event takes place. However, all these efforts focus on temporal ordering of either events or states of the world and do not extract timestamps for events. By contrast, the proposed system extracts temporal expressions and also produces an ordering of the timestamps of relational facts between entities.

The current state-of-the-art systems for TSF have been the RPI-Blender system by Artiles et al. (2011) and the UNED system by Garrido et al. (2011; 2012). These systems obtained the top scores in the 2011 TAC TSF evaluation by outperforming the other participants such as the Stanford Distant Supervision system (Surdeanu et al., 2011). Similar to our work, these systems use distant supervision to assign temporal labels to relations extracted from text. While we employ Wikipedia infoboxes in conjunction with Wikipedia text, the RPI-Blender and UNED systems use tuples from structured repositories like Freebase. There are major differences in terms of learning strategies of these systems: the UNED system uses a rich graph-based document-level representation to generate novel features whereas RPI-Blender uses an ensemble of classifiers combining flat features based on surface text and dependency paths with tree kernels. Our system employs language models based on Wikipedia that are annotated automatically with entity tags in a boosted-trees learning framework. A less important difference between TSRF and RPI-Blender is that the latter makes use of an additional temporal label (Start-And-End) for facts within a time range; TSRF employs Start, End, and In labels.

3 The Temporal Slot Filling Task

3.1 Input

The input format for a TSF system as instantiated for the relation `per:spouse`(Brad Pitt, Jennifer

Aniston) is shown in Table 2. The field Column 1 contains a unique query ID for the relation. Column 2 is the name of the relationship, which also encodes the type of the target entity. Column 3 contains the name of the query entity, i.e., the subject of the relation. Column 4 contains a valid document ID and Column 5 indicates the slot-filler entity. Columns 6 through 8 are offsets of the slot-filler, query entity and the relationship justification in the given text. Column 9 contains a confidence score set to 1 to indicate that the relation is correct. Columns 10 and 11 contain the IDs in the KBP knowledge base of the entity and filler, respectively. All of the above are provided by TAC. For the query in this example, a TSF system has to scope temporally the `per:spouse` relation between Brad Pitt and Jennifer Aniston.

3.2 Output

Similar to the regular slot filling task in TAC, the TSF output includes the offsets for at least one entity mention and up to two temporal mentions used for the extraction and normalization of hypothesized answer. For instance, assume that a system extracts the relative timestamp "Monday" and normalizes it to "2010-10-04" for the relation `org:top_employee`(Twitter, Williams) using the document date from the following document:

```
<DOCID> AFP_ENG_20101004.0053.LDC2010T13 </DOCID>
<DATETIME> 2010-10-04 </DATETIME>
<HEADLINE>
Twitter co-founder steps down as CEO
</HEADLINE>
<TEXT>
<P>
Twitter co-founder Evan Williams announced on Monday
that he was stepping down as chief executive [...]
```

The system must report the offsets for both "Monday" in the text body and "2010-10-04" in the DATETIME block for the justification.

The TAC-TSF task uses the following representation for the temporal information extracted: For each relation provided in the input, TSF systems must produce a 4-tuple of dates: [T1, T2, T3, T4], which indicates that the relation is true for a period beginning at some point in time between T1 and T2 and ending at some time between T3 and T4. By convention, a hyphen in one of the positions implies a lack of a constraint. Thus, [-, 20120101, 20120101, -] implies that the relation was true starting on or before January 1, 2012 and ending on or after January 1, 2012. As discussed in the TAC 2011 pilot study by Ji et al. (2011), there are situations that cannot be covered by this representation, such as recurring events, for example repeated marriages between two persons. However, the most common situations for the relations covered in this task are captured correctly by this 4-tuple representation.

4 Challenges

We discuss here some of the main challenges encountered in building a temporal scoping system.

