Under consideration for publication in Knowledge and Information Systems

Tackling Representation, Annotation and Classification Challenges for Temporal Knowledge Base Population

Heng Ji, Taylor Cassidy, Qi Li, Suzanne Tamang

Computer Science Department and Linguistics Department
Queens College and Graduate Center
City University of New York
New York, NY 11367, USA
{hengjicuny, taylorcassidy64, liqiearth, suzanne.tamang}@gmail.com

Abstract.

Temporal Information Extraction (TIE) plays an important role in many Natural Language Processing and Database applications. Temporal Slot Filling (TSF) is a new and ambitious TIE task prepared for the Knowledge Base Population (KBP2011) track of NIST Text Analysis Conference. TSF requires systems to discover temporally bound facts about entities and their attributes in order to populate a structured knowledge base. In this paper we will provide an overview of the unique challenges of this new task and our novel approaches to address these challenges. We present challenges from three perspectives: (1) Temporal information representation: We will review the relevant linguistic semantic theories of temporal information and their limitations, motivating the need to develop a new (4-tuple) representation framework for the task. (2) Annotation acquisition: The lack of substantial labeled training data for supervised learning is a limiting factor in the design of TSF systems. Our work examines the use of multi-class logistic regression methods to improve the labeling quality of training data obtained by distant supervision. (3) Temporal information classification: Another key challenge lies in capturing relations between salient text elements separated by a long context. We develop two approaches for temporal classification and combine them through cross-document aggregation: a flat approach that uses lexical context and shallow dependency features and a structured approach that captures long syntactic contexts by using a dependency path kernel tailored for this task. Experimental results demonstrated that our annotation enhancement approach dramatically increased the speed of the training procedure (by almost 100 times), and that the flat and structured classification approaches were complementary, together yielding a state-of-the-art TSF system.

1. Introduction

The concept of time enables us to communicate and reason about physical and mental change. Propositions that pertain to time can be seen as making reference to scalar quantities. Information analysis has benefitted from representing time in scalar terms,

and mapping events and states into their extensions in a temporal ontology that naturally lends itself to logical reasoning; however, a complete account of the derivation of the ordering of *eventualities*[1] mentioned in discourse and the uncertainty associated with such an ordering must incorporate world knowledge regarding which eventualities are contingent upon and causally related to others, how long they tend to endure, as well as linguistic knowledge. Temporal knowledge and the ability to reason with it is crucial to our understanding of world states.

Many natural language statements in text are temporally qualified, such as those expressing relations (a type of state) between entities, or that events in which they participated have occurred, both of which may be temporally bounded (e.g. those describing attributes such as a person's place of residence or employer, or an organization's members; the duration of a war between two countries, or the precise time at which a plane landed). In addition, it has been estimated that one of every fifty lines of database application code involves a date or time value [65]; and on average, each news document in PropBank [38] includes eight temporal arguments. Temporal Information Extraction (TIE) is of significant interest for a variety of Natural Language Processing (NLP) applications such as Textual Inference [6], Multi-document Text summarization [27], Temporal Event Tracking [17, 35], Template Based Question Answering [1, 62], and Temporal Grounding for Semantic Relations [25].

The extraction of temporal arguments for relations and events has recently received the attention of various research programs such as TempEval [58], U.S. DARPA Machine Reading Program, and commercial applications such as Europe Media Monitor These programs have attracted much participation from researchers with varying interests within the NLP community over the last ten years [1, 39, 13, 9, 10, 16, 17, 82, 30, 35, 43, 25]. Most early work focused on extracting temporal relations from individual documents in isolation. The majority of the proposed methods (e.g. [12, 39, 9, 10] have been developed around the TempEval task [80] using TimeBank [56].

In practice, however, we may need to gather information about entities, in terms of associated eventualities, that are scattered among the documents of a large collection. This requires the ability to identify the relevant documents and to integrate facts - possibly redundant, possibly complementary, possibly in conflict - expressed in these documents. Furthermore, we may want to use the extracted information to augment an existing data base. The Temporal Slot Filling (TSF) task of the NIST Text Analysis Conference (TAC)'s Knowledge Base Population (KBP) track [33] was introduced to address these challenges. The task is to automatically extract the values, or "*slot fills*", of time dependent attribute types, or "*slots*", for people and organizations, from the documents in a corpus. The resulting facts are used to populate a knowledge base (e.g. "*James Parsons*" lived in "*San Diego, CA*" "*from 1999 to 2001*").

Automated KBP systems are powerful because they can populate existing knowledge bases with information scattered throughout a tremendous amount of electronic documents at a rate that is impossible for humans to perform manually. Recent research on TSF advances traditional TIE from a single-document to a cross-document paradigm, so that much richer information can be discovered from large-scale corpora using cross-document aggregation. This new setting presents new challenges pertaining to temporal information representation, annotation acquisition and system design. The goal of this paper is to provide an overview of these challenges, linguistic theories behind various representations, general algorithmic framework/tools, state-of-the-art algorithmic approaches to specific problems, and analysis of evaluation results. We will focus on concrete temporal classification

[1] We refer to events and states collectively as *eventualities*, following [5]

and structured prediction solutions that deal with the lack of data (rich annotations and instance re-labeling to enhance distant supervision) and other specific issues.

Various theories have been proposed to represent temporal information in the literature, but most of the previous work has focused on temporal relations extracted from single documents. Based on the previous representations, it is difficult to construct a temporal knowledge base or a timeline of events across documents. In this paper we will review these theories and describe a 4-tuple representation that was designed for the TSF task to accommodate incomplete information, uncertainty, and different temporal granularities.

The lack of annotated data presents another significant challenge for TSF systems. Distant supervision [50] has been shown to be an effective method for generating training data automatically using a structured knowledge base and an unlabeled unstructured corpus. In comparison to the regular slot filling task in which temporal information is ignored, applying this framework to obtain training data for the TSF task presents two unique challenges: (1) More noisy instances are introduced. In response, we exploit multi-level rich annotations to improve the quality of annotations obtained via distant supervision; and (2) Large amounts of features are required to generalize complicated contexts, which makes learning supervised models unfeasible. In this paper we propose a novel approach based on self-training and regression to reduce features and *re-label* instances. Our experimental results show that this approach can speed up the training process by almost 100 times with slight gains in extraction quality.

The final key challenge of TSF lies in the fact that the relevant textual elements are often separated by long contexts. Syntactic parsing allows us to *compress* long contexts based on underlying dependency structures, revealing common syntactic patterns. However, applying NLP tools to extract a deeper representation of the text tends to introduce errors. Furthermore, in cases where there is a short context, surface features can provide an appropriate generalization. To address this tradeoff we have developed and combined two complementary approaches to the temporal classification problem: a structured approach that captures long syntactic contexts using a dependency path kernel, and a flat approach that exploits surface patterns and coarse grained dependency relations. This hybrid approach outperforms a number of baselines, yielding a state-of-the-art TSF system.

In comparison with previous work, the novel contributions of this paper are as follows:

1. The first work to systematically review various temporal information representation frameworks and assess their pros and cons for temporal slot filling. (Section 2).
2. Analyze the key annotation challenges of distant supervision, and propose two novel methods to address these challenges in the temporal KBP setting (Section 4).
3. Propose a hybrid framework consisting of both a structured and flat approach to achieve state-of-the-art TSF performance (Section 5).

2. Temporal Information Representation Theories

2.1. General Goal

About what sorts of things do we ask, "*when?*". We can certainly ask, "*when was it the case that ϕ?*", where ϕ is a proposition asserting the existence of an eventuality. Suppose that ϕ = "*George Bush is the President of the United States*"; then, in this case our eventuality is a state, and possible responses include, ‘*‘before Barack Obama (was President of the United States)*", "*From 2000 to 2008*", "*after he was elected*", "*upon*

being sworn in", "*on January 1st, 2003, at least*", "*until at least 2004*", "*for at least four years*", "*over 4 years ago*". Intuitively, were ϕ = "*George bush was elected president*", possible responses would be systematically related to those above. All of these answers share the property that they serve to assign ϕ a location or *extension* in time. Some are relative to eventualities, while others are specified explicitly with *temporal expressions*, which denote regions in time with respect to some a unit of temporal measurement. In this paper we will consider standard units given by the calendar and clock. The capacity to produce an answer derived from linguistically represented information, including discourse, given background knowledge, from which the answer is accessible, is achieved via a process of reasoning that operates over some representation of what happens, and when. In this section we describe theories of how such information is represented and processed, focusing on how they relate to TSF.

2.2. Semantics of Eventualities

Formal *reification* of eventualities came out of attempts to define the logical form of action sentences.[2] The *neo-Davidsonian* analysis says that a proposition about an eventuality (i.e. an event or state) asserts that it exists, and that it is predicated of some predicate denoted by a verb. The treatment of eventualities are objects, analogies between semantic and syntactic analysis of eventuality structure, and verbal tense and aspect has lead to a better understanding of how what happens in the real world is described in natural language.

[21] tried to account for the intuition that the entailment patterns between the sentences below should be expressible in terms of their meanings:

1. Brutus stabbed Caesar
2. Brutus stabbed Caesar violently
3. Brutus stabbed Caesar with a knife
4. Brutus stabbed Caesar violently with a knife

The *Davidsonian* analysis proposed that the logical form in which the fact that an event occurred is expressed with existential quantification over event objects, and that the predicate denoted by the verb includes a place for an *event argument*, in addition to other necessary arguments. Modification of a core VP (e.g. *Brutus stabbed Caesar*) is expressed in the logical form of the sentence by conjunctive applications of predicates of events and objects (as shown in (4)).

(4) $\exists e[\text{Stab}(B, C, e) \wedge \text{violent}(e) \wedge \text{with}(K, e)]$

(4′) $\exists e[\text{Stab}(B, C, e, \text{violent,a knife})]$

Previous accounts would express modification by adding additional places in the Stab predicate as in 4′. Given the representation in (4) the correct entailments follow from simple conjunctive reasoning:

$(2) \rightarrow (1), (3) \rightarrow (1), (4) \rightarrow (2) \wedge (3)$, but $(2) \wedge (3) \not\rightarrow (4)$).

Parsons' theory of underlying events extends and modifies Davidson's, so that the logical form of (4) would be 4^p, a *neo-Davidsonian* analysis [53].

(4^p) $\exists e[\text{Stabbing}(e) \wedge \text{Agent}(B, e) \wedge \text{Theme}(C, e) \wedge \text{violently}(e)$
$\wedge \text{with}(K, e)]$

The theory is supported by its utility in explaining phenomena such as adverbial modification, nominalization of events, perceptual idioms, causative and inchoative

[2] We assume throughout that eventualities as concepts correspond with situations in the physical world.

constructions, transitive and intransitive verbs, the progressive tense, the imperfective paradox, and temporal modifiers.

In what follows we assume a neo-Davidsonian treatment of eventualities, including states (but see [37]). The idea particular eventualities are things, the predicates of which can take various forms in text, and can stand in a relation to other eventualities and entities provides for an intuitive description of IE: detecting mentions[3]of eventualities, and discovering their relations to detected mentions of entities. Then, TIE consists of organizing these structures in a particular way - with respect to time.

2.3. Grounding Eventualities in Time

The task of ordering eventualities expressed in text in a temporal ontology constitutes an overlap between semantics, Artificial Intelligence (AI), and NLP. Semantics aims to formalize what sort of information is encoded in grammatical elements like tense, lexical aspect, temporal expressions, and adverbial modifiers, and how it is used to map eventualities to their temporal extensions. Although semantics need not specify which interpretation of a text conforms with the "normal course of events" derived from real world knowledge, it should specify the dimensions along which such interpretations can be made. Given a mapping between eventualities and their associated time intervals, AI theory aims to specify how relations not explicitly represented in discourse can be inferred, including those inferred from combining information from multiple discourses, as well as how to incorporate world knowledge. The more precise the mapping from eventuality to interval, the more effective such processes will be. Statistical NLP aims to specify a machine readable format (e.g. an annotation scheme) in which linguistic information that is relevant to the task at hand can be represented; of course, not all information deemed relevant will be worth representing due to difficulties and costs, of both annotation and implementation. Statistical models can then be learned that aim to replicate a task that humans perform (e.g. temporal discourse processing).

