Temporally Anchored Relation Extraction

Guillermo Garrido, Anselmo Peñas, Bernardo Cabaleiro, and **Álvaro Rodrigo**
NLP & IR Group at UNED
Madrid, Spain
{ggarrido,anselmo,bcabaleiro,alvarory}@lsi.uned.es

Abstract

Although much work on relation extraction has aimed at obtaining static facts, many of the target relations are actually *fluents*, as their validity is naturally anchored to a certain time period. This paper proposes a methodological approach to temporally anchored relation extraction. Our proposal performs distant supervised learning to extract a set of relations from a natural language corpus, and anchors each of them to an interval of temporal validity, aggregating evidence from documents supporting the relation. We use a rich graph-based document-level representation to generate novel features for this task. Results show that our implementation for temporal anchoring is able to achieve a 69% of the upper bound performance imposed by the relation extraction step. Compared to the state of the art, the overall system achieves the highest precision reported.

1 Introduction

A question that arises when extracting a relation is how to capture its temporal validity: Can we assign a period of time when the obtained relation held? As pointed out in (Ling and Weld, 2010), while much research in automatic relation extraction has focused on distilling static facts from text, many of the target relations are in fact *fluents*, dynamic relations whose truth value is dependent on time (Russell and Norvig, 2010).

The *Temporally anchored relation extraction* problem consists in, given a natural language text document corpus, C, a target entity, e, and a target relation, r, extracting from the corpus the value of that relation for the entity, and a temporal interval for which the relation was valid.

In this paper, we introduce a methodological approach to temporal anchoring of relations automatically extracted from unrestricted text. Our system (see Figure 1) extracts relational facts from text using distant supervision (Mintz et al., 2009) and then anchors the relation to an interval of temporal validity. The intuition is that a distant supervised system can effectively extract relations from the source text collection, and a straightforward date aggregation can then be applied to anchor them. We propose a four step process for temporal anchoring: (1) represent temporal evidence; (2) select temporal information relevant to the relation; (3) decide how a relational fact and its relevant temporal information are themselves related; and (4) aggregate imprecise temporal intervals across multiple documents. In contrast with previous approaches that aim at intra-document temporal information extraction (Ling and Weld, 2010), we focus on mining a corpus aggregating temporal evidences across the supporting documents.

We address the following research questions: (1) Validate whether distant supervised learning is suitable for the task, and evaluate its shortcomings. (2) Explore whether the use of features extracted from a document-level rich representation could improve distant supervised learning. (3) Compare the use of document metadata against temporal expressions within the document for relation temporal anchoring. (4) Analyze how, in a pipeline architecture, the propagation of errors limits the overall system's

Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pages 107–116,
Jeju, Republic of Korea, 8-14 July 2012.

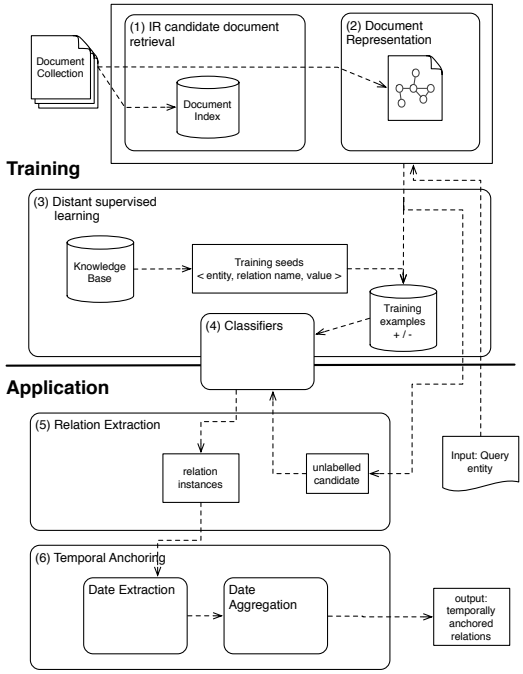

Figure 1: System overview diagram.

performance.

The representation we use for temporal information is detailed in section 2; the rich document-level representation we exploit is described in section 3. For a query entity and target relation, the system first performs relation extraction (section 4); then, we find and aggregate time constraint evidence for the same relation across different documents, to establish a temporal validity anchor interval (section 5). Empirical comparative evaluation of our approach is introduced in section 6; while some related work is shown in section 7 and conclusions in section 8.