4.1 Lack of Annotated Data

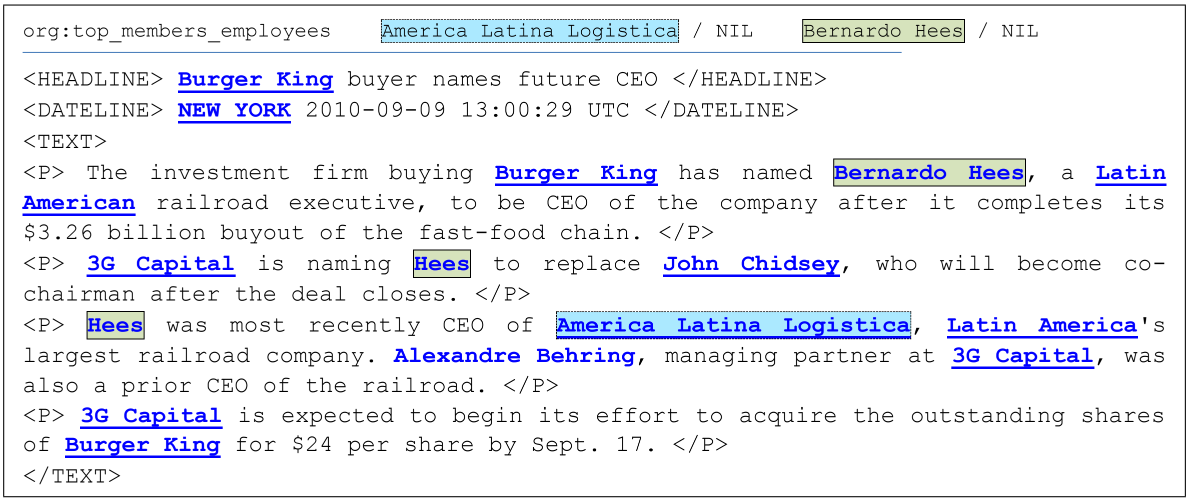
Annotation of data for this task is expensive, as the human annotators must have extensive background knowledge and need to analyze the evidence in text and reliable knowledge resources. As per (Ji et al., 2013), a large team of human annotators were able to generate only 1,172 training instances for 8 slots for KBP 2011. The authors of the study concluded that such amount of data is not enough for training a supervised temporal scoping system. They also noted that only 32% of `employee_Of` queries were found to have potential temporal arguments, and only one third of the queries could have reliable start or end dates.

4.2 Date Normalization

Sometimes temporal knowledge is not stated explicitly in terms of dates or timestamps. For example, from the text "they got married on Valentine's Day" a system can extract Valentine's Day as the surface form of the start of the `per:spouse` relation. However, for a temporal scoping system it needs to normalize the temporal string to the date of February 14 and the year to which the document refers to explicitly in text or implicitly, such as the year in which the document was published.

4.3 Lexico-Syntactic Variety

A relation can be specified in text by employing numerous syntactic and lexical constructions; *e.g.* for the `per:spouse` relation the patterns "got married on [DATE]" and "vowed to spend eternity on [DATE]" have the same meaning. Additionally, entities can appear mentioned in text in various forms, different from the canonical form given as input. For instance, Figure 1 shows an example in which the input entity Bernardo Hees, which is not in Wikipedia, is mentioned three times, with two of the mentions using a shorter form (the last name of the person).



```
org:top_members_employees    America Latina Logistica / NIL    Bernardo Hees / NIL

<HEADLINE> Burger King buyer names future CEO </HEADLINE>
<DATELINE> NEW YORK 2010-09-09 13:00:29 UTC </DATELINE>
<TEXT>
<P> The investment firm buying Burger King has named Bernardo Hees, a Latin
American railroad executive, to be CEO of the company after it completes its
$3.26 billion buyout of the fast-food chain. </P>
<P> 3G Capital is naming Hees to replace John Chidsey, who will become co-
chairman after the deal closes. </P>
<P> Hees was most recently CEO of America Latina Logistica, Latin America's
largest railroad company. Alexandre Behring, managing partner at 3G Capital, was
also a prior CEO of the railroad. </P>
<P> 3G Capital is expected to begin its effort to acquire the outstanding shares
of Burger King for $24 per share by Sept. 17. </P>
</TEXT>
```

Figure 1: Example data point from the TAC TSF 2013 training set, with the annotations hypothesized by our system. The entity mentions identified by the entity linking (EL) component are shown in bold blue; those that were linked to Wikipedia are also underlined. The highlighting (blue and green) is used to show the mentions in the coreference chains identified for the two input entities, “America Latina Logistica” and “Bernardo Hees”.

4.4 Inferred Meaning

A temporal scoping system also needs to learn the inter-dependence of relations, and how one event affects another. For instance, in our automatically generated training data, we learn that a death event specified by n-grams like “was assassinated” affects the per:title relation, and it indicates that the relationship ended at that point. In Figure 1, while the CEO relationships for Bernardo Hees with America Latina Logistica and Burger King are indicated by clear patterns (“was most recently CEO of” and “to be CEO of”), the temporal stamping is difficult to achieve in both cases, as there is no standard normalization for “recently” in the former, and it is relative to the completion of the buyout event in the latter.

4.5 Pattern Trustworthiness

A temporal scoping system should also be able to model the trustworthiness of text patterns, and even the evolution of patterns that indicate a relationship over time. For example, in current news, the birth of a child does not imply that a couple is married, although it does carry a strong signal about the marriage relationship.