There are many ways to verbally express that events occur and that states endure, and when. Eventualities may be distinguished based on properties such as *telicity*, *dynamicity*, and *durativity*. [77] defined four classes of lexical aspect, or Aktionsarten: *states*, *activities*, *accomplishments*, and *achievements*. States exhibit no change while they hold. Activities exhibit change, though their having occurred is not contingent on a particular outcome having been reached. Accomplishments "take time", and do have an obligatory completion condition to be considered as having occurred, while achievements are similar but occur instantaneously. [73] note that tests to distinguish between Aktionsarten based on adverbial modification are available in many languages. The relations in TSF are most accurately considered states, but their having endured at a particular time, or throughout a period of time, can be indicated in a variety of ways based on a multitude of semantic and syntactic cues. Under the lexical aspectual system defined in [22], eventualities may be complex and may be decomposable into constituent eventualities, which may interact with temporal expressions and modifiers differently depending on their type. For example, any *state* is the *result_of* a *change*, which is a durative, dynamic, telic, resultative eventuality. Thus, the beginning of a state of employment may be indicated by a contractual signing which endured over a matter of seconds, which was *culminated_by* a hiring *boundary* (a boundary is a non-durative, intrinsically instantaneous event). Similarly, being employed is akin to

[3] We consider the usage of a predicate of eventualities to be a mention of the eventuality for which it returns a positive truth value.

"*working for company X*", which is exemplified by the *process* built from constituent *eventualities* including, e.g., making announcements on behalf of the company. [76] invokes the concept of *establishing times* to describe how aspect is used to refer to parts of eventualities, drawing an analogy to physical descriptions that refer to parts of objects (e.g. we can say "*the church is East of Main street*" even if the church is South another non-salient part of Main Street). Thus, a good TSF system should be able to learn both that a state endured and when this was the case using clues derived from constituents of that state, or constituents of constituents of that state, etc. Eventualities may be lexicalized in a variety of ways in practice, as alluded to in [77], or as formalized by [22], and as a result it would be quite difficult to enumerate each way in which a given state might be expressed. [4]

A natural reference from which to locate an eventuality's temporal extension is the time of utterance. A clause in the simple past, simple present, or simple future tense about an eventuality indicates its location in time relative to the time of utterance. Information more specific than *before*, *during*, or *after* speech time can be specified using temporal adverbial modifiers and the perfect tenses. Reichenbach's (1947) formalization of the English tense system defines each tense in terms of speech (S), event (E), and reference (R) time. For an event occurrence expressed by a verb, the verb's tense determines the relative position of S, E and R. Simple tenses equate E and R, but perfect tensed clauses locate E with respect to R. For example in *John had lived in New York*, the event time during which *John lives in New York* is asserted to be true is prior to R, which is prior to S.The relative position of eventualities in a compound sentence is determined by comparing reference times, but the role played in these processes by aspectual class is not defined.

Reichenbach's ideas were later combined with interval semantics [72] and a better understanding of temporal anaphora [54] and aspectual class to formalize the process by which the order of eventualities is inferred while a discourse is processed. [55, 31, 26] focus on the progression of reference time through a narrative, in which the default assumption is that eventualities in consecutive clauses in the simple past may not overlap in time unless one or both are stative. [55] proposes a theory in which the way reference times are associated with temporal anaphora is formally analogous to the way in which nominal anaphora bind their referents. Discourse representation theory (DRT) [36] is extended to treat eventualities as primitive elements that can be ordered with respect to reference times by $<$ and $\subseteq$. [31] provides a DRT based formal depiction of how aspectual class of a main clause verb, temporal conjunctions, and temporal adverbs can move the reference time in a discourse. A discourse is represented as an *event structure*, built by processing sentences in order, ordering any newly introduced E with respect to R and S via *before* and *inclusion* relations, using rules that make explicit reference to an eventuality's aspectual class and whether the progressive is used. [26] argues that the aspectual class of a clause is based on the compositional semantics of the verbs, their modifiers, and arguments, as opposed to their syntactic relations. Previous DRT based theories require a complete discourse representation structure (DRS) before the compositional semantic interpretation of a discourse can be specified. But in these theories eventualities are ordered in time as their containing clauses are added to the DRS; thus, if semantic properties specify aspectual class, they could not possibly be used in building the DRS. To avoid circularity an aspect-independent interpretation

[4] Vendler alluded to the fact that not only verbs, but adjectives and nouns may be used to express eventualities as well. Dölling gave a formal account of aspectual coercion, in which the canonical conceptualization of an event (modifier of an event) may be adjusted based on factors such as which modifier (event) is applied to it (it is applied to), as well as world knowledge.

principle is proposed for consecutive sentences within a discourse, defining their aspectual class in terms of interval semantics.

The temporal discourse interpretation principle (TDIP) [26] says that given a sequence $S_1, S_2, ..., S_n$, the reference time for S_i is either:

1. A time consistent with the time adverbs in S_i or
2. A time which immediately follows the reference time of S_{i-1}

This principle is not as complex nor as precise as what Hinrichs or Partee propose, but Dowty makes a good case that much of what remains to be specified during a particular instance of temporal discourse processing is outside the realm of semantics.

World knowledge must play a role in the temporal processing of discourse. As indicated above, formal semantics should specify what role is played by world knowledge and discourse context. [31] conforms to this principle by specifying possible temporal orderings of eventualities for a variety of particular cases. States may fail to overlap an eventuality in the previous sentence due to world knowledge, as in Hinrichs' example:

Jameson entered the room, shut the door carefully and switched off the light. It was pitch dark around him because the Venetian blinds were closed.

Only knowing that turning off a light switch often leads to a dark room allows the inference that the light-switching event is *before* the state of pitch darkness. [55] postulates that world knowledge may dictate the value of a higher-order parameter that controls discourse relations which in turn may help to resolve ordering ambiguity. [26] claims that world knowledge may be necessary to determine aspectual class of verb phrases and clauses (see also [51] regarding aspectual class *coercion*), the degree to which the reference time "moves forward" when a new eventuality is introduced is determined using world knowledge, and the extent to which the interval associated with an eventuality is assumed to be larger (in either direction) than what is explicitly asserted. [40] formalize rules relating Gricean maxims and world knowledge in which possible eventuality orderings are disambiguated based on discourse relations between clauses using defeasible logic. [32] claim that any robust system based on this method would likely be computationally unfeasible. While Hitzeman et al's framework doesn't rely on postulates and allows tense, aspect, discourse relations and world knowledge to be mutually constraining, their world knowledge representation is weaker.The formal semantic accounts described above provide a number of considerations for researchers in TIE, including:

1. Formal account of how in a narrative reference time moves forward, states overlap and other eventualities don't, by default.
2. Determining the aspectual class of an eventuality may require world knowledge and how event time (E) and reference time (R) are related.
3. Choosing between possible orderings of E and R times, durations of intervals that separate eventualities, perceived starting and ending points of intervals as well as degree of overlap may require world knowledge.
4. Discourse relations between clauses may determine default eventuality ordering across consecutive sentences.

Thus, the strategy of mapping all eventualities to an extension in time and subsequently applying logical reasoning over a temporal ontology, all without very specific world knowledge is ill-founded, since world knowledge and reasoning about order is needed to create such a mapping. Furthermore, it is rarely the case that an entire eventuality is grounded in a fully specified temporal interval in a single document, let alone a single sentence (though a short biography is a notable exception). Recognition of this obstacle, however, is not meant to discount the potential power of interval-based temporal reasoning.

Temporal relations among eventualities can be formally stated in terms of intervals on the timeline. [2] formalizes 13 interval relations (the union of seven basic relations and their inverses). That an Allen relation holds between two intervals can serve to specify a relation between two temporal expressions, an eventuality and a temporal expression, or two eventualities. For example, let $\tau : E \cup T \rightarrow I$, where R is an Allen relation. Then τ maps an eventuality ($e \in E$) to the interval in which it occurs or obtains, and a temporal expression ($t \in T$) to the interval it denotes. There's a sense in which τ is an isomorphism between ($E \cup T$, "before") and ($I, <$): e_a can be said to have occurred before e_b if and only if $\tau(e_a) < \tau(e_b)$. For example, in "*Smith stepped down as President before founding the charitable organization in September of 1993*," the relationship between the text and the timeline is straightforward. We think it is clear that such relationships hold for the other Allen relations, and the mapping from $(E \cup T, R)$ to (I, A) is somewhat trivial; however their utility is contingent upon the ability to map linguistic data to formal expressions of the form, $x_a R x_b$, where $x_a, x_b \in E \cup T$.The example above is a canonical form. We've seen how other derivations of $x_a R x_b$ might require devices such as tense, aspect, discourse structure, pragmatic considerations and world knowledge. But in addition, some facts of the form $x_a R x_b$ are implicit in the sense that they are not associated with any overt linguistic form.

Consider the task of implementing a computational system whose aim is to order eventualities based on a corpus of newspaper articles. Semantic temporal discourse processing theories may provide some helpful tools; however, they provide little guidance as to straightforward incorporation of world knowledge, and news reporting tends to deviate from the narrative structure both in order of presentation of facts [7] (events are mentioned in order of importance as opposed to approximately linear order).

The TimeML annotation scheme [56] is used to label eventualities and specify their attributes, including their temporal extent, their tense, grammatical aspect, polarity, modality, class (e.g. *occurrence*, *state*, *intentional state*). In addition, phrases that specify Allen relations holding between events, and mapping between time expressions and intervals, are annotated. Together these labels implicitly subsume ordering information in terms of Allen relations in a calendar ontology. TimeML is arguably the standard temporal markup language, and has been used to annotate standard corpora used for temporal information extraction tasks [57, 63, 64]. TimeML provides a means by which statistical models can learn to ground eventualities in time given the context in which they occur, incorporating both interval based reasoning and linguistic features. World knowledge, however, is not explicitly encoded.

TimeML's treatment of states could not fully accommodate the TSF framework. While *"state"* is a possible value of the *"class"* attribute of the *"event"* tag, only states whose initiation, culmination, or change are annotated. TSF requires that examples like *"President Obama spoke at the commencement In July, 2001"* be viewed as conveying information about a portion of a state of being president. It is possible to recover the temporal extension of a state given only times at which it holds as well as the bounds of related states, given information about state ordering as well as world knowledge, so explicit indications of a state's beginning and end are not required. For example, if we know that the United States always has a president, and that the last three presidents were Clinton, Bush, and Obama, if we know Clinton's end time and Obama's start time, we can infer Bush's start and end times.

Interval based reasoning has been applied as a means of enriching annotated corpora labeled with temporal relations between eventualities by adding labels for implicit relations [71, 45, 46, 78]. [45, 46] described the application of a statistical learning procedure to predict some Allen relations in news documents. Depending on evaluation context the statistical method either outperforms or underperforms a method based on hand-coded

rules written in terms of TimeML attributes and part-of-speech (POS) tags, along with lexical patterns from VerbOcean, but only after augmenting the training data using interval based reasoning. Their results improved after incorporating linguistic information: knowing that eventuality mentions shared the same agent, referred to the same eventuality type via different verbs, were a part of the same sentence, and the tense of surrounding eventualities were all helpful features. [39] showed that linguistic features help to classify ordering of eventualities within the same sentence, and that verbs' *supersenses* from WordNet [28] might be a good proxy for certain world knowledge, but noted that aspectual class will be difficult to identify automatically. [24] note that most systems in the literature reporting high scores make use of either gold standard event-to-time mapping, or operate in a setting in which this mapping is not very difficult. This is most likely due to the close interaction between linguistic cues and world knowledge.