2 Temporal Anchors

We will denominate *relation instance* a triple $\langle entity, relation\ name, value \rangle$. We aim at anchoring relation instances to their temporal validity. We need a representation flexible enough to capture the imprecise temporal information available in text, but expressed in a structured style. Allen's (1983) interval-based algebra for temporal representation and reasoning, underlies much research, such as the Tempeval challenges (Verhagen et al., 2007; Pustejovsky and Verhagen, 2009). Our task is different, as we focus on obtaining the temporal interval associated to a fact, rather than reasoning about the temporal relations among the events appearing in a single text.

Let us assume that each relation instance is valid during a certain temporal interval, $I = [t_0, t_f]$. This sharp temporal interval fails to capture the imprecision of temporal boundaries conveyed in natural language text. The Temporal Slot Filling task at TAC-KBP 2011 (Ji et al., 2011) proposed a 4-tuple representation that we will refer to as *imprecise anchor intervals*. An imprecise temporal interval is defined as an ordered 4-tuple of time points: (t_1, t_2, t_3, t_4), with the following semantics: the relation is true for a period which starts at some point between t_1 and t_2 and ends between t_3 and t_4. It should hold that: $t_1 \leq t_2$, $t_3 \leq t_4$, and $t_1 \leq t_4$. Any of the four endpoints can be left unconstrained (t_1 or t_3 would be $-\infty$, and t_2 or t_4 would be $+\infty$). This representation is flexible and expressive, although it cannot capture certain types of information (Ji et al., 2011).

3 Document Representation

We use a rich document representation that employs a graph structure obtained by augmenting the syntactic dependency analysis of the document with semantic information.

A document D is represented as a *document graph* G_D; with node set V_D and edge set, E_D. Each node $v \in V_D$ represents a *chunk* of text, which is a sequence of words[1]. Each node is labeled with a dictionary of attributes, some of which are common for every node: the words it contains, their part-of-speech annotations (POS) and lemmas. Also, a representative *descriptor*, which is a normalized string value, is generated from the chunks in the node. Certain nodes are also annotated with one or more *types*. There are three families of types: Events (verbs that describe an action, annotated with tense, polarity and aspect); standardized Time Expressions; and Named Entities, with additional annotations such as gender or age.

Edges in the document graph, $e \in E_D$, represent four kinds of relations between the nodes:

- Syntactic: a dependency relation.
- Coreference: indicates that two chunks refer to

[1]Most chunks consist in one word; we join words into a chunk (and a node) in two cases: a multi-word named entity and a verb and its auxiliaries.

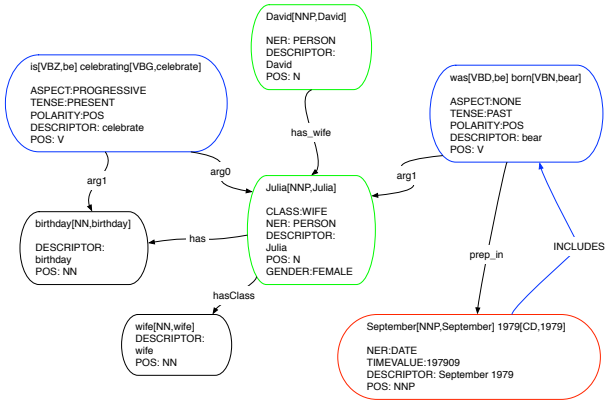

Figure 2: Collapsed document graph representation, G_C, for the sample text document "David's wife, Julia, is celebrating her birthday. She was born in September 1979".

the same *discourse referent*.

- Semantic relations between two nodes, such as `hasClass`, `hasProperty` and `hasAge`.
- Temporal relations between events and time expressions.

The processing includes dependency parsing, named entity recognition and coreference resolution, done with the Stanford CoreNLP software (Klein and Manning, 2003); and events and temporal information extraction, via the TARSQI Toolkit (Verhagen et al., 2005).

The document graph G_D is then further transformed into a *collapsed document graph*, G_C. Each node of G_C clusters together coreferent nodes, representing a *discourse referent*. Thus, a node u in G_C is a cluster of nodes $u_1, \ldots, u_k$ of G_D. There is an edge (u, v) in G_C if there was an edge between any of the nodes clustered into u and any of the nodes $v_1, \ldots, v_{k'}$. The coreference edges do not appear in this representation. Additional semantic information is also blended into this representation: normalization of genitives, semantic class indicators inferred from appositions and genitives, and gender annotation inferred from pronouns. A final graph example can be seen in Figure 2.