5 Learning to Attach Temporal Scope

5.1 Automatically Generating Training Data

As outlined in Section 4, one of the biggest challenges of a temporal scoping system is the lack of annotated data to create a strong information extraction system. Previous work on relation extraction such as (Mintz et al., 2009) has shown that distant supervision can be highly effective in building a classifier for this purpose. Similar to supervised classification techniques, some advantages of using distant supervision are:

- It allows building classifiers with a large number of features;
- The supervision is provided intrinsically by the detailed user-contributed knowledge;
- There is no need to expand patterns iteratively.

Mintz et al. also point out that similar to unsupervised systems, distant supervision also allows:

- Using large amounts of unlabeled data such as the Web and social media;
- Employing techniques that are not sensitive to the genre of training data.

We follow the same premise as (Cucerzan, 2007; Weld et al., 2009) that the richness of the Wikipedia collection, whether semantic, lexical, syntactic, or structural, is a key enabler in redefining the state-of-the-art for many NLP and IR task. Our target is to use distant supervision from Wikipedia data to build an automatic temporal scoping system. However, for most relations, we find that Wikipedia does not indicate specific start or end dates in a structured form. In addition to this, we need our system to be able to predict whether two entities are currently in a relationship or not based on the document date as well.

Hence, in our first step, we build an automatic system which takes as input a binary relation between two entities *e.g.* `per:spouse`(Brad Pitt, Jennifer Aniston) and a number of documents. The system needs to extract highly ranked/relevant sentences, which indicate that the two entities are in the targeted relationship. The next component takes as input the top k sentences generated in the previous step and extracts temporal labels for the input relation. Note that our target is to develop algorithms that are not relation-specific but rather can work well for a multitude of relations. We elaborate on these two system components further.

5.1.1 Using Wikipedia as a Resource for Distant Supervision

Wikipedia is the largest freely available encyclopedic collection, which is built and organized as a user-contributed knowledge base (KB) of entities. The current version of the English Wikipedia contains information about 4.2 million entities. In addition to the plain text about these entities, Wikipedia also contains structured components. One of these is the *infobox*. Infoboxes contain information about a large number of relations for the target entity of the Wikipedia page, *e.g.* names of spouses, birth and death dates, residence *etc.*. Similar to structured databases, the infoboxes contain the most important/useful relations in which entities take part, while the text of Wikipedia pages contains mentions and descriptions of these relations. Because of this, Wikipedia can be seen as a knowledge repository that contains parallel structured and unstructured information about entities, and therefore, can be employed more easily than Freebase or other structured databases for building a relation extraction system. Figure 2 shows how sentences from Wikipedia can be used to train a system for the temporal slot filling task.

5.1.2 Extracting Relevant Sentences

For every relation, we extract slot-filler names from infoboxes of each Wikipedia article. We also leverage Wikipedia's rich interlinking model to automatically retrieve labeled entity mentions in text. Because the format of the text values provided by different users for the infobox attributes can vary greatly, we rely on regular expressions to extract slot-filler names from the infoboxes. For every relation targeted, we build a large set of regular expressions to extract entity names and filter out noise *e.g.* html tags, redundant text *etc.*.

To extract all occurrences of named-entities in the Wikipedia text, we relabel each Wikipedia article with Wikipedia interlinks by employing the entity linking (EL) system by Cucerzan (2012), which obtained the top scores for the EL task in successive TAC evaluations. This implementation takes into account and preserves the interlinks created by the Wikipedia contributors, and extracts all other entity mentions and links them to Wikipedia pages if possible or hypothesizes coreference chains for the mentions of entities that are not in Wikipedia. The latter are extremely important when the slot-filler for a relation is an entity that does not have a Wikipedia page, as often is the case with spouses or other family members of famous people (as shown in Figure 1 for the slot-filler Bernardo Hees).

As stated in Section 4, temporal information in text is specified in various forms. To resolve temporal mentions, we use the Stanford SUTime (Chang and Manning, 2012) *temporal tagger*. The system exhibits strong performance outperforming state-of-the-art systems like HeidelTime (Strötgen and Gertz, 2010) on the TempEval-2 Task A (Verhagen et al., 2010) in English. SUTime is a rule-based temporal tagger that employs regular expression. Its input is English text in tokenized format; its output contains annotations in the form of TIMEX3 tags. TIMEX3 is a part of the TimeML annotation language as introduced by (Pustejovsky et al., 2003) and is used to markup date and time, events, and their temporal relations in text. When processing Web text, we often encounter date expressions that contain a relative time *e.g.* "last Thursday". To resolve them to actual dates/time is a non-trivial task. However, the heuristic of employing the document's publication date as the reference works very well in practice *e.g.* for a document published on 2011-07-05, SUTime resolves "last Thursday" to 2011-06-30. It provides temporal tags in the following labels: Time, Duration, Set and Interval. For our experiments we used Time and Duration.