2.4. Temporal Slot Filling

TSF can be seen as the extraction of facts in the form of statements about the temporal extensions of *consequent states* of eventualities. The experiments reported above, as well as linguistic theories, do not explicitly address the aggregation of temporal information about eventualities across discourses, though in practice much important information is gained in this way. Some fact of the form, "x stands in relation r with y during the interval beginning at time t_{init} and lasting until time t_{end}" may be recoverable from a news corpus yet never explicitly stated in its entirety.

The Automatic Content Extraction (ACE) program involved temporal argument extraction for relations and events. Some recent work aimed to extract temporal facts from large-scale texts and couple these facts with an existing temporal knowledge base [81, 68]. Although temporal event trigger ordering has been extensively studied in the TempEval task [58] and various approaches have been developed [8, 15], detecting temporal boundaries requires us to classify the way in which temporal expressions are associated with specific entity, slot value pairs. Finally, though the Allen interval representation is ultimately adequate, it cannot support intermediate under-specified temporal information about beginning and endpoints which will necessarily be expressed when facts are grounded in time across documents. The new Temporal Slot Filling (TSF) task of the TAC-KBP was formulated as a step toward filling these gaps.

The TSF task requires that a system specify, to the most specific extent possible given the context from which a given fact can be extracted, the time during which the fact holds. A TSF system should add temporal (duration) information to the following slots: SPOUSE_OF, TITLE_OF, EMPLOYEE_OF, MEMBER_OF, CITIES_OF_RESIDENCE, STATES OR PROVINCES_OF_RESIDENCE and COUNTRIES_OF_RESIDENCE for people, and the TOP_EMPLOYEES/MEMBERS slot for organizations.

KBP temporal representation [3] makes use of tuples of the form $T =< t_1, t_2, t_3, t_4 >$, which can be seen as generating a set:

$$S = \{< t_{init}, t_{end} > |(t_1 < t_{init} < t_2) \wedge (t_3 < t_{end} < t_4)\}$$

Where all t are dates. In other words, t_1 and t_2 represent the lower and upper bounds for the beginning of a relation, while t_3 and t_4 represent the lower and upper bounds for the end of the relation.

This temporal representation model can accommodate temporal aggregation, temporal relations between eventualities and times [2], and temporal relations between two eventualities when one of them is anchored in time. It also provides a straightforward method to detect inconsistencies when aggregating temporal information in a tuple.

Table 1. 4-tuple Representation Example

Document text	T1	T2	T3	T4
Chairman Smith	-	20010101	20010101	-
Smith, who has been chairman for two years	-	19990101	20010101	-
Smith, who was named chairman two years ago	19990101	19990101	19990101	-
Smith, who resigned last October	-	20001001	20001001	20001031
Smith served as chairman for 7 years before leaving in 1991	19840101	19841231	19910101	19911231
Smith was named chairman in 1980	19800101	19801231	19800101	-

Table 1 presents some examples of 4-tuple representation, assuming the publication date of the text is January 1, 2001.

The main limitation of assuming that events are continuous is that our representation model is not able to capture certain eventuality structures such as regularly recurring events (*"each Friday"*), some fuzzy relations (*"lately"*, *"recently"*) that are encoded with the SET type in TimeML [56], durations where neither endpoint is known (*"he worked for IBM for 7 years"*), and relations between slots (*''she married Fred two years after moving to Seattle"*). Because eventualities of the same type with the same participants are not distinguished, those which are true over multiple disjoint intervals (*"Cleveland was President from 1885 to 1889 and from 1893 to 1897"*) cannot be accommodated; moreover, because only binary relations are considered, the same slot value (*"President"*) affiliated with different entities will not be accommodated (*"Mary was the President of Student Union from 1998 to 2003 and the President of Woman Sports Association from 2002 to 2005"*).

Given a query entity, a knowledge base (KB) and a source corpus, a system must return slot fills and temporal information must be gathered across the entire corpus. There are two subtasks: full and diagnostic. For the full temporal task, the system is given an entity name and a document where this name is mentioned and is expected to find the relevant slot fills using the entire document collection. For example, given the following query:

```
⟨query id="SFT201"⟩
   ⟨name⟩Angela Merkel⟨/name⟩
   ⟨docid⟩NYT_ENG_20071015.0123.LDC2009T13⟨/docid⟩
   ⟨enttype⟩PER⟨/enttype⟩
   ⟨nodeid⟩E0288830⟨/nodeid⟩
⟨/query⟩
```

A full TSF system should generate the query ID, slot type, temporal tuple element, tuple answer, supporting document ID and slot fill as follows:

- SFT201 per:countries_of_residence t2 20051231 AFP_ENG_20081022.0383 Germany
- SFT201 per:countries_of_residence t3 20081022 AFP_ENG_20081022.0383 Germany
- SFT201 per:spouse t1 19980101 APW_ENG_20051122.0372 Joachim Sauer
- SFT201 per:spouse t2 19981231 APW_ENG_20051122.0372 Joachim Sauer
- SFT201 per:spouse t3 20051122 APW_ENG_20051122.0372 Joachim Sauer

For the diagnostic temporal task, the system is given the entity name and a set of slot,

slot value pairs along with supporting documents. The system only needs to determine the temporal information for each pair, based solely on the information in the provided support document.

3. Approach Overview

In this section we will present an overview of our TSF system pipeline and its basic components. Note that we use the term *query* to refer to either the query entity alone or to a particular query entity/slot pair, *fact* to refer to an instance expressing that a slot relation holds between a query and a slot fill, and *temporal fact* to refer to a fact's relation to a particular time.

3.1. Overall Pipeline

The full TSF system architecture is depicted in Figure 1. For a given query, a regular slot filling system extracts slot fills independent of temporal information. Next, using the query entity and slot fills as search terms the system retrieves relevant documents and annotates them using various NLP techniques (Section 4.3). Sentence retrieval considers not only content relevance but also time richness, namely that the sentence should include the query entity, a slot fill, as well as one or more temporal expressions identified by the time expression extraction and normalization (TIMEX/TIMEML) component. The remaining processing can be decomposed into two problems: (1) for a given sentence, the classification of the relationship between each temporal expression to the fact possibly expressed therein; and (2) temporal aggregation to form a coherent 4-tuple for each slot specified in each query.

3.2. Regular Slot Filling

We apply a high-performing regular slot filling system [19] to identify KBP slot fills for a given query. For example, it will extract "*Joachim Sauer*" as the slot fill for the SPOUSE of the query entity "*Angela Merkel*". This system includes a supervised Information Extraction (IE) pipeline, a pattern matching pipeline and a top-down question answering (QA) pipeline.

The supervised IE pipeline includes entity extraction, relation extraction and event extraction based on Maximum Entropy models that incorporate diverse lexical, syntactic, semantic and ontological knowledge [34]. The extracted ACE relations and events are then mapped to KBP slot fills.

For the pattern matching pipeline, patterns for each slot type were extracted via distant supervision [50] using query/slot fill pairs from Wikipedia Infoboxes and Freebase [11], and ranked based on confidence. Sentences retrieved for evaluation queries are then checked for the presence of patterns whose confidence exceeds a certain threshold in order to extract slot fills for query entities. We set a low threshold to increase recall, yielding more candidate facts, and then applied several filtering steps to remove facts which have inappropriate entity types for a given slot, or involve inappropriate dependency paths containing the entities.

The QA pipeline employs an open domain QA system, OpenEphyra [61], treating answers as candidate slot fills. To estimate the relevance of an answer, we use the Corrected Conditional Probability (CCP) for answer validation.

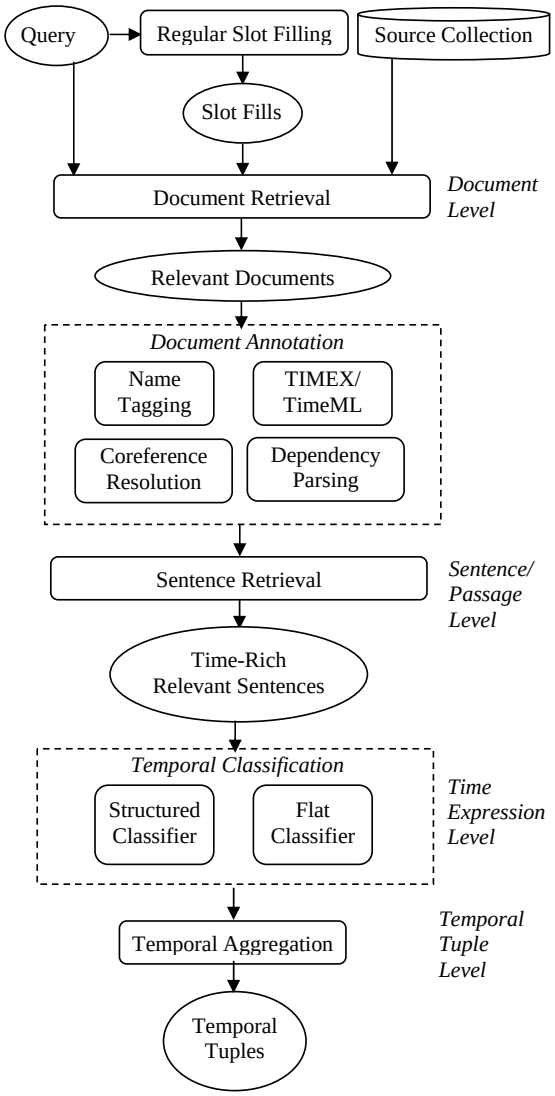


Fig. 1. General Temporal Slot Filling System Architecture

A Maximum Entropy based supervised re-ranking method [70] is then employed to combine the results from these three pipelines. The re-ranking features are defined in terms of external gazetteers, confidence values, dependency parsing paths, majority voting values and slot types.

Finally, we develop a Markov Logic Networks (MLN) [59] based reasoning component to avoid mutually inconsistent slot fills and enforce slot dependencies.

A detailed description of each regular slot-filling pipeline described above can be found in [19].

Table 2. Description of Temporal Classes

Class	Temporal Role	Four tuple
START	beginning of the slot fill	$< t_a, t_b, t_a, \infty >$
END	end of the slot fill	$< -\infty, t_b, t_a, t_b >$
HOLD	a time at which the slot fill is valid	$< -\infty, t_b, t_a, \infty >$
RANGE	a range in which the slot fill is valid	$< t_a, t_a, t_b, t_b >$
NONE	unrelated to the slot fill	$< -\infty, \infty, -\infty, \infty >$

3.3. Temporal Classification

For each candidate fact, relevant sentences are retrieved. Then for each temporal expression in a given sentence, *temporal classification* is applied to the corresponding candidate temporal fact instance, labeling it as exemplifying one of the following temporal classes: "*START*", "*END*", "*HOLDS*", "*RANGE*" or "*NONE*", with respect to the fact in question, as described in Table 2. Take, for example, the following sentence:

"In 1975$_{[time\ expression]}$, after being fired from Columbia amid allegations that he$_{[query\ entity]}$ used company funds to pay for his$_{[query\ entity]}$ son's bar mitzvah, Davis $_{[query\ entity]}$ founded Arista$_{[slot\ fill]}$"

In this case, the temporal classification component should determine that "*1975*" stands in the START relation to the fact EMPLOYEE_OF("*Davis*", "*Arista*").

3.4. Temporal Aggregation

Individual temporal classifications tend to provide incomplete temporal information, and moreover may be contradictory due to classifier error. Therefore, after each temporal fact instance is classified and mapped to a 4-tuple as described in Table 2, we need to aggregate all 4-tuples for a given fact. A fact's final 4-tuple is obtained by a process of aggregation in which the bounds on t_{init} and t_{end} are restricted which approximates the greatest extent licensed by the intermediate temporal classifications assigned to all temporal fact instances. Given two four-tuples T and T' for the same fact, we use the following equation for aggregation.

$$T \wedge T' =$$
$$< max(t_1, t'_1), min(t_2, t'_2), max(t_3, t'_3), min(t_4, t'_4) >$$

At each step we update the tuple only if the result is consistent (i.e. $t_1 \leq t_2$, $t_3 \leq t_4$, and $t_1 \leq t_4$). The process is applied iteratively after 4-tuples are ordered by classifier confidence.