4 Distant Supervised Relation Extraction

To perform relation extraction, our proposal follows a distant supervision approach (Mintz et al., 2009), which has also inspired other slot filling systems (Agirre et al., 2009; Surdeanu et al., 2010). We capture long distance relations by introducing a document-level representation and deriving novel features from deep syntactic and semantic analysis.

Seed harvesting. From a reference Knowledge Base (KB), we extract a set of relation triples or *seeds*: $\langle entity, relation, value \rangle$, where the $relation$ is one of the target relations. Our document-level distant supervision assumption is that if entity and value are found in a document graph (see section 3), and there is a path connecting them, then the document expresses the relation.

Relation candidates gathering. From a seed triple, we retrieve candidate documents that contain both the entity and value, within a span of 20 tokens, using a standard IR approach. Then, entity and value are matched to the document graph representation. We first use approximate string comparison to find nodes matching the seed entity. After an entity node has been found we use local breadth-first-search (BFS) to find a matching value and the shortest connecting path between them. We enforce the Named Entity type of entity and value to match a expected type, predefined for the relation.

Our procedure traverses the document graph looking for entity and value nodes meeting those conditions; when found, we generate features for a *positive example* for the relation[2]. If we encounter a node that matches the expected NE type of the relation, but does not match the seed value, we generate a *negative example* for that relation.

Training. From positive and negative examples, we generate binary features; some of them are inspired by previous work (Surdeanu and Ciaramita, 2007; Mintz et al., 2009; Riedel et al., 2010; Surdeanu et al., 2010), and others are novel, taking advantage of our graph representation. Table 1 summarizes our choice of features. Features appearing in less than 5 training examples were discarded.

Relation instance extraction. Given an input entity and a target relation, we aim at finding a filler value for a relation instance. This task is known as Slot Filling. From the set of retrieved documents relevant to the query entity, represented as document graphs,

[2]From the collapsed document graph representation we obtained an average of 9213 positive training examples per slot; from the uncollapsed document graph, a slightly lower average of 8178.5 positive examples per slot.

Feature name	Description
path	dependency path between ENTITY and VALUE in the sentence
X-annotation	NE annotations for X
X-pos	Part-of-speech annotations for X
X-gov	Governor of X in the dependency path
X-mod	Modifiers of X in the dependency path
X-has_age	X is a NE, with an age attribute
X-has_class-C	X is a NE, with a class C
X-property-P	X is a NE, and it has a property P
X-has-Y	X is a NE, with a possessive relation with another NE, Y
X-is-Y	X is a NE, in a copula with another NE, Y
X-gender-G	X is a NE, and it has gender G
V-tense	Tense of the verb V in the path
V-aspect	Aspect of the verb V in the path
V-polarity	Polarity (positive or negative) of the verb V

Table 1: Features included in the model. X stands for ENTITY and VALUE. Verb features are generated from the verbs, V, identified in the path between ENTITY and VALUE.

we locate matching entities and start a local BFS of candidate values, generating for them an unlabelled example. For each of the relations to extract, a binary classifier (extractor) decides whether the example is a valid relation instance. For each particular relation classifier, only candidates with the expected entity and value types for the relation were used in the application phase. Each extractor was a SVM classifier with linear kernel (Joachims, 2002). All learning parameters were set to their default values.

The classification process yields a predicted class label, plus a real number indicating the margin. We performed an aggregation phase to sum the margins over distinct occurrences of the same extracted value. The rationale is that when the same value is extracted from more than one document, we should accumulate that evidence.

The output of this phase is the set of extracted relations (positive for each of the classifiers), plus the documents where the same fact was detected (*supporting documents*).

5 Temporal Anchoring of Relations

In this section, we propose and discuss a unified methodological approach for temporal anchoring of relations. We assume the input is a relation instance and a set of *supporting documents*. The task is establishing a imprecise temporal anchor interval for the relation.

We present a four-step methodological approach: (1) representation of intra-document temporal information; (2) selection of relevant temporal information for the relation; (3) mapping of the link between relational fact and temporal information into an interval; and (4) aggregation of imprecise intervals.

Temporal representation. The first methodological step is to obtain and represent the available intra-document temporal information; the input is a document, and the task is to identify temporal signals and possible *links* among them. We use the term *link* for a relation between a temporal expression (a date) and an event; we want to avoid confusion with the term *relation* (a relational fact extracted from text).