After running the Stanford SUTime, which automatically converts date expressions to their normalized form, we collect sets of contiguous sentences from the page that contain one mention of the targeted entity and one mention of the slot-filler, as extracted by the entity linking system. We then build a large language model by bootstrapping textual patterns supporting the relations, similar to (Agichtein and Gravano, 2000). The general intuition is that a set of sentences that mention the two entities are likely to state something about relationships in which they are.

For assigning sentences a relevance score with respect to a targeted relation, we represent the sentences in an input document (i.e., Wikipedia page) as d dimensional feature vectors, which incorporate statistics about how relevant sentences are to the relation between a query_entity q and the slot_filler z. For example, for the `per:spouse` relation, one binary feature is "does the input sentence contain the n-gram "QUERY_ENTITY got married"". Note that the various surface forms/mentions of q and z are resolved to their canonical target at this stage.

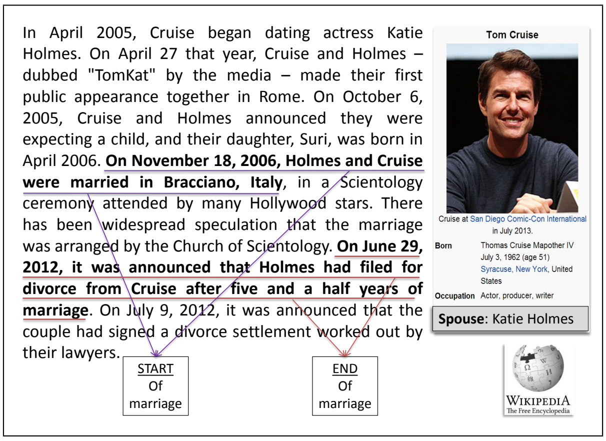
We were able to extract 61,872 tuples of query entity and slot filler relations from Wikipedia for the `per:spouse` relation. Figure 2 shows how we extract relevant sentences using slot-filler names from Wikipedia. Consider the following text (already processed by our EL system and Stanford SUTime) taken from the Wikipedia page of Tom Cruise:

> On [November 18, 2006$|_{2006-11-18}$], [Holmes$|_{Katie_Holmes}$] and [Cruise$|_{Tom_Cruise}$] were married in [Bracciano$|_{Bracciano}$] . . .
>
> On [June 29, 2012$|_{2012-06-29}$], [Holmes$|_{Katie_Holmes}$] filed for divorce from [Cruise$|_{Tom_Cruise}$] after five and a half years of marriage.

Considering Tom Cruise as the query entity and his wife Katie Holmes as the slot filler for the `per:spouse` relation, we normalize the above text to the following form to extract features:

> On _DATE, SLOT_FILLER and QUERY_ENTITY were married in _LOCATION . . .
>
> On _DATE, SLOT_FILLER filed for divorce from QUERY_ENTITY after five and a half years of marriage.

Our language model consists of n-grams ($n \leq 5$) like "SLOT_FILLER and QUERY_ENTITY were married", "SLOT_FILLER filed for divorce from" which provides clues for the marriage relation. These n-grams are then used as features with an implementation of a gradient boosted decision trees classifier similar to that described by (Friedman, 2001; Burges, 2010). We also use features provided by the EL system which are based on entity types and categories. We call this "relationship" classifier RELCL. The output of this step is a ranked list of sentences which indicate whether there exists a relationship between the query entity and the slot filler.



In April 2005, Cruise began dating actress Katie Holmes. On April 27 that year, Cruise and Holmes – dubbed "TomKat" by the media – made their first public appearance together in Rome. On October 6, 2005, Cruise and Holmes announced they were expecting a child, and their daughter, Suri, was born in April 2006. **On November 18, 2006, Holmes and Cruise were married in Bracciano, Italy**, in a Scientology ceremony attended by many Hollywood stars. There has been widespread speculation that the marriage was arranged by the Church of Scientology. **On June 29, 2012, it was announced that Holmes had filed for divorce from Cruise after five and a half years of marriage**. On July 9, 2012, it was announced that the couple had signed a divorce settlement worked out by their lawyers.

Tom Cruise

Cruise at San Diego Comic-Con International in July 2013.

Born	Thomas Cruise Mapother IV July 3, 1962 (age 51) Syracuse, New York, United States
Occupation	Actor, producer, writer

Spouse: Katie Holmes

Figure 2: Example of relevant sentences extracted by using query entity and slot-filler names from Wikipedia for the `per:spouse` relation.