4. Annotation Challenges

Temporal annotation is an extremely expensive and difficult task. In this section we elaborate these problems and propose solutions to obtain reliable training data automatically.

4.1. Limitations of Human Annotation

Quantity Limitations:

Manual annotation of temporal relations is a very expensive task, and annotation for TSF in particular is no exception. For example, [79] reported that annotators of TimeBank [57] documents typically only annotate about 1% of all possible TimeML TLINK tags, wherein event-event or event-time interval relations are specified. In the case of TSF, even after a large team of expert human annotators for put forth several months worth of effort, the KBP2011 official training data only includes only 1,172 instances covering all of the 8 slot types, which is not enough to sufficiently train a supervised temporal classifier. It is worth noting that only 35 out of the 107 queries with *employee_of* answers in this gold standard data set were found to have documents that included potential temporal arguments for that relation, and only one third of the queries could be reliably associated with either a start or end date, with both explicit start and end dates given for only one such query. On average, 518 relevant documents were found for each query entity/slot-fill, but only 21 sentences included a query entity mention, slot fill, and temporal expression. This indicates that explicit temporal information is very sparse in news and web data, even when the scope of data collection is very large. Even for queries with many relevant documents, the temporal information is generally scattered across documents.

Quality Limitations:

A three-way manual annotation of 324 temporal expressions in 281 sentences related to different queries yielded moderate inter-annotator agreement (pairwise Cohen's Kappa of 0.57), reflecting the difficulties of the annotation task. In addition, [79] reported that annotators chose to annotate a temporal relation between the same two temporal entities (events or time expressions) in only 10% of all annotations.

In addition, the manually annotated data include a lot of short sentences with very rich temporal information expressed with simple patterns. Such instances differ fundamentally at the syntactic level in the way tense and temporal adverbial modification is used to modify predicates of eventualities. Moreover, they are hardly representative of the entire source collection, and therefore are not useful for learning new features:

"Tom LaSorda, president and CEO, Sept. 2005-Aug. 2007
Dieter Zetsche, president and CEO, Nov. 2000- Sept. 2005
... Lee A. Iacocca, chairman and CEO, Sept. 1979-Dec. 1992 (president from Nov. 1978-Sept. 1979)
Eugene A. Cafiero, president, Oct. 1975-Nov. 1978

4.2. Distant Supervision for TSF

Given the prohibitive nature of the manual annotation task, we intend to automate the process of acquiring training data.

We adopted a distant supervision approach [50] to obtain a large amount of training data from the Web without human intervention. Distant supervision is a learning paradigm that exploits known relations (usually obtained from an existing database) to extract contexts exemplifying those relations from a large document collection, and automatically label them accordingly. The general intuition is that whenever two entities known to participate in a relation appear in the same context, this context is likely to express this relation in some way. By extracting many such contexts, different ways of expressing the same relation will be captured and a general model may be abstracted by applying machine learning methods to the annotated data.

The TSF task requires the annotation of three elements at once - query entity,

Table 3. Number of Human and Distantly Supervised Training Instances

Category	Type	Total	Start	End	Holds	Range	Others
Spouse	Manual	28	10	3	15	0	9
	Automatic	**10,196**	**2,463**	**716**	**1,705**	**182**	**5,130**
Title	Manual	461	69	42	318	2	30
	Automatic	**14,983**	**2,229**	**501**	**7,989**	**275**	**3,989**
Employment	Manual	592	111	67	272	6	146
	Automatic	**17,315**	**3,888**	**965**	**5,833**	**403**	**6,226**
Residence	Manual	91	2	9	79	0	1
	Automatic	**4,168**	**930**	**240**	**727**	**18**	**2,253**

slot fill and temporal expression. We use Freebase [11] which contains instances of the eight relations of interest, along with the start and end dates of those particular relations. We assume that we can label a context containing a temporal expression, query entity, and candidate slot-fill by comparing the temporal expression to the start/end temporal information that is stored in our database. For instance, Freebase states that "*John*" worked for "*Nissan*" between "*2001*" and "*2009*". Confronted with the sentence, "*Nissan chief engineer John Smith gave a lecture at UCLA on the 12th of December, 2003.*" our distant supervision method compares the temporal expression "*12th of December, 2003*" to the known temporal information in the database. Since this time expression "*2003*" falls in between the "*2001*" and "*2009*", it is assigned a "*HOLDS*" label. Note that our framework essentially treats classification instances as being of the form: $<$SLOT_TYPE(*query*, *slot fill*), T $>$, because the labels (except for "*NONE*") describe how a regular slot filling relation relates to a temporal expression. However, the "*NONE*" label is used to indicate that the complete triadic relation between the query entity, slot fill, and temporal expression, does not hold, even if the regular slot filling relation itself holds.

Table 3 compares the number of temporal relations identified by human annotators in the official TAC KBP corpus with what we were able to retrieve from the Web without human intervention. We can see that our automatic method has obtained substantially more training data (over 40,000 instances).

Furthermore, the major advantage of using Web data to retrieve candidate instances of temporal relations is the diversity of contexts that can be obtained. For example, expressions captured in this larger data set included common patterns "*Alexander and Susan married on Jan. of 2005*", as well less common phrases, "*On September 2002 Mary Jones and John Smith eloped on the SF bay*". It can also include instances which include implied temporal information. For example, the sentence "*In 1997, John and Mary renewed their vows in Florida.*" implicitly indicates that the "*spouse*" relation between "*Mary*" and "*John*" held prior to and on the date alluded to, i.e. some day in 1997.

Thus, the lack of training data for supervised learning is a bottleneck for automated, supervised KBP systems and distant supervision is an effective method to expedite the labeling of training data at low cost. In addition to providing a mechanism to create a labeled training set when no manually annotated data set is available, classifiers trained with distant supervision are less prone to over-fitting than those learned from manual annotations. However, not surprisingly, using such a simple heuristics to label temporal relation instances leads to both quality and scalability problems for training a TSF system. Manual evaluation on a subset of the automatically generated training data shows that only about 75% sentences include relevant contexts. As we will show later in the experiment section 6, simple bag-of-words based features that do not make use of any rich annotations yield poor performance. We manually evaluated the patterns

which were extracted from a subset of the distantly supervised data and matched more than one instance yielding an accuracy of between 10%-60%. Patterns learned for slot types such as "*title*" and "*spouse*" have higher accuracy than those for "*employment*" and "*residence*" because the latter involves more diverse and implicit expressions for both relation and temporal information.

In the following two subsections, we will discuss further challenges and our proposed solutions to enhance distant supervision.

4.3. Enhancing the Quality of Distant Supervision: Addressing Naïve Assumptions

In this section we list some naïve assumptions made by current distant supervision methods, expanding those listed in [60]. Recall that we assume our classification instances are of the form <SLOT_TYPE(*query*, *slot fill*), T >, where T is a normalized temporal expression.

(1) The Distant Supervision Assumption (DSA): The traditional *distant supervision assumption* is as follows: given a relation in a structured knowledge base of the form $R(x_1, \ldots, x_k)$, assume that any context that mentions $x_1, \ldots, x_k$ expresses the relation R. In most prior work $k = 2$. In principle, $k = 3$ in our case since we intend to classify the relationship between a query, slot fill, and temporal expression. But our pipeline framework assumes that by the temporal classification step our fact, of the form SLOT_TYPE(*query*, *slot fill*), has been correctly identified, so in this sense we adopt the distant supervision assumption for the fact. However, as any given temporal expression is related to a multitude of events and relations, coincidental co-occurrence is very common. The DSA is simply untenable when dealing with temporal expressions, especially those of vague granularity, and when the relation in question endured for a long period of time. A possible solution is to assume that all entities are competing for the temporal expression. Then we can compare their confidence values to filter out some false positives. For example, adopting the DSA, the following sentence was mistakenly identified as a context for <EMPLOYEE(*"Chris Kronner"*, *"Slow Club"*), *"December"*> instead of the correct tuple <EMPLOYEE(*"Chris Kronner"*, *"Serpentine"*), *"December"*>:

"**Slow Club**'s **Chris Kronner** faced similar challenges taking on his second executive chef position at Serpentine, which opened in **December**."

On the other hand, this will not work for all cases, for example, in *"US President Obama joined Mothers Against Drunk Driving (MADD) last week"*, the temporal expression *"last week"* can be linked to an EMPLOYEE, RESIDENT, and a MEMBER relation. We address this issue by including the *"NONE"* label which is analogous to how [60] handle query-slotfill relations. Furthermore, our relabeling scheme (Section 4.4) helps correct erroneous *"HOLDS"* labels to *"NONE"* for a variety of reasons. [5]

(2) One sense per string: Many elements of a KB are polysemous - their meaning changes depending on context. For example, one KB entry indicates that "*Raul Castro*" is a "*general*". Adopting the "*one sense per string*" assumption would lead to a false positive given the following example:

[5] In fact, *"NONE"* is ambiguous between (1) the query and slot fill are in relation SLOT_TYPE, but in this context it is not explicitly related to time T, and (2) the query and slot fill are not in relation SLOT_TYPE, or any other relation, and (3) the query and slot fill are in relation SLOT_TYPE*, which is not explicitly related to T, and (4) the query and slot fill are in relation SLOT_TYPE*, which is explicitly related to T, but we still label <SLOT_TYPE(*query*, *slot fill*), T > as *"NONE"*.

"Monday, **Raul Castro** set the date for local (city and town) **general** elections as **October 21** with a second round **October 28**."

We did not specifically address this concern in the current implementation, although our semi-supervised self-training procedure could, in theory, eliminate certain erroneous instance types.

(3) One sentence per relation: Traditional distant supervision methods [50] are usually applied at the sentence level, under the assumption that an instance of a relation is specified only if all elements involved are present in the same sentence (after applying entity coreference resolution). This assumption is invalid when a document is typically talking about a *centroid entity*, such as the employment history of a person or an organization. In such a document, a distant supervision approach should locate the centroid query or slot fill and search for context sentences that include the other elements of a KB entry for that query. We address one particular instance of this concern: if a sentence contains the query and slot fill, but no temporal expression, we classify $<$ SLOT_TYPE$(\textit{query}, \textit{slot fill}), DCT >$, where DCT is the normalized document creation time.

The basic heuristic of distant supervision for binary relations described in this paper, extended to include temporal expressions, is to label according to the temporal information provided in Freebase. In most cases the distant supervision hypothesis is correct, but various errors can be introduced by preprocessing steps, especially when the training data is collected from the Web. Some common causes of these errors are:

- Coreference errors that yield incorrect name matching.
- Temporal expressions that are normalized incorrectly.
- Temporal information with different granularities has to be compared. For example, the KB states that "*John* married *Mary* in *1997*", but not the exact day and month. Should we label a classification instance containing a more specific temporal expression such as "*September 3, 1997*" as a "*START*"?
- Information offered by the KB may be incorrect, or contradictory to information in Web documents.

In order to enhance the quality of distant supervision, we propose to exploit multi-layer rich annotations in distant supervision. We apply the Stanford NLP Core toolkit [29] to generate name tags, co-reference chains, dependency parses and to normalize temporal expressions, and each temporal expression was represented as in interval at the day granularity.

In most news and web blog documents, we can use the document creation time as the reference date to normalize time expressions. However, some events may appear in a list of parallel items in which each item has its own reference date. In the following example, we should use "*Aug. 6, 2007*" as the reference date for the slot fill "*Tom LaSorda*" as a "*top_employee*" of "*Daimler Chrysler AG*":

"**Aug. 3, 2007**: **Daimler Chrysler AG** finalizes the sale of Chrysler to Cerberus.
Aug. 6, 2007: Bob Nardelli appointed Chrysler chairman and CEO. **Tom LaSorda** becomes vice chairman and president."

We segmented such documents according to temporal blocks so that each block has its own local reference date.