In our particular implementation:

- We use TARSQI to extract temporal expressions and link them to events. In particular, TARSQI uses the following temporal links: *included*, *simultaneous*, *after*, *before*, *begun_by* or *ended*.
- We focus also on the syntactic pattern [*Event-preposition-Time*] within the lexical context of the candidate entity and value.
- Both are normalized into one from a set of predefined temporal links: *within*, *throughout*, *beginning*, *ending*, *after* and *before*.

Selection of temporal evidence. For each document and relational instance, we have to select those temporal expressions that are relevant.

a. Document-level metadata. The default value we use is the *document creation time* (DCT), if available. The underlying assumption is that there is a *within* link from each fact expressed in the text and the document creation time.

b. Temporal expressions. Temporal evidence comes also from the temporal expressions present in the context of a relation. In our particular implementation, we followed a straightforward approach, looking for the time expression closest in the document graph to the shortest path between the entity and value nodes. This search is performed via a limited depth BFS, starting from the nodes in the path, in order from value to entity.

Mapping of temporal links into intervals. The third step is deciding how a relational fact and its relevant temporal information are themselves related. We have to map this information, expressed in text,

Temporal link	Constraints mapping
Before	$t_4 = first$
After	$t_1 = last$
Within and Throughout	$t_2 = first$ and $t_3 = last$
Beginning	$t_1 = first$ and $t_2 = last$
Ending	$t_3 = first$ and $t_4 = last$

Table 2: Mapping from time expression and temporal relation to temporal constraints.

to a temporal representation. We will use the imprecise anchor intervals described is section 2.

Let T be a temporal expression identified in the document or its metadata. Now, the mapping of temporal constraints depends on the temporal link to the time expression identified; also, the semantics of the event have to be considered in order to decide the *time period* associated to a relation instance. This step is important because the event could refer just to the beginning of the relation, its ending, or both. For instance, it is obvious that having the event *marry* is different to having the event *divorce*, when deciding the temporal constraints associated to the *spouse* relation.

Table 2 shows our particular mapping between temporal links and constraints. In particular, for the default document creation time, we suppose that a relation which appears in a document with creation time d held true at least in that date; that is, we are assuming a *within* link, and we map $t_2 = d$, $t_3 = d$.

Inter-document temporal evidence aggregation. The last step is aggregating all the time constraints found for the same relation and value across different documents. If we found that a relation started after two dates d and d', where $d' > d$, the closest constraint to the real start of the relation is d'. Mapped to temporal constraints, it means that we would choose the biggest t_1 possible. Following the same reasoning, we would want to maximize t_3. On the other side, when a relation started before two dates d_2 and d'_2, where $d'_2 > d_2$, the closest constraint is d_2 and we would choose the smallest t_2. In summary, we will maximize t_1 and t_3 and minimize t_2 and t_4, so we will narrow the margins.

6 Evaluation

We have used for our evaluation the dataset compiled within the TAC-KBP 2011 Temporal Slot Filling Task (Ji et al., 2011). We employed as initial KB the one distributed to participants in the task, which has been compiled from Wikipedia infoboxes. It contains 898 triples $\langle entity, slot\_type, value \rangle$ for 100 different entities and up to 8 different slots (relations) per entity[3]. This gold standard contains the correct responses pooled from the participant systems plus a set of responses manually found by annotators. Each triple has associated a temporal anchor. The relations had to be extracted from a domain-general collection of 1.7 million documents. Our system was one of the five that took part in the task.We have evaluated the overall system and the two main components of the architecture: Relation Extraction, and Temporal Anchoring of the relations. Due to space limitations, the description of our implementation is very concise; refer to (Garrido et al., 2011) for further details.

6.1 Evaluation of Relation Extraction

System response in the relation extraction step consists in a set of triples $\langle entity, slot\_type, value \rangle$. Performance is measured using precision, recall and F-measure (harmonic mean) with respect to the 898 triples in the key. Target relations (slots) are potentially *list-valued*, that is, more than one value can be valid for a relation (possibly at different points in time). Only correct values yield any score, and redundant triples are ignored.

Experiments. We run two different system settings for the relation extraction step. They differ in the document representation used (detailed in section3), in order to empirically assess whether clustering of discourse referents into single nodes benefits the extraction. In SETTING 1, each document is represented as a document graph, G_D, while in SETTING 2 collapsed document graph representation, G_C, is employed.