5.1.3 Learning Algorithm

Our objective is to rank the sentences in a document based on the premise that entities q and z are in the targeted relation r. We tackle this ranking task by using gradient boosted decision trees (GBDT) to learn temporal scope for entity relations. Previous work such as Sil et al. (2011a; 2011b) used SVMs for ranking event preconditions and (Cucerzan, 2012) and (Zhou et al., 2010) employed GBDT for ranking entities. GBDT can achieve high accuracy as they can easily combine features of different scale and missing values. In our experiments, GBDT outperforms both SVMs and MaxEnt models.

We employ the stochastic version of GBDT similar to (Friedman, 2001; Burges, 2010). Basically, the model performs a numerical optimization in the function space by computing a function approximation in a sequence of steps. By building a smaller decision tree at each step, the model computes residuals obtained in the previous step. Note that in the stochastic variant of GBDT, for computing the loss function, the model absorbs several samples instead of using the whole training data. The parameters for our GBDT model were tuned on a development set sampled from our Wikipedia dump independent from the training set. These parameters include the number of regression trees and the shrinkage factor.

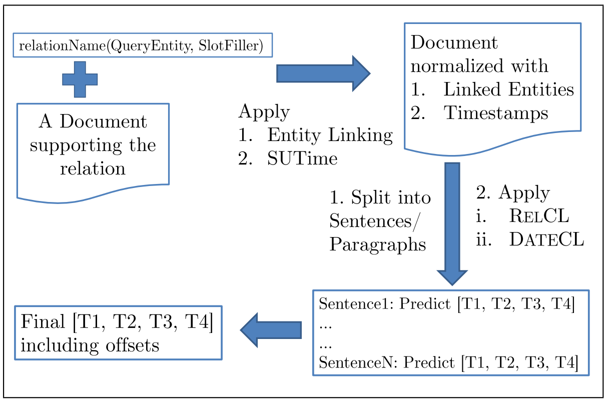

relationName(QueryEntity, SlotFiller)
A Document supporting the relation
Apply
1. Entity Linking
2. SUTime
Document normalized with
1. Linked Entities
2. Timestamps
1. Split into Sentences/ Paragraphs
2. Apply
i. RELCL
ii. DATECL
Sentence1: Predict [T1, T2, T3, T4]
...
...
SentenceN: Predict [T1, T2, T3, T4]
Final [T1, T2, T3, T4] including offsets

Figure 3: Architecture of the proposed system. Every input document is processed by the (Cucerzan, 2012) entity linking system and the Stanford SUTime system. Temporal information is then extracted automatically using RELCL and DATECL.

5.1.4 Gathering Relevant Sentences

On the unseen test data, we apply our trained model and obtain a score for each new sentence s that contains mentions of entities q and z that are in a targeted relationship by turning s into a feature vector as shown previously. Among all sentences that contain mentions of q and z, we choose the top k with the highest score. The value of k was tuned based on the performance of TSRF on our development set.

5.1.5 Extracting Timestamps

To predict timestamps for each relation, we build another classifier, DATECL similar to that described in the previous section, by using language models for "Start", "End" and "In" predictors of relationship. The "Start" model predicts T1, T2; "End" predicts T3, T4 and "In" predicts T2, T3.

Raw Trigger Features: Similar to previous work by (Sil et al., 2010) on using discriminative words as features, each of these models compose of "Trigger Words" that indicate when a relationship begins or ends. In the current implementation, these triggers are chosen manually from the language model automatically bootstrapped from Wikipedia. Future directions include how to automatically learn these triggers. For example, for the `per:spouse` relation, the triggers for "Start" contain n-grams such as "married since _DATE" and "married SLOT_FILLER on"; the "End" model contains n-grams such as "estranged husband QUERY_ENTITY", "split in _DATE"; the "In" model contains "happily married", "QUERY_ENTITY with his wife" *etc.*. For an input sentence with query entity q and slot-filler z, a first class of raw trigger features consists of cosine-similarity(Text(q, z), Triggers(r)) where $r \in Start, End, In$. Here, Text(q, z) indicates the full sentence as context. We also employ another feature that computes cosine-similarity(Context(q, z), Triggers(r)), which constructs a *mini-sentence* Context(q, z) from the original by choosing windows of three words before and after q and z, and ignoring duplicates.