Finally, we treat each normalized temporal expression as a range of two dates. When the normalized date corresponds to an exact day, month and year, this two dates are equal. Whenever the date is more vague, expressing a month, a year or a decade, we use the two dates to record the corresponding range of time. This allows us to work with different temporal granularities and facilitates the aggregation of temporal information later in our system.

The annotated output is used when searching for sentences that mention both the

query entity and the slot value. Finding these sentences by string matching provides only very limited coverage, so we use named entity recognition and coreference results to expand the set of relevant sentences. We search for the coreference chains that contain the provided slot value or entity name and we select sentences that mention both, according to the coreference chains.

4.4. Enhancing Scalability of Distant Supervision

Our TSF system attempts to capture relationships that can be expressed in many ways. Both query entities and slot fills can be denoted canonically by name, or instead by a part of their name, pronoun, title or another nominal expression. In addition, there are many temporal expressions that denote regions in time that are constituents of the entire region of time during which a particular TSF relation endures, which may even co-occur with query and slot fill by coincidence. Thus, over 100,000 features were required to generalize these complicated contexts for each slot type, which made it unfeasible to learn supervised models. In addition, only a few features were relevant to each instance, making the data quite sparse.

In order to improve the quality of extracted training instances and incorporate local contexts not captured by distant supervision, we applied *self-training*, a semi-supervised learning method that has been used to label data for tasks such as parsing [48]. Using a small set of human annotations, or *seed* examples, we iteratively labeled the partitioned unlabeled set, retaining only the most confident labels for retraining the classifier in each round. However, the size of the training data set resulted in a prohibitively large, sparse feature space. We perform two steps in order to generate a more parsimonious classification model that can be used for self-training: (1) *feature elimination* was used to identify a minimal set of model features; followed by (2) *relabeling* using the reduced feature set and a lasso regression (least absolute shrinkage and selection operator) [74, 75] classifier.

Recent work has demonstrated that regularized logistic regression provides outstanding predictive performance across a range of text classification tasks and corpora using sparse data sets [4]. For sparse data sets, L1 (lasso) and L2 (ridge) regularized regression can be used to constrain the coefficients in a classification model for the purpose of identifying a subset of features that are strong predictors for the given label [52]. Lasso regularized regression has been successfully applied to bioinformatics and pharmacology [41].

Feature Elimination: Intuitively, variable selection might be performed to address the curse-of-dimensionality in order to reduce the storing cost or to help process the predictive variables. For each of the M features in the set $F = \{f_1, ...f_M\}$ extracted from the training data, we evaluate the feature's independence given each class label, inserting only those features that meet a threshold p-value into the minimal feature set F'.

Self-training: To re-label the instances using the reduced feature set F', we annotated a small training set by hand and used lasso regression, which has the benefit of shrinking the coefficients of features towards zero so that only the subset of features with the strongest effects are incorporated into the classifier [52]. The shrinkage parameter, a constant $s > 0$, is tuned using cross-validation. For a collection of N training instances, $D = \{(x_1, y_1), ..., (x_N, y_N)\}$, of dimension d, the lasso coefficients $\hat{\beta}$ are calculated as

follows:

$$\hat{\beta}^{lasso} = \arg\min_{\beta} \left\{ \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{d} \beta_j x_{ij})^2 \right\}$$

$$\text{subject to: } \Sigma_{j=1}^{d} |\beta_j| \leq s$$

Lasso regression limits the expression of extraneous information and as a result provides additional feature selection properties. The lasso minimizes the residual sum of squares with the constraint that the absolute value of the regression coefficients must be less than s. When s is large enough, there is no effect on the solution, but when it shrinks it has the effect of reducing some model coefficients to zero, or almost zero. We used cross-validation to determine the best values for s in our experiments.

In our experiments, we used .005%-.101% of training instances from distant supervision data as the initial labeling seeds for self-training. We used the agreement between classification results for two different values of s. As the new data portion is labeled, those retained for retraining are instances for which there is an agreement reached by multiple classifiers. Similar ideas for re-labeling instances have been applied to improve distant supervision for relation extraction. For example, [66] developed a multi-instance multi-labeling learning framework to improve classification, and [67] developed a generative model to predict whether each pattern expresses each relation via hidden variables.

5. Classification Challenges

After obtaining a large training data set, the remaining key issue is to train effective temporal classifiers.

5.1. How Much to Compress?

For many NLP tasks including this new TSF task, one main challenge lies in capturing long contexts between related elements. Without direct access a logical form representation, a TSF system must determine which words in a sentence are predicates of eventualities that could have any bearing on the fact types of interest, which words correspond to potential participants, whether each participant stands in some relevant thematic relation to any eventuality, and whether any temporal expressions stand in some relation to any predicates of eventualities (and if so, which sort of relation). More words in a sentence means there are *a priori* more candidates for each component of sentence meaning mentioned above; also, when predicates of eventualities and their arguments are separated by many intervening words the relationships in question may be best explained in terms of a structured representation. Semantic analysis such as dependency parsing can *compress* long contexts by extracting their syntactic structure, and thus reduce ambiguities. For example, there is a long context between the query "*Mugabe*", the time expression "*1980*" and its slot fill "*ZANU-PF*" in the following sentence "*ZANU, which was renamed **ZANU-PF** after taking over ZAPU, has been the country's ruling party and led by **Mugabe** since **1980**.*" Some context words such as "*fired*" can cause ambiguity ("*end position*" or "*attack*"). Thus intuitively structured approaches might be exploited to remove irrelevant information. For example, "*1980*" can be identified as the starting date based on the short dependency paths between "*ZANU-PF*", "*Mugabe*" and "*1980*".

However, current core NLP annotation tools such as dependency parsing and coreference resolution do not yet perform well enough to produce ideal results for real applications. The deeper the representation is, the greater the risk of introducing annotation errors. Furthermore, for certain types of slots such as "*title*", since the contexts are relatively short between the query entity and its slot fill (e.g. "*Today[Time] President[Title] Obama [Query]...*"), structured representation is not necessary.

5.2. Combination of Flat Approach and Structured Approach

We developed a hybrid system combining benefits of a flat approach designed to capture surface features and a structured approach designed to incorporate a more complex syntactic representation as follows.

5.2.1. Flat Approach

The flat approach uses two types of features: window features and dependency features. The window feature is a set of all tokens that occur in the normalized sentence within 4 tokens in either direction of a mention of any *target*, i.e. that of a query entity, a slot value or a target temporal expression. Two dependency feature values for each target are extracted from each example using the Stanford dependency parser [47], resulting in two sets of tokens for each target T. One set contains all tokens that any instance of T governs (see next subsection), the other set contains all tokens governed by any instance of T.

For two feature values U, V, let K_T be the normalized size of their intersection

$$K_T(U,V) = \frac{|U \cap V|}{\sqrt{|U|^2 + |V|^2}} \tag{1}$$

Let F denote the flat features. Then for any $G \subseteq F$, let K_S be the kernel function for a pair of examples, and $x.i$ the feature value for the i^{th} feature value type for example x:

$$K_S(x,y) = \sum_{i \in G} K_T(x.i, y.i) \tag{2}$$

We trained a classifier using Support Vector Machines [20] setting $G = K$.

5.2.2. Structured Approach

Binary dependency relations can reveal information about an entity's attributes, but relations between arguments of the same event may be implicit. In the structured approach, we exploit collapsed dependency parsed graphs to capture relevant grammatical relations and discover syntactic patterns. A dependency graph is a set of syntactic relations called dependencies. The criteria for the *governor* and *dependent* places for each of the Stanford Dependency Parser's 53 dependency types are defined in terms of syntactic tree structure. For example, *nsubj* is defined as [23]:

A nominal subject is a noun phrase which is the syntactic subject of a clause. The governor of this relation might not always be a verb: when the verb is a copular verb, the root of the clause is the complement of the copular verb, which can be an adjective or a noun.

Part of the dependency parse graph for the example in section 3.3 is provided in Figure 2. Figure 2 reveals that *Davis* and *1975* are related, which would have been difficult to infer from the surface structure given the distance between the two terms.

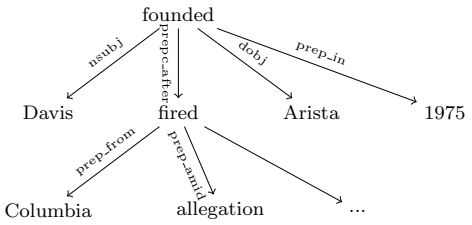


Fig. 2. Dependency Parse Graph of the Sample Sentence

Davis $\xleftarrow{\text{nsubj}}$ founded/found/VBD/Start-Position $\xrightarrow{\text{prep\_in}}$ 1975

Arista $\xleftarrow{\text{dobj}}$ founded/found/VBD/Start-Position $\xrightarrow{\text{prep\_in}}$ 1975

[] Davis $\xleftarrow{\text{nsubj}}$ founded/found/VBD/Start-Position $\xrightarrow{\text{dobj}}$ Arista

Fig. 3. Three shortest paths from Figure 2

We extend the idea of shortest path on a dependency graph to include three items: query entity, slot fill and time expression. Each instance is represented by three paths: (i) the path between query entity and temporal expression (P_1), (ii) the path between slot fill and temporal expression (P_2); and (iii) the path between query entity and slot fill (P_3).

Each shortest path P_i is represented as a vector $< t_1, t_2, ..., t_n >$, where t_i can be either a vertex or a typed edge in the dependency graph. Each edge is represented by one attribute, which is formed by combining the corresponding dependency type and arrow from the governor to the dependent. More formally, attribute $a \in \mathcal{D} \times \{\leftarrow, \rightarrow\}$, where $\mathcal{D}$ is the set of dependency types, and the arrow is directed from the governor to the dependent word. Vertices, on the other hand, may contain different levels of features, including word, part-of-speech tag, lemma, entity type, semantic classes (only for verbs that were considered event *trigger words* in the Automatic Content Extraction (ACE) 2005 corpus [6]). For example, in the sentence represented in Figure 2, there exists *prep_in* dependency from *founded* to *1975*. *Prep_in* represents prepositional relation between these two words, meaning that the action *founded* happened at time *1975*.

When we search for the shortest path between two nodes, we consider all mentions of the query entity and the slot fill in a sentence. For this reason there could be more than one candidate for each P_i. If some candidate paths contain predefined trigger words, we choose the shortest such path. Otherwise, we choose the shortest path among all candidates. Figure 3 shows three shortest paths that result from the sentence in Figure 2. These paths contain both lexical features and syntactic relations.

Following [14] and [44], we present a string kernel function based on dependency paths. The main idea is to use the kernel trick to deal with common substring similarity between dependency paths, and to extract syntax-rich patterns from dependency paths. Let x, y be two instances. We use $l(P)$ to denote the length of a dependency path P, $P[k]$ to denote the set of all substrings of P which have length k, and $a \in P[k]$ is a substring of P with length k. The kernel function of x and y is defined as follows:

[6] http://projects.ldc.upenn.edu/ace/

$$K_s(x,y) = \sum_{i=1}^{3} K_p(x.P_i, y.P_i) \tag{3}$$

$$K_p(P_x,P_y) = \sum_{k=1}^{Min(l(P_x),l(P_y))} \sum_{a\in P_x[k], b\in P_y[k]} \prod_{i=1}^{k} c(a_i,b_i) \tag{4}$$

Where K_p is a kernel function on two dependency paths P_x and P_y which sums the number of common substrings of feature paths in P_x and P_y with length from 1 to the maximum length. In $c(a_i, b_i)$ we calculate the inner product of the attribute vectors of a_i and b_i, where a_i and b_i are elements of two paths respectively. The final kernel function K_s does the summation of the partial results of the three dependency paths (query entity-slot fill, query entity-temporal expression, slot fill-temporal expression).