Results. Results are shown in Table 3 in the column *Relation Extraction*. Both settings have a similar performance with a slight increase in the case of graphs with clustered referents. Although precision is close to 0.5, recall is lower than 0.1. We have studied the limits of the assumptions our approach

[3]There are 7 *person* relations: cities_of_residence, stateorprovinces_of_residence, countries_of_residence, employee_of, member_of, title, spouse, and an *organization* relation: top_members/employees.

is based on. First, our standard retrieval component performance limits the overall system's. As a matter of example, if we retrieve the first 100 documents per entity, we find relevant documents only for 62% of the triples in the key. This number means that no matter how good relation extraction method is, 38% of relations will not be found.

Second, the *distant supervision assumption* underlying our approach is that for a seed relation instance $\langle entity, relation, value \rangle$, any textual mention of *entity* and *value* expresses the *relation*. It has been shown that this assumption is more often violated when training knowledge base and document collection are of different type, e.g. Wikipedia and news-wire (Riedel et al., 2010). We have realized that a more determinant factor is the relation itself and the type of arguments it takes. We randomly sampled 100 training examples per relation, and manually inspected them to assess if they were indeed mentions of the relation. While for the relation *cities_of_residence* only 30% of the training examples are expressing the relation, for *spouse* the number goes up to 59%. For *title*, up to 90% of the examples are correct. This fact explains, at least partially, the zeros we obtain for some relations.

6.2 Evaluation of Temporal Anchoring

Under the evaluation metrics proposed by TAC-KBP 2011, if the value of the relation instance is judged as correct, the score for temporal anchoring depends on how well the returned interval matches the one provided in the key. More precisely, let the correct imprecise anchor interval in the gold standard key be $S_k = (k_1, k_2, k_3, k_4)$ and the system response be $S = (r_1, r_2, r_3, r_4)$. The absence of a constraint in t_1 or t_3 is treated as a value of $-\infty$; the absence of a constraint in t_2 or t_4 is treated as a value of $+\infty$. Then, let $d_i = |k_i - r_i|$, for $i \in 1, \ldots, 4$, be the difference, a real number *measured in years*. The score for the system response is:

$$Q(S) = \frac{1}{4} \sum_{i=1}^{4} \frac{1}{1 + d_i}$$

The score for a target relation $Q(r)$ is computed by summing $Q(S)$ over all unique instances of the relation whose value is correct. If the gold standard contains N responses, and the system output M responses, then precision is: $\mathrm{P} = Q(r)/M$, and recall: $\mathrm{R} = Q(r)/N$; F_1 is the harmonic mean of P and R.

Experiments. We evaluated two different settings for the temporal anchoring step; both use the collapsed document graph representation, G_C (SETTING 2). The goal of the experiment is twofold. First, test the strength of the *document creation time* as evidence for temporal anchoring. Second, test how hard this metadata-level baseline is to beat using contextual temporal expressions.

The SETTING 2-I assumes a *within* temporal link between the document creation time and any relation expressed inside the document, and aggregates this information across the documents that we have identified as supporting the relation. The SETTING 2-II considers documents content in order to extract temporal links from the context of the text that expresses the relation. If no temporal expression is found, the date of the document is used as default. Temporal links from all supporting documents are mapped into intervals and aggregated as detailed in section 5.

The performance on relation extraction is an upper bound for temporal anchoring, attainable if temporal anchoring is perfect. Thus, we also evaluate the temporal anchoring performance as the percentage the final system achieves with respect to the relation extraction upper bound.

Results. Results are shown in Table 3 under column *Temporal Anchoring*. They are low, due to the upper bound that error propagation in candidate retrieval and relation extraction imposes upon this step: temporally anchoring alone achives 69% of its upper bound. This value corresponds to the baseline SETTING 2-I, showing its strength. The difference with SETTING 2-II shows that this baseline is difficult to beat by considering temporal evidence inside the document content. There is a reason for this. The temporal link mapping into time intervals does not depend only on the type of link, but also on the semantics of the text that expresses the relation as we pointed out above. We have to decide how to transform the link between relation and temporal expression into a temporal interval. Learning a model for this is a hard open research problem that has a strong adversary in the baseline proposed.