External Event Triggers: Our system also considers the presence of other events as triggers *e.g.* a "death" event signaled by "SLOT_FILLER died" might imply that a relationship ended on that timestamp. Similarly, a "birth" event can imply that an entity started living in a particular location *e.g.* the `per:born-In`(Obama, Honolulu) relation from the sentence "President Obama was born in Honolulu in 1961" indicates that T1 = 1961-01-01 and T2 = 1961-12-31 for the relation `per:cities_of_residence`(Obama, Honolulu).

At each step, TSRF extracts the top timestamps for predicting "Start", "End" and "In" based on the confidence values of DATECL. Similar to previous work by (Artiles et al., 2011), we aggregate and update the extracted timestamps using the following heuristics:

Step 1: Initialize T= [-∞, +∞, -∞,+ ∞]
Step 2: Iterate through the classified timestamps
Step 3: For a new T' aggregate :

$$T\&\&T' = [max(t_1, t_1'), min(t_2, t_2'),$$
$$max(t_3, t_3'), min(t_4, t_4')]$$

Update only if: $t_1 \leq t_2; t_3 \leq t_4; t_1 \leq t_4$

This novel two-step classification strategy removes noise introduced by distant supervision training and decides if the extracted (entity, filler, timestamp) tuples belong to the relation under consideration or not. For example, for the `per:spouse` relation between the entities Brad Pitt and Jennifer Aniston, TSRF extracts sentences like "*..On November 22, 2001, Pitt made a guest appearance in the television series Friends, playing a man with a grudge against Rachel Green, played by Jennifer Aniston..*" and "*Pitt met Jennifer Aniston in 1998 and married her in a private wedding ceremony in Malibu on July 29, 2000..*". Note that both sentences contain the query entity and the slot filler. The system automatically rejects the extraction of temporal information from

	S1	S2	S3	S4	S5	S6	S7	**ALL**	StDev
Baseline	24.70	17.40	15.18	17.83	14.75	21.08	23.20	19.10	3.60
TSRF	**31.94**	**36.06**	**32.85**	**40.12**	**33.04**	**31.85**	27.35	**33.15**	3.66
RPI-Blender	31.19	13.07	14.93	26.71	29.04	17.24	**34.68**	23.42	7.98
UNED	26.20	6.88	8.16	15.24	14.47	14.41	19.34	14.79	6.07
CMU-NELL	19.95	7.46	8.47	16.52	13.43	5.65	11.95	11.53	4.77
Abby-Compreno	0.0	2.42	8.56	0.0	13.50	7.91	0.0	5.14	4.99
LDC	69.87	60.22	58.26	72.27	81.10	54.07	91.18	68.84	12.32

Table 3: Results for the TAC-TSF 2013 test set, overall and for individual slots. The slots notation is: S1: org:top members employees, S2: per:city of residence, S3: per:country of residence, S4: per:employee or member of, S5: per:spouse, S6: per:statesorprovince of residence, S7: per:title. The score for the output created by the LDC experts is also shown.

the former even though the sentence contains mentions of both entities. This is because the language model for the marriage relation does not match well this candidate sentence, which is actually focussing on the two entities being in the different relation of co-acting/appearing in the same motion picture. The latter sentence is determined as matching the language model for the marriage relation, and TSRF extracts the temporal scope July 29, 2000 and attaches the START label to it. Most previous systems do not perform this noise removal step, which is a critical component in our distant supervision approach.

6 Experiments

For evaluation, we train our system on the infobox tuples and sentences extracted from the Wikipedia dump of May 2013. We set aside a portion of the dump as our development data. We chose to use the top-relevant n-grams based on the performance on the development data as features. We employ then the TAC evaluation data, which is publicly available through LDC.

We utilize the evaluation metric developed for TAC (Dang and Surdeanu, 2013). In order for a temporal constraint (T1-T4) to be valid, the document must justify both the query relation (which is similar to the regular English slot filling task) and the temporal constraint. Since the time information provided in text may be approximate, the TAC metric measures the similarity of each constraint in the key and system response. Formally, if the date in the gold standard is k_i, while the date hypothesized by the system is r_i, and $d_i = |k_i - r_i|$ is their difference measured in years, then the score for the set of temporal constraints on a slot is computed as:

$$Score(slot) = \frac{1}{4}\sum_{i=1}^{4} \frac{c}{c + d_i}$$

TAC sets the constant c to one year, so that predictions that differ from the gold standard by one year get 50% credit. The absence of a constraint in T1 or T3 is treated as a value of $-\infty$ and the absence of a constraint in T2 or T4 is treated as $+\infty$, which lead to zero-value terms in the scoring sum. Therefore, the overall achievable score has a range between 0 and 1.