A problem of Equation (4) is that K_p has a bias toward longer dependency paths. To avoid this bias, we normalize K_p as in [44]. This normalization scales the feature vector $\phi(P)$ in the kernel space to $\phi'(P) = \frac{\phi(P)}{|\phi(P)|}$:

$$K'_p(P_x,P_y) = \frac{K_p(P_x,P_y)}{\sqrt{K_p(P_x,P_x)\cdot K_p(P_y,P_y)}} \tag{5}$$

A deviation from [44] and [14] is that we count common substrings from m to maximum, rather than a fixed length. Furthermore, we only consider contiguous substrings in K_p because each substring feature in the kernel space is treated as a pattern. Non-contiguous substrings with the same length can be safely discarded as different patterns. Given the representation and kernel function, we trained classifiers based Support Vector Machines (SVMs). To combine two classifiers, we consider the output from the structured classifier as the default output. If the output equals $< -\infty, \infty, -\infty, \infty >$, we combine it with the output from the flat classifier using temporal aggregation.

6. Experiments

In this section we present the overall performance of our system, and the impact of each approach to one of the annotation or classification problems outlined above.

6.1. Data Set

The source collection includes 1,286,609 newswire documents, 490,596 web documents and hundreds of transcribed spoken documents. The KB is from the October 2008 dump of English Wikipedia and includes 818,741 nodes.

The evaluation data includes 100 queries, with 80 person entities and 20 organization entities. As is the case with Information Retrieval (IR) evaluations, it is not feasible to prepare a comprehensive temporal slot filling answer key in advance. Because of the difficulty of finding temporal information in such a large corpus, any manually-prepared key is likely to be quite incomplete. Instead (as for IR) KBP2011 organizers pooled the responses from all the systems and human annotators, and have human assessors judging the responses. The output for the full temporal task is scored through system output pooling. The diagnostic temporal task is based on a set of slot fills tagged through

[ht!]

Table 4. Evaluation Data Set

Slot Type	Task	# of Tuples
per:countries_of_residence	diagnostic	79
	full	**287**
per:statesorprovinces_of_residence	diagnostic	41
	full	**44**
per:cities_of_residence	diagnostic	47
	full	**109**
per:member_of	diagnostic	61
	full	**86**
per:employee_of	diagnostic	6
	full	**20**
per:title	diagnostic	19
	full	**89**
per:spouse	diagnostic	24
	full	**52**
org:top_members/employees	diagnostic	15
	full	**24**
total	diagnostic	292
	full	**711**

manual annotation, and is scored automatically. Table 4 presents the number of tuples in the diagnostic task and full task respectively.

6.2. Scoring Metric

KBP2011 defined a metric $Q(S)$ that compares a system's output $S =< t_1, t_2, t_3, t_4 >$ against a gold standard tuple $S_g =< g_1, g_2, g_3, g_4 >$, based on the absolute distances between t_i and g_i:

$$Q(S) = \frac{1}{4} \sum_i \frac{1}{1 + |t_i - g_i|}$$

The absence of a constraint on t_1 or t_3 is treated as a value of -∞ and the absence of a constraint on t_2 or t_4 is treated as a value of +∞.

Assume the set of gold standard tuples is $\{G^1, G^2, ..., G^N\}$, and the set of system output tuples is $\{S^1, S^2, ..., S^M\}$, where each G^i is a four tuple for the i-th gold standard slot fill $< g_1, g_2, g_3, g_4 >$, each S^j is a four tuple for the j-th slot fill in system output $< t_1, t_2, t_3, t_4 >$. Each element is associated with an instance of a unique slot fill and scored independently. KBP2011 defined the following Precision, Recall and F-measure scores:

$$Precision = \frac{\sum_{S^i \in C(S)} Q(S^i)}{M}$$

$$Recall = \frac{\sum_{S^i \in C(S)} Q(S^i)}{N}$$

$$F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

Where $C(S)$ is set of all instances in the system output which have correct slot filling answers, and $Q(S)$ is quality value of S.

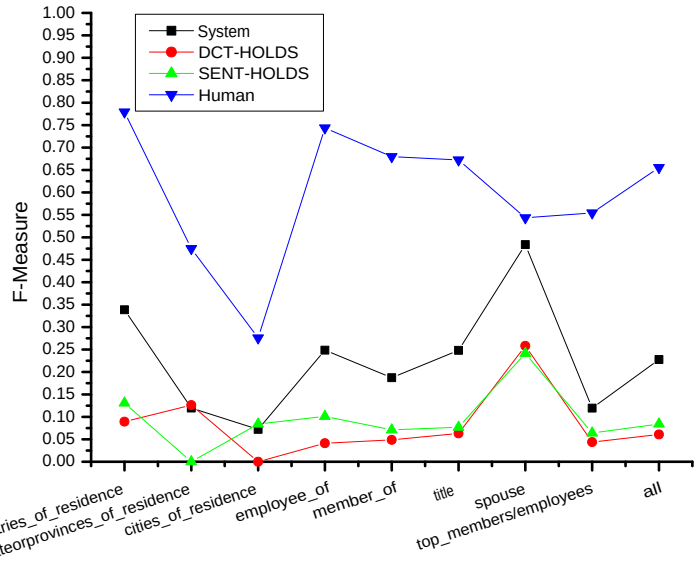


Fig. 4. F-Measure of Systems and Human Annotators in Full Task

6.3. Overall Performance

Figure 4 and Figure 5 present the overall performance of our system in the full task and the diagnostic task. Two baselines are included for comparison: DCT-HOLDS, in which each query is labeled 'HOLDS' for the associated Document Creation Time (DCT); and SENT-HOLDS, in which each classification instance is labeled HOLDS, but if no temporal expression appears in the context of the query entity and slot value then DCT-HOLDS is used. The performance of human annotators is included as well.

The system achieved the top performance in both the full and diagnostic tasks in the KBP2011 evaluation, exceeding the DCT-HOLDS baseline by 16.7% F-measure, the SENT-HOLDS baseline by 14.4% F-measure, and the mean score of all system submissions by 11.82% F-measure.

Our regular slot filling system had a 32% F-measure for the full temporal task. The disparity in performance between the full and diagnostic tasks is largely due to slot filling errors, which are not recoverable. The human annotators only achieved 65.5% F-measure (70.4% Precision and 61.3% Recall), which reflects the fact that accurate coverage of all temporal information is very difficult. For the "*spouse*" slot, our system closely approaches human performance.

Due to the limited time allowed for annotation, newswire documents with a much higher than normal concentration of explicit time information were selected for the diagnostic task. The system barely exceeded the baselines in the diagnostic task in part due to high concentration (more than 58.4%) of sentences whose correct label was *"HOLDS"*. The "*per:cities_of_residence*" was easier than other slot types, mainly because one evaluation document included a lot of short lists containing only the relevant facts, each of which should be labeled "*HOLDS*":

"EUROPE DMITRI MEDVEDEV Prime minister of Russia 42 Moscow, Russia
...
LIONEL MESSI Soccer player 20 Barcelona, Spain
...

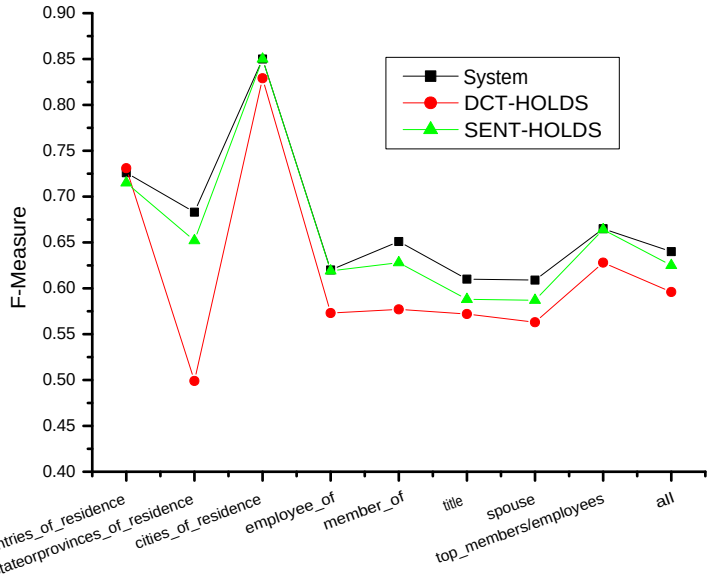


Fig. 5. F-Measure of Temporal Slot Filling Systems in Diagnostic Task

[h]

Table 5. Feature Reduction Using Multi-class Logistic Regression

Features	residence	title	spouse	employment
Initial set	10757	31974	40979	51399
Final set	451	2024	1247	2151
Reduction (%)	95.81	93.67	96.96	95.82

ASIA PERVEZ MUSHARRAF President of Pakistan 64 Islamabad,Pakistan
...
ZHOU XIAOCHUAN Governor of China's central bank 59 Beijing, China
...".

6.4. Distant Supervision Scalability Results

Table 5 shows the initial number of features in the baseline system, and after feature reduction. The most notable benefit from our annotation enhancement is the ability to build more parsimonious classification models with performance comparable to what was achieved with the full feature set. Not only does the new representation increase efficiency, but it facilitates the interpretation of the model by providing information about the importance of features.

Figure 6 presents the performance of our system on the full TSF task, before and after applying feature reduction and re-labeling techniques. We can see that our methods dramatically enhanced the speed (almost 100%) of training while slightly improving the overall performance (F-measure from 22.56% to 22.77%). Experimental results have also shown that the F-measure gain on each slot type correlates (.978) with the number of seed instances used in self-training based re-labeling. The most dramatic

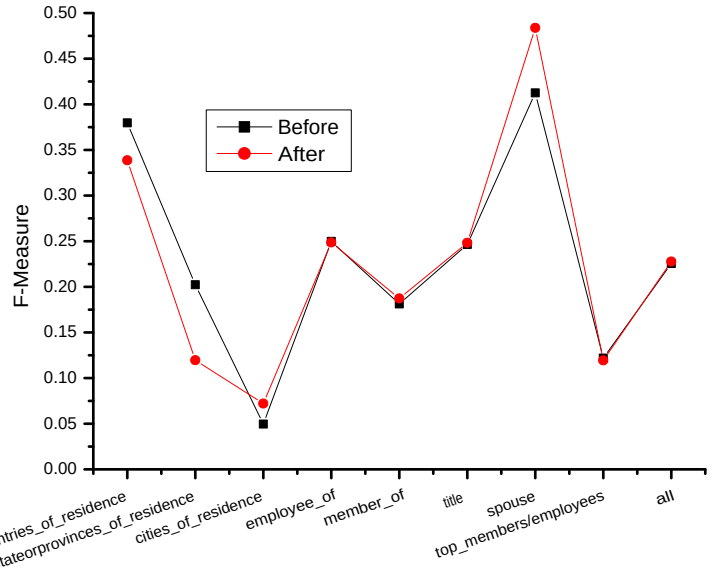


Fig. 6. Impact of Feature Reduction and Instance Re-Labeling on Full Task

improvements are obtained for the "*per:spouse*" slot (7.12% absolute F-Measure gain). Although the system performance without enhancements for better data quality is still notable, we believe better quality training can provide a developer with even higher precision. For example, given an entry in Freebase "*Query = Constanze Mozart; Slot Type = per:spouse; Slot Value = Wolfgang Amadeus Mozart; Start-Date = 1790*", the baseline distant supervision approach mistakenly labeled the time expression "*1790*" in the following sentence as "*START*" based on simple string matching and entity coreference resolution. Since the relation and time must be explicitly entailed and not inferred based on reasoning, the sentence lacks the context features that are associated with the "*START*" label for a marriage relation. Since the less important features have been eliminated all together, or have little or no impact on label assignment due to shrinkage, our re-labeling method successfully corrected the label to "*NONE*".