	Relation Extraction						Temporal Anchoring							
	SETTING 1			SETTING 2			SETTING 2-I				SETTING 2-II			
	P	R	F	P	R	F	P	R	F	%	P	R	F	%
(1)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(2)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(3)	0.33	0.02	0.03	0	0	0	0	0	0	0	0	0	0	0
(4)	0.22	0.09	0.13	0.29	0.11	0.16	0.23	0.09	0.13	79	0.21	0.08	0.11	72
(5)	0.53	0.13	0.20	0.54	0.12	0.19	0.34	0.07	0.12	63	0.30	0.06	0.11	56
(6)	0.70	0.12	0.20	0.75	0.13	0.22	0.57	0.10	0.16	76	0.50	0.08	0.14	67
(7)	0.50	0.06	0.10	0.50	0.07	0.12	0.29	0.04	0.07	58	0.25	0.04	0.06	50
(8)	0.25	0.04	0.07	0.20	0.04	0.07	0.15	0.03	0.05	75	0.06	0.01	0.02	30
(9)	0.42	0.08	0.14	0.45	0.08	0.14	0.31	0.06	0.10	69	0.27	0.05	0.09	60

Table 3: Results of experiments for each relation: (1) per:stateorprovinces_of_residence; (2) per:employee_of; (3) per:countries_of_residence; (4) per:member_of; (5) per:title; (6) org:top_members/employees; (7) per:spouse; (8) per:cities_of_residence; (9) overall results (calculated as a micro-average).

System	# Filled	Precision	Recall	F1
BLENDER2	1206	0.1789	0.3030	0.2250
BLENDER1	1116	0.1796	0.2942	0.2231
BLENDER3	1215	0.1744	0.2976	0.2199
IIRG1	346	0.2457	0.1194	0.1607
Setting 2-1	**167**	**0.2996**	**0.0703**	**0.1139**
Setting 2-2	**167**	**0.2596**	**0.0609**	**0.0986**
Stanford 12	5140	0.0233	0.1680	0.0409
Stanford 11	4353	0.0238	0.1453	0.0408
USFD20112	328	0.0152	0.0070	0.0096
USFD20113	127	0.0079	0.0014	0.0024

Table 4: System ID, number of filled responses of the system, precision, recall and F measure.

6.3 Comparative Evaluation

Our approach was compared with the other four participants at the KBP Temporal Slot Filling Task 2011. Table 4 shows results sorted by F-measure in comparison to our two settings (described above). These official results correspond to a previous dataset containing 712 triples[4].

As shown in column *Filled* our approach returns less triples than other systems, explaining low recall. However, our system achieves the highest precision for the complete task of temporally anchored relation extraction. Despite low recall, our system obtains the third best F_1 value. This is a very promising result, since several directions can be explored to consider more candidates and increase recall.

7 Related Work

Compiling a Knowledge Base of temporally anchored facts is an open research challenge (Weikum et al., 2011). Despite the vast amount of research focusing on understanding temporal expressions and their relation to events in natural language, the complete problem of temporally anchored relation extraction remains relatively unexplored. Also, while much research has focused on single-document extraction, it seems clear that extracting temporally anchored relations needs the aggregation of evidences across multiple documents.

There have been attempts to extend an existing knowledge base. Wang et al. (2010) use regular expressions to mine Wikipedia infoboxes and categories and it is not suited for unrestricted text. An earlier attempt (Zhang et al., 2008), is specific for business and difficult to generalize to other relations. Two recent promising works are more related to our research. Wang et al. (2011) uses manually defined patterns to collect candidate facts and explicit dates, and re-rank them using a graph label propagation algorithm; their approach is complementary to ours, as our aim is not to harvest temporal facts but to extract the relations in which a query entity takes part; unlike us, they require entity, value, and a explicit date to appear in the same sentence. Talukdar et al. (2012) focus on the partial task of temporally anchoring already known facts, showing the usefulness of the document creation time as temporal signal, aggregated across documents.

Earlier work has dealt mainly with partial aspects of the problem. The TempEval community focused on the classification of the temporal links between pairs of events, or an event and a temporal expression; using shallow features (Mani et al., 2003; Lapata and Lascarides, 2004; Chambers et al., 2007), or syntactic-based structured features (Bethard and Martin, 2007; Puşcaşu, 2007; Cheng et al., 2007).

Aggregating evidence across different documents

[4]Slot-fillers from human assessors were not considered

to temporally anchor facts has been explored in settings different to Information Extraction, such as answering of definition questions (Paşca, 2008) or extracting possible dates of well-known historical events (Schockaert et al., 2010).

Temporal inference or reasoning to solve conflicting temporal expressions and induce temporal order of events has been used in TempEval (Tatu and Srikanth, 2008; Yoshikawa et al., 2009) and ACE (Gupta and Ji, 2009) tasks, but focused on single-document extraction. Ling et al. (2010), use cross-event joint inference to extract temporal facts, but only inside a single document.