We compare TSRF against four other TSF systems: **(i)** RPI-Blender (Artiles et al., 2011), **(ii)** CMU-NELL (Talukdar et al. (2012a; 2012b)), **(iii)** UNED (Garrido et al. (2011; 2012)) and **(iv)** Abby-Compreno (Kozlova et al., 2012). Most of these systems employ distant supervision strategies too. RPI-Blender and UNED obtained the top scores in the 2011 TAC TSF pilot evaluation, and thus, could be considered as the state-of-the-art at the time.

We also compare our system with a reasonable baseline similar to (Ji et al., 2011). This baseline makes the simple assumption that the corresponding relation is valid at the document date. That means that it creates a "within" tuple as follows: $< -\infty$, *doc_date*, *doc_date*, $+\infty >$. Hence, this baseline system for a particular relation always predicts T2 = T3 = the date of the document.

Table 3 lists the results obtained by our system on the TAC test set of 201 queries, overall and for each individual slot, in conjunction with the results of the other systems evaluated and the output generated by the LDC human experts. Only two out of the five systems evaluated, TSRF and RPI-Blender, are able to beat the "within" baseline.

TSRF achieves approximately 48% of human performance (LDC) and outperforms all other sys-

	TSF Accuracy	SF F1	SF Prec	SF Recall
LDC	68.8	83.1	97.3	72.5
TSRF	**33.1**	**77.3**	**96.8**	**64.4**
RPI-Blender	23.4	51.8	69.2	41.4
UNED	14.8	46.6	69.9	35.0
CMU-NELL	11.5	32.2	38.5	27.6
Abby-Compreno	5.1	18.5	53.6	11.2

Table 4: Extraction accuracy for slot-filler mentions. TSRF clearly outperforms all systems and comes close to human performance (LDC).

tems in overall score, as well as for all individual relations with the exception of `per:title`, for which RPI-Blender obtains a better score. In fact, TSRF outperforms the next best systems by 10 and 19 points. These two systems obtained the top score in TAC 2011, and outperformed other systems such as Stanford (Surdeanu et al., 2011). TSRF also outperforms CMU-NELL which employs a very large KB of relational facts already extracted from the Web and makes use of the Google N-gram corpus (http://books.google.com/ngrams).

We believe that this large performance difference is due in part to the fact that TSRF uses a language model to clean up the noise introduced by distant supervision before the actual temporal classification step. Also, the learning algorithm employed, GBDT, is highly effective in using the extracted n-grams as features to decide whether the extracted (entity, filler, time) tuples belong to the relation under consideration or not. Finally, Table 4 shows another reason that gives TSRF an edge in obtaining the best score. The employed EL component (Cucerzan, 2012) is a state-of-the-art system for extracting and linking entities, and resolving coreference chains. By using this system, we have been able to extract slot-filler mentions with a precision of 96.8% at 66.4% recall, which is substantially higher than the extraction results of all other systems. Encouragingly, the performance of this component also comes close to that of the LDC annotators, which obtained a precision of 97.3% at 72.5% recall.

It is also important to note that our system exhibits a balanced performance on the relations on which it was tested. As shown in column StDev in Table 3, this system achieves the lowest standard deviation in the performance across the relations tested. It is interesting to note also that TSRF achieves the best performance on the `employee_of` (S4) and `city_of_residence` (S2) relations even though the system development was done on the `spouse` relation (S1) as an encouraging sign that our distant supervision algorithm can be transferred successfully across relations for domain-specific temporal scoping.

7 Conclusion and Future Work

The paper described an automatic temporal scoping system that requires no manual labeling effort. The system uses distant supervision from Wikipedia to obtain a large training set of tuples for training. It uses a novel two-step classification to remove the noise introduced by the distant supervision training. The same algorithm was employed for multiple relations and exhibited similarly high accuracy. Experimentally, the system outperforms by a large margin several other systems that address this relatively less explored problem. Future directions of development include extracting joint slot filler names and temporal information, and leveraging the changes observed over time in Wikipedia for a query entity and a slot filler in a target relation.

References

E. Agichtein and L. Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Procs. of the Fifth ACM International Conference on Digital Libraries*.

Javier Artiles, Qi Li, Taylor Cassidy, Suzanne Tamang, and Heng Ji. 2011. CUNY BLENDER TACKBP2011 Temporal Slot Filling System Description. In *TAC*.

Steven Bethard and James H Martin. 2007. Cu-tmp: Temporal relation classification using syntactic and semantic features. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 129–132.

Chris Burges. 2010. From ranknet to lambdarank to lambdamart: An overview. *Learning*, 11:23–581.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka Jr, and Tom M Mitchell. 2010. Toward an architecture for never-ending language learning. In *AAAI*.

Nathanael Chambers, Shan Wang, and Dan Jurafsky. 2007. Classifying temporal relations between events. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 173–176.