"*During this time **Mozart** made long journeys hoping to improve **his** fortunes: a visit in spring of 1789 to Frankfurt, and a **1790** visit to Frankfurt.*"

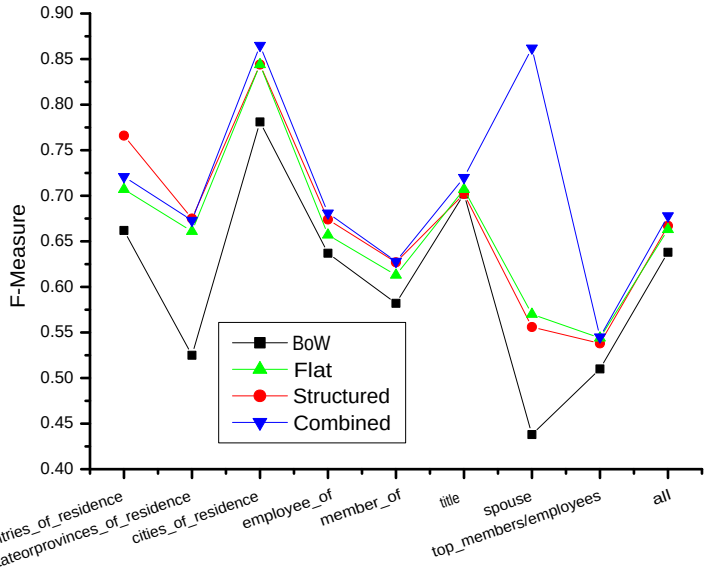
6.5. Comparison of Classification Approaches

To assess the relative contribution of the flat and structured features, as well as the extent to which they are complementary, the diagnostic task was performed on the KBP2011 manually labeled training data as shown in Table 3. This data set contains 430 query entities, 748 slot fills and the corresponding temporal 4-tuples.

We used the LIBSVM library [18][7] to train SVM classifiers. Figure 7 presents the performance of the proposed combination approach against *Structured*, *Flat*, and *BoW*. The baseline *BoW* uses only bag-of-words based features, in SVMs. Compared to other

[7] http://www.csie.ntu.edu.tw/∼cjlin/libsvm/

Fig. 7. Comparison of Temporal Classification Approaches on Diagnostic Task



approaches, *BoW* achieves the lowest performance. The combined system outperforms both of the structured approach and the flat approach alone, and achieves the highest scores in 7 slot types. The Wilcoxon Matched-Pairs Signed-Ranks Test was conducted on a four-tuple basis, showing that the combined system is an improvement over both the flat (99.9% confidence level) and structured approach (99.8% confidence level) alone.

The results indicate the flat approach and the structured approach are complementary.

The structured approach is particularly effective for capturing long contexts when the query has multiple slot fills in one sentence. For example, in the sentence "***Trong*** *became secretary of the Hanoi Party Committee in January 2000, chairman of the Central Theoretical Council in 2001, member of the CPVCC in April 2001, and member of the* ***Political Bureau*** *in* ***April 2006***", the structured approach can compress the long contexts among the query "*Nguyen Phu Trong*", slot fill "*Political Bureau*" and time expression "*April 2006" into "[Query] member of [Slot Fill] in [Time]*" while the flat approach failed due to confusions stemming from other entities in the context.

On the other hand, dependency parsing can produce errors. For example, it failed to capture the dependency relation between "*September 2005*" and the pair ("*Avi Dichter*", "*the Brookings Institute*") in the following sentence "*In* ***September 2005***, ***Dichter*** *left office and became a research fellow at* ***the Brookings Institute*** *in Washington , D.C.*". In contrast the flat approach can easily identify "*September 2005*" as the starting date for the query "*Avi Dichter*" to be a member of "*the Brookings Institute*" based on lexical features such as "*became*".

Similar to the results of instance re-labeling, the most gain of this combined classification approach was achieved for the "*SPOUSE*" slot because of the associated contexts are very diverse, including both simple surface patterns (e.g. "*John married with Mary in 1990.*"), as well as long and complicated contexts (e.g. "*After John 's death in 2003, Mary fought with his children over the distribution of the Senator's estate, etc.*"

implicitly indicates that the "*spouse*" relation between "*Mary*" and "*John*" ended in 2003.)

In addition, the bag-of-words baseline had the lowest score for the "*spouse*" slot because of noise in the training data. Compared to other slot types such as "*employment*" and "*title*", the "*spouse*" entries in Freebase include many more errors, perhaps because it is more difficult to track the detailed start and end dates of people's marriage because such information is less credible and appears less frequently in public data. For example, Freebase indicates the "*spouse*" relation between "*Jon Voight*" and "*Marcheline Bertrand*" is valid in the range of 1971-1978. Therefore the time expression "*1976*" in the following training sentence was mistakenly labeled as "*HOLDS*" instead of the correct label "*ENDS*": "*According to former babysitter late mother Marcheline Bertrand virtually abandoned her baby daughter after a painful 1976 split from husband Jon Voight.*".

7. Remaining Challenges - The Task at Hand

Temporal slot filling is a brand new and ambitious task, and we have observed reasonable success both in obtaining training data automatically and developing a high-performing system. However, some challenges still remain. In the following we summarize the significant issues and suggest some research directions.

7.1. Capturing Implicit and Wide Contexts

In some examples, there is no eventuality that explicitly connects the important elements, and no common surface pattern, thus the flat and structured approaches will fail:

- *Profit*: "**Daimler Chrysler** reports 2004 profits of $3.3 billion; **Chrysler earns** $1.9 billion" (indicates "Schrempp Chrysler" is an employee of "Daimler Chrysler" which HOLDS at 2004)
- *Speech*: ""**Daimler Chrysler** is not yet where we want it to be, but we are headed precisely in the right direction, " **Schrempp** says." indicates "Schrempp" is an employee of "Daimler Chrysler AG" WITHIN the reference date.

In other examples, the query entity himself is quoted. The system has no way of distinguishing such cases, so patterns specific to it could not be learned.

- *Speech*: **Daimler Chrysler** is not yet where we want it to be, but we are headed precisely in the right direction **Schrempp** says. (indicates "*Schrempp*" is an employee of "*Daimler Chrysler AG*" during the reference date)

Some other cases require a system to capture deep lexical semantic knowledge from multiple phrases. For example, "**Query** will continue as a member of **Slot Fill**" indicates the membership relation is "*HOLDS*" for the reference time although it includes a future tense; "**Query** filed for divorce from **Slot Fill**" indicates the spouse relation is "*HOLDS*" the reference time although it includes "divorce" which is normally a keyword for "*END*".

7.2. Coreference Resolution Errors

As in other IE tasks, coreference resolution is another bottleneck in this TSF task. These errors can be categorized into the following three types:

(1) Name Coreference Errors:

These errors normally appear between entity mentions with different entity types. For example, a slot fill "*Republican Party*" is stored as "*R*" in the database used for distant supervision. A Coreference system that allows inexact matching mistakenly linked it to other names that include the letter "*R*". However, a system that requires exact matching would have lower recall.

(2) Nominal Coreference Errors:

Nominal coreference resolution remains very challenging, especially when multiple candidate antecedents appear prior to the nominal mention in various sentences. For example, coreference resolution systems failed to identify the link between the slot fill "*Giuliani Partners*" and "*the firm*" in the following sentences:

"Almost overnight, he became fabulously rich, with a $3-million book deal, a $100,000 speech making fee, and a lucrative multifaceted consulting business, **Giuliani Partners**. As a celebrity rainmaker and lawyer, his income last year exceeded $17 million. His consulting partners included seven of those who were with him on 9/11, and in 2002 Alan Placa, his boyhood pal, went to work at **the firm**.".

(3) Pronoun Coreference Errors:

Most of the coreference errors propagated to TSF are pronoun resolution errors, especially for female pronouns. Some newswire documents include centroid entities to which most pronouns should be linked. For example, in an article entitled, "*3rd Ld-Writethrou: Nguyen Phu Trong re-elected Vietnamese top legislator*", the query entity "*Nguyen Phu Trong*" is the centroid entity - this individual is salient enough throughout the discourse that in many paragraphs he is only referred to using "*he*".

"**He** pursued a masters degree in political economics at the Nguyen Ai Quoc High- ranking Party Institute from September 1973 to April 1976. **He** worked as an editor at the magazine's Party Construction Department between May 1976 and August 1981, ... ".

In the above paragraph, a great deal of temporal information can be easily extracted if the coreference resolution component successfully links these pronoun mentions to the query.

7.3. "Long Tail" Problem

The final challenge lies in the long-tailed distribution of temporal context patterns - in spite of a few patterns that match many instances, it is still the case that a high percentage of all patterns only match one or a few instances. For example, we extracted indicative contexts from distant supervision and summarized the number of instances that match each START pattern in Figure 8. Dependency parsing can filter out some irrelevant contexts but deeper understanding may be required to generalize over the diversity of lexical expressions that can be used to indicate the same relation. For example, the starting date of an employment relation can be expressed by many long-tail patterns such as "*would join*", "*would be appointed*", "*will start at*", "*went to work*", "*was transferred to*", "*was recruited by*", "*took over as*", "*succeeded PERSON*", "*began to teach piano*", etc.

7.4. Toward Temporal Reasoning

The goal of temporal slot filling is to identify facts and ground them in time. The goal of the task at hand was to extract ranges of temporal values for the beginning and ending of KBP2011 TSF relations that were as constrained as could be inferred with 100% certainty, as verified by human annotators. The current approach is conservative in that a good system won't over-constrain the bounds. Thus, as more documents are retrieved

Fig. 8. START Pattern Distribution

for a given query the simple four-tuple aggregation process described above would not have to consider retrenchment to larger ranges on interval endpoints. This approach parallels that taken to yield the gold standard annotation. A less conservative but more realistic approach might attempt to incorporate world knowledge to capture what is expected, rather than merely what is asserted with certainty, about an eventuality's extensions in time.

Some previous work in the TempEval [82, 71, 43] and ACE temporal information extraction [30] conducted temporal reasoning, but all of them focused on single-document extraction. When the temporal facts become inter-dependent across multiple entities or relations, we can also learn global constraints and incorporate these constraints into an Integer Linear Programming Framework [42, 49, 25, 69]. For example, [49] learned statistical models to classify relations between query entity, slot fill, and temporal expression as start, end, or other, and to maximize the coherence of all resulting timelines among all entities and slot types, and applied these components jointly to perform a modified slot filling task. Their modified task is evaluated with respect to Freebase as opposed to manual annotation extracted only from a document, and so gold standard annotation was given in terms of exact start and end dates only (though +/- infinity values were allowed). In our system we applied shallow reasoning, such as the propagation of temporal information from *cities_of_residence* to *stateorprovinces_of_residence*. However, deeper reasoning techniques are required for the remaining difficult cases. We can roughly categorize them into the types listed below.

Temporal four-tuples that more closely reflect the time intervals for their associated facts could be obtained using various forms of reasoning. *Cross-slot reasoning* consists of propagating logical constraints between relations and their associated time intervals. A simple example of cross-slot reasoning for a fixed entity involves birth and death: a given entity cannot be in an employment, spouse, or residence relation in a time period before being born nor after death; therefore, given the time interval associated with

an entity's life ($L =< l_{init}, l_{end} >$), assuming 100% confidence, a four-tuple $T =< t_1, t_2, t_3, t_4 >$ for the above relations is constrained as follows $t_1 >= l_{init} \wedge t_4 <= l_{end}$.