Evaluation campaigns, such as ACE and TAC-KBP 2011 have had an important role in promoting this research. While ACE required only to identify time expressions and classify their relation to events, KBP requires to infer explicitly the start/end time of relations, which is a realistic approach in the context of building time-aware knowledge bases. KBP represents an important step for the evaluation of temporal information extraction systems. In general, the participant systems adapted existing slot filling systems, adding a temporal classification component: distant supervised (Chen et al., 2010; Surdeanu et al., 2010) on manually-defined patterns (Byrne and Dunnion, 2010).

8 Conclusions

This paper introduces the problem of extracting, from unrestricted natural language text, relational knowledge anchored to a temporal span, aggregating temporal evidence from a collection of documents. Although compiling time-aware knowledge bases is an important open challenge (Weikum et al., 2011), it has remained unexplored until very recently (Wang et al., 2011; Talukdar et al., 2012).

We have elucidated the two challenges of the task, namely relation extraction and temporal anchoring of the extracted relations.

We have studied how, in a pipeline architecture, the propagation of errors limits the overall system's performance. The performance attainable in the full task is limited by the quality of the output of the three main phases: retrieval of candidate passages/ documents, extraction of relations and temporal anchoring of those.

We have also studied the limits of the distant supervision approach to relation extraction, showing empirically that its performance depends not only on the nature of reference knowledge base and document corpus (Riedel et al., 2010), but also on the relation to be extracted. Given a relation between two arguments, if it is not dominant among textual expressions of those arguments, the distant supervision assumption will be more often violated.

We have introduced a novel graph-based document level representation, that has allowed us to generate new features for the task of relation extraction, capturing long distance structured contexts. Our results show how, in a document level syntactic representation, it yields better results to collapse coreferent nodes.

We have presented a methodological approach to temporal anchoring composed of: (1) intra-document temporal information representation; (2) selection of relation-dependent relevant temporal information; (3) mapping of temporal links to an interval representation; and (4) aggregation of imprecise intervals.

Our proposal has been evaluated within a framework that allows for comparability. It has been able to extract temporally anchored relational information with the highest precision among the participant systems taking part in the competitive evaluation TAC-KBP 2011.

For the temporal anchoring sub-problem, we have demonstrated the strength of the document creation time as a temporal signal. It is possible to achieve a performance of 69% of the upper-bound imposed by relation extraction by assuming that any relation mentioned in a document held at the document creation time (there is a *within* link between the relational fact and the document creation time). This baseline has proved stronger than extracting and analyzing the temporal expressions present in the document content.

Acknowledgments

This work has been partially supported by the Spanish Ministry of Science and Innovation, through the project Holopedia (TIN2010-21128-C02), and the Regional Government of Madrid, through the project MA2VICMR (S2009/TIC1542).

References

Eneko Agirre, Angel X. Chang, Daniel S. Jurafsky, Christopher D. Manning, Valentin I. Spitkovsky, and Eric Yeh. 2009. Stanford-UBC at TAC-KBP. In *TAC 2009*, November.

James F. Allen. 1983. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26:832–843, November.

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*, SemEval '07, pages 129–132, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lorna Byrne and John Dunnion. 2010. UCD IIRG at TAC 2010 KBP Slot Filling Task. In *Proceedings of the Third Text Analysis Conference (TAC 2010)*. NIST, November.

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*, ACL '07, pages 173–176, Stroudsburg, PA, USA. Association for Computational Linguistics.

Zheng Chen, Suzanne Tamang, Adam Lee, Xiang Li, Wen-Pin Lin, Matthew Snover, Javier Artiles, Marissa Passantino, and Heng Ji. 2010. CUNY-BLENDER TAC-KBP2010: Entity linking and slot filling system description. In *Proceedings of the Third Text Analysis Conference (TAC 2010)*. NIST, November.

Yuchang Cheng, Masayuki Asahara, and Yuji Matsumoto. 2007. Naist.japan: temporal relation identification using dependency parsed tree. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, pages 245–248, Stroudsburg, PA, USA. Association for Computational Linguistics.

Guillermo Garrido, Bernardo Cabaleiro, Anselmo Peas, varo Rodrigo, and Damiano Spina. 2011. A distant supervised learning system for the TAC-KBP Slot Filling and Temporal Slot Filling Tasks. In *Text Analysis Conference, TAC 2011 Proceedings Papers*.

Prashant Gupta and Heng Ji. 2009. Predicting unknown time arguments based on cross-event propagation. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, ACLShort '09, pages 369–372, Stroudsburg, PA, USA. Association for Computational Linguistics.