Angel X Chang and Christopher Manning. 2012. Sutime: A library for recognizing and normalizing time expressions. In *LREC*, pages 3735–3740.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL*, pages 708–716.

Silviu Cucerzan. 2012. The MSR System for Entity Linking at TAC 2012. In *TAC*.

Hoa Trang Dang and Mihai Surdeanu. 2013. Task description for knowledge-base population at TAC 2013. In *TAC*.

O. Etzioni, M. Cafarella, D. Downey, S. Kok, A. Popescu, T. Shaked, S. Soderland, D. Weld, and A. Yates. 2004. Web-Scale Information Extraction in KnowItAll. In *WWW*, New York City, New York.

R. Fikes and N. Nilsson. 1971. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3/4):189–208.

Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pages 1189–1232.

Guillermo Garrido, Bernardo Cabaleiro, Anselmo Penas, Alvaro Rodrigo, and Damiano Spina. 2011. A distant supervised learning system for the tac-kbp slot filling and temporal slot filling tasks. In *TAC*.

Guillermo Garrido, Anselmo Penas, Bernardo Cabaleiro, and Alvaro Rodrigo. 2012. Temporally anchored relation extraction. In *ACL*.

Heng Ji, Ralph Grishman, and Hoa Trang Dang. 2011. Overview of the tac2011 knowledge base population track. In *TAC*.

Heng Ji, Taylor Cassidy, Qi Li, and Suzanne Tamang. 2013. Tackling representation, annotation and classification challenges for temporal knowledge base population. *Knowledge and Information Systems*, pages 1–36.

Ekaterina Kozlova, Manicheva Maria, Petrova Elena, and Tatiana Popova. 2012. The compreno semantic model as an integral framework for a multilingual lexical database. In *3rd Workshop on Cognitive Aspects of the Lexicon (CogALex-III)*.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *ACL*, pages 1003–1011.

James Pustejovsky and Marc Verhagen. 2009. Semeval-2010 task 13: evaluating events, time expressions, and temporal relations (tempeval-2). In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pages 112–116.

James Pustejovsky, José M Castano, Robert Ingria, Roser Sauri, Robert J Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R Radev. 2003. Timeml: Robust specification of event and temporal expressions in text. *New directions in question answering*, 3:28–34.

Avirup Sil and Alexander Yates. 2011a. Extracting STRIPS representations of actions and events. In *RANLP*.

Avirup Sil and Alexander Yates. 2011b. Machine Reading between the Lines: A Simple Evaluation Framework for Extracted Knowledge Bases. In *Workshop on Information Extraction and Knowledge Acquisition (IEKA)*.

Avirup Sil, Fei Huang, and Alexander Yates. 2010. Extracting action and event semantics from web text. In *AAAI Fall Symposium on Common-Sense Knowledge (CSK)*.

Jannik Strötgen and Michael Gertz. 2010. Heideltime: High quality rule-based extraction and normalization of temporal expressions. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 321–324.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *WWW*.

Mihai Surdeanu, Sonal Gupta, John Bauer, David McClosky, Angel X Chang, Valentin I Spitkovsky, and Christopher D Manning. 2011. Stanfords distantly-supervised slot-filling system. In *TAC*.

Partha Pratim Talukdar, Derry Wijaya, and Tom Mitchell. 2012a. Acquiring temporal constraints between relations. In *CIKM*.

Partha Pratim Talukdar, Derry Wijaya, and Tom Mitchell. 2012b. Coupled temporal scoping of relational facts. In *WSDM*.

Marc Verhagen, Roser Sauri, Tommaso Caselli, and James Pustejovsky. 2010. Semeval-2010 task 13: Tempeval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 57–62.

Yafang Wang, Mingjie Zhu, Lizhen Qu, Marc Spaniol, and Gerhard Weikum. 2010. Timely yago: harvesting, querying, and visualizing temporal knowledge from wikipedia. In *Proceedings of the 13th International Conference on Extending Database Technology*, pages 697–700. ACM.

Yafang Wang, Bin Yang, Lizhen Qu, Marc Spaniol, and Gerhard Weikum. 2011. Harvesting facts from textual web sources by constrained label propagation. In *CIKM*, pages 837–846.

Daniel S. Weld, Raphael Hoffmann, and Fei Wu. 2009. Using Wikipedia to Bootstrap Open Information Extraction. In *ACM SIGMOD Record*.

Yiping Zhou, Lan Nie, Omid Rouhani-Kalleh, Flavian Vasile, and Scott Gaffney. 2010. Resolving surface forms to wikipedia topics. In *COLING*, pages 1335–1343.