A slightly more complex case is described below:

Sheetrit is a Knesset (parliament) member...
Sheetrit was born in Morocco and immigrated to Israel in 1957

Consider the relations (1) $resident\_of$(Sheetrit,Israel), (2) $resident\_of$(Sheetrit,Morocco), and (3) $member\_of$(Sheetrit,Knesset), and their corresponding four tuples T^i, each initialized at $< -\infty, \infty, -\infty, \infty >$. After processing the first sentence we have $T^3 =< -\infty, 2008-09-17, 2008-09-17, \infty >$. If the system knows the rule: $member\_of(x$,Knesset$) \rightarrow resident_of(x$,Israel), then T_2^3 and T_3^3 can be propagated to T^1. Processing the second sentence yields, $T^2 =< -\infty, \infty, 1957-01-01, 1957-12-31 >$, at which point the rule that an entity can only be resident of one country at a time propagates T_3^2 and T_4^2 to T_1^1 and T_2^1 respectively. Then, the contrapositive of the first rule allows propagation from T_3^2 to T_1^3. This process yields,

- $T^1 =< 1957-01-01, 1957-12-31, 2008-09-17, \infty >$
- $T^2 =< -\infty, \infty, 1957-01-01, 1957-12-31 >$
- $T^3 =< 1957-01-01, 2008-09-17, 2008-09-17, \infty >$

The above reasoning procedures were performed over different relations for a fixed entity. Conversely, *cross-entity reasoning* can be performed for a unary relation (the consequence of satisfying one place in a binary relation) over entities. Certain binary relation, attribute combinations yield *noncontemporanous* unary relations that are such that only one entity can satisfy them during a particular temporal interval, for example $spouse\_of$(Person, x), where *Person* is a citizen of a country where polygamy is outlawed, or $title\_of$(President of the United States, x).

Let x and y be entities that satisfy the noncontemporanous relation R, T^x and T^y being their four-tuples. If $T_4^x < T_3^y$, then $R(x)$ necessarily ended before $R(y)$. Therefore, $R(y)$ necessarily started after $R(x)$ ended, which means $T_1^y > T_4^x$ can be enforced. In this way values of $-\infty$ could be converted to more informative values. However, this rule breaks down when it is possible that the same entity satisfies R during two distinct intervals of time. In addition, there are nearly noncontemporanous relations: it is highly unlikely that someone has two employers at the same time, and even less likely that someone is the resident of two countries simultaneously, but this is possible. That said, in a question answering setting in which multiple possible answers to questions may be generated, and each is associated with a confidence level or probability, soft constraints that take into account the possibility of error may be desirable. Indeed, evaluating TSF output under [49] could indicate to what extent reasoning can be used to yield a closer approximation of a fact's true temporal extension.

8. Conclusions and Future Work

In this paper, we reviewed a new task called temporal Knowledge Base Population, focusing on the challenges this new task brings to temporal information representation, annotation and classification. We developed a novel approach that uses self-training and regression for feature reduction and instance re-labeling. This approach dramatically reduced the feature space, while achieving consistent improvement in the quality of temporal classification. In addition, we have observed that features derived from a structured text representation can help compress noisy context and reduce ambiguity; while on the other hand, surface lexical features are more robust and effective in other

cases. To capture the long and complicated contexts in temporal classification, we developed two approaches: a flat approach that uses lexical context and shallow dependency features, and a structured approach that captures long syntactic contexts by using a dependency path kernel tailored for this task. Experiment results showed that these two classification approaches were complementary and yielded a state-of-the-art system. In the future, we will continue to assess the impact of feature reduction on overall performance based on careful seed selection. We will also focus on addressing the remaining challenges outlined above, especially in exploiting information redundancy for temporal reasoning.

Acknowledgments

This work was supported by the U.S. Army Research Laboratory under Cooperative Agreement No. W911NF- 09-2-0053 (NS-CTA), the U.S. NSF CAREER Award under Grant IIS-0953149, the U.S. NSF EAGER Award under Grant No. IIS-1144111, the U.S. DARPA FA8750-13-2-0041 - Deep Exploration and Filtering of Text (DEFT) Program and CUNY Junior Faculty Award. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

[1] D Ahn, S.F. Adafre and M. de Rijke. Extracting Temporal Information from Open Domain Text: A Comparative Exploration. *Digital Information Management*, 2005.
[2] J. F. Allen. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26:832–843, 1983.
[3] E. Amigo, J. Artiles, Q. Li, and H. Ji. An evaluation framework for aggregated temporal information extraction. In *Pric SIGIR2011 Workshop on Entity-Oriented Search*, 2011.
[4] S. Aseervatham, A. Antoniadis, E. Gaussier, M. Burlet, and Y. Denneulin. A sparse version of the ridge logistic regression for large-scale text categorization. *Pattern Recogn. Lett.*, 32:101–106, January 2011.
[5] E. Bach. The algebra of events. *Linguistics and Philosophy*, 9:5–16, 1986.
[6] C. Baral, G. Gelfond, M. Gelfond, and R. B. Scherl. Textual Inference by Combining Multiple Logic Programming Paradigms. In *Proc. AAAI 2005 Workshop on Inference for Textual Question Answering*, 2005.
[7] A. Bell. *News Stories and Narratives*, pages 236–251. Oxford University Press, 1999.
[8] S. Bethard and J. H. Martin. Cu-tmp: Temporal relation classification using syntactic and semantic features. In *In SemEval-2007: 4th International Workshop on Semantic Evaluations*, 2007.
[9] S. Bethard, J. H. Martin and S. Klingenstein. Finding Temporal Structure in Text: Machine Learning of Syntactic Temporal Relations. In *International Journal of Semantic Computing (IJSC)*, 1(4), 2007.
[10] S. Bethard and J. H. Martin. Learning Semantic Links from a Corpus of Parallel Temporal and Causal Relations. In *Proc. Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, 1(4), 2008.
[11] K. Bollacker, R. Cook, and P. Tufts. Freebase: A shared database of structured general human knowledge. In *Proc. National Conference on Artificial Intelligence*, 2008.
[12] B. Bouguraev and R. K. Ando. TimeBank-Driven TimeML Analysis. In *Proc. Annotating, Extracting and Reasoning about Time and Events*, 2005.
[13] P. Bramsen, P. Deshpande, Y. K. Lee and R. Barzilay. Inducing temporal graphs. In *Proc. Conference on Empirical Methods in Natural Language Processing*, 2006.
[14] R. C. Bunescu and R. J. Mooney. A shortest path dependency kernel for relation extraction. In *Proc. of the HLT and EMNLP*, pages 724–731, 2005.
[15] N. Chambers, S. Wang, and D. Jurafsky. Classifying temporal relations between events. In *In Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 173–176, 2007.
[16] N. Chambers and D. Jurafsky. Jointly Combining Implicit Constraints Improves Temporal Ordering. In *Proc. Empirical Methods in Natural Language Processing*, 2008.

[17]N. Chambers and D. Jurafsky. Unsupervised Learning of Narrative Schemas and their Participants. In *Proc. the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP 2009)*, pages 173–176, 2007.

[18]C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001.

[19]Z. Chen, S. Tamang, A. Lee, X. Li, W. Lin, M. Snover, J. Artiles, M. Passantino, and H. Ji. Cuny-blender tac-kbp2010 entity linking and slot filling system description. In *Proceedings of the 2010 Text Analysis Conference*, 2010.

[20]C. Cortes and V. Vapnik. Support-vector networks. In *Machine Learning*, pages 273–297, 1995.

[21]D. Davidson. The logical form of action sentences. In N. Rescher, editor, *The Logic of Decision and Action*. University of Pittsburg Press, Pittsburg, PA, 1967.

[22]J. Dölling. Aspectual Coercion and Eventuality Structure. In K. Rovering & V. Engerer, editors, *Verbal Semantics* , 2011.

[23]M.-C. de Marneffe and C. D. Manning. Stanford typed dependencies manual. Technical report, Department of Computer Science, Stanford University, 2006.

[24]P. Denis and P. Muller. Predicting globally-coherent temporal structures from texts via endpoint inference and graph decomposition. In *IJCAI*, pages 1788–1793, 2011.

[25]Q. Do, W. Lu and D. Roth. Joint Inference for Event Timeline Construction. In *Proc. Empirical Methods for Natural Language Processing (EMNLP2012)*, 2012.

[26]D. Dowty. The effects of aspectual class on the temporal structure of discourse: Semantics of pragmatics? *Linguistics and Philosophy*, 9:37–61, 1986.

[27]N. Elhadad, R. Barzilay, and K. McKeown. Inferring Strategies for Sentence Ordering in Multidocument Summarization. *JAIR*, 17:35–55, 2002.

[28]C. Fellbaum. *WordNet: An Electronical Lexical Database*. The MIT Press, Cambridge, MA, 1998.

[29]J. R. Finkel, T. Grenager, and C. D. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL*, 2005.

[30]P. Gupta and H. Ji. Predicting unknown time arguments based on cross-event propagation. In *Proc. ACL-IJCNLP2009*, 2009.

[31]E. Hinrichs. Temporal Anaphora in Discourses of English. *Linguistics and Philosophy*, 9:63–82, 1986.

[32]J. Hitzeman, M. Moens, and C. Grover. Algorithms for analysing the temporal structure of discourse. In *Proceedings of the seventh conference on European chapter of the Association for Computational Linguistics*, EACL '95, pages 253–260, 1995.

[33]H. Ji, R. Grishman, and H. T. Dang. An Overview of the TAC2011 Knowledge Base Population Track. In *Proc. Text Analytics Conference (TAC)*, 2011.

[34]H. Ji and R. Grishman. Refining Event Extraction Through Unsupervised Cross-document Inference. In *Proc. the Annual Meeting of the Association of Computational Linguistics*, 2008.

[35]H. Ji, R. Grishman, Z. Chen and P. Gupta. Cross-document Event Extraction and Tracking: Task, Evaluation, Techniques and Challenges. In *Proc. Recent Advances in Natural Language Processing*, 2009.

[36]H. Kamp. *A Theory of Truth and Semantic Representation*, pages 189–222. Blackwell Publishers Ltd, 1981.

[37]G. Katz. Anti neo-Davidsonianism. In *Events as Grammatical Objects*, pages 393–416. CSLI Publications, 2000.

[38]P. Kingsbury and M. Palmer. From TreeBank to PropBank. *Proc. the 3rd International Conference on Language Resources and Evaluation (LREC)*, 2002.

[39]M. Lapata and A. Lascarides. Learning sentence-internal temporal relations. In *In Journal of AI Research*, pages 85–117, 2006.

[40]A. Lascarides and N. Asher. Temporal interpretation, discourse relations, and common sense entailment. *Linguistics and Philosophy*, 16:437–493, 1993.

[41]F. Li, Y. Yang, and E. P. Xing. From Lasso regression to feature vector machine. In *NIPS2005*, 2005.

[42]Q. Li, S. Anzaroot, W. Lin, X. Li and H. Ji. Joint inference for cross-document information extraction. In *Proc. 20th ACM Conference on Information and Knowledge Management (CIKM2011)*, 2011.

[43]X. Ling and D. Weld. Temporal information extraction. In *Proceedings of the Twenty Fifth National Conference on Artificial Intelligence*, 2010.

[44]H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins. Text classification using string kernels. *The Journal of Machine Learning Research*, 2:419–444, 2002.

[45]I. Mani, M. Verhagen, B. Wellner, C. M. Lee, and J. Pustejovsky. Machine learning of temporal relations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 753–760, 2006.

[46]I. Mani, B. Wellner, M. Verhagen, and J. Pustejovsky. Three approaches to learning tlinks in timeml.

Technical Report CS-07-268, Department of Computer Science, Brandeis University, Waltham, USA, 2007.
[47]M.-C. D. Marneffe, B. Maccartney, and C. D. Manning. Generating Typed Dependency Parses from Phrase Structure Parses. In *LREC 2006*, 2006.
[48]D. Mcclosky, E. Charniak, and M. Johnson. Effective self-training for parsing. In *In Proc. N. American ACL (NAACL)*, pages 152–159, 2006.
[49]D. Mcclosky and C. D. Manning. Learning Constraints for Consistent Timeline Extraction. In *Proc. EMNLP*, 2012.
[50]M. Mintz, S. Bills, R. Snow, and D. Jurafsky. Distant supervision for relation extraction without labeled data. In *ACL/AFNLP*, pages 1003–1011, 2009.
[51]M. Moens and M. Steedman. Temporal ontology and temporal reference. *Computational Linguistics*, 14:15–28, 1988.
[52]A. Y. Ng. Feature selection, L1 vs. L2 regularization, and rotational invariance. In *ICML*, 2004.
[53]T. Parsons. *