Heng Ji, Ralph Grishman, and Hoa Trang Dang. 2011. Overview of the tac2011 knowledge base population track. In *Text Analysis Conference, TAC 2011 Workshop, Notebook Papers*.

T. Joachims. 2002. *Learning to Classify Text Using Support Vector Machines – Methods, Theory, and Algorithms*. Kluwer/Springer. We used Joachim's SVMLight implementation available at `http://svmlight.joachims.org/`.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *ACL 2003*, pages 423–430.

Mirella Lapata and Alex Lascarides. 2004. Inferring sentence-internal temporal relations. In *HLT 2004*.

Xiao Ling and Daniel S. Weld. 2010. Temporal information extraction. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10)*.

Inderjeet Mani, Barry Schiffman, and Jianping Zhang. 2003. Inferring temporal ordering of events in news. In *NAACL-Short'03*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *ACL 2009*, pages 1003–1011, Stroudsburg, PA, USA. Association for Computational Linguistics.

M Paşca. 2008. Answering Definition Questions via Temporally-Anchored Text Snippets. *Proc. of IJCNLP2008*.

Georgiana Puşcaşu. 2007. Wvali: temporal relation identification by syntactico-semantic analysis. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, pages 484–487, Stroudsburg, PA, USA. Association for Computational Linguistics.

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*, DEW '09, pages 112–116, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In José Balcázar, Francesco Bonchi, Aristides Gionis, and Michèle Sebag, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 6323 of *LNCS*, pages 148–163. Springer Berlin / Heidelberg.

Stuart J. Russell and Peter Norvig. 2010. *Artificial Intelligence - A Modern Approach (3. internat. ed.)*. Pearson Education.

Steven Schockaert, Martine De Cock, and Etienne Kerre. 2010. Reasoning about fuzzy temporal information from the web: towards retrieval of historical events. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 14:869–886.

Mihai Surdeanu and Massimiliano Ciaramita. 2007. Robust information extraction with perceptrons. In *ACE07*, March.

Mihai Surdeanu, David McClosky, Julie Tibshirani, John Bauer, Angel X. Chang, Valentin I. Spitkovsky, and Christopher D. Manning. 2010. A simple distant supervision approach for the tac-kbp slot filling task. In *Proceedings of the Third Text Analysis Conference (TAC 2010)*, Gaithersburg, Maryland, USA, November. NIST.

Partha Pratim Talukdar, Derry Wijaya, and Tom Mitchell. 2012. Coupled temporal scoping of relational facts. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining (WSDM)*, Seattle, Washington, USA, February. Association for Computing Machinery.

Marta Tatu and Munirathnam Srikanth. 2008. Experiments with reasoning for temporal relations between events. In *COLING'08*.

Marc Verhagen, Inderjeet Mani, Roser Sauri, Robert Knippen, Seok Bae Jang, Jessica Littman, Anna Rumshisky, John Phillips, and James Pustejovsky. 2005. Automating temporal annotation with TARSQI. In *ACLdemo'05*.

Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz, and James Pustejovsky. 2007. SemEval-2007 task 15: TempEval temporal relation identification. In *SemEval'07*.

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*, EDBT '10, pages 697–700, New York, NY, USA. ACM.

Yafang Wang, Bin Yang, Lizhen Qu, Marc Spaniol, and Gerhard Weikum. 2011. Harvesting facts from textual web sources by constrained label propagation. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, pages 837–846, New York, NY, USA. ACM.

Gerhard Weikum, Srikanta Bedathur, and Ralf Schenkel. 2011. Temporal knowledge for timely intelligence. In Malu Castellanos, Umeshwar Dayal, Volker Markl, Wil Aalst, John Mylopoulos, Michael Rosemann, Michael J. Shaw, and Clemens Szyperski, editors, *Enabling Real-Time Business Intelligence*, volume 84 of *Lecture Notes in Business Information Processing*, pages 1–6. Springer Berlin Heidelberg.

Katsumasa Yoshikawa, Sebastian Riedel, Masayuki Asahara, and Yuji Matsumoto. 2009. Jointly identifying temporal relations with Markov Logic. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, ACL '09, pages 405–413, Stroudsburg, PA, USA. Association for Computational Linguistics.

Qi Zhang, Fabian M. Suchanek, Lihua Yue, and Gerhard Weikum. 2008. TOB: Timely ontologies for business relations. In *11th International Workshop on the Web and Databases, WebDB*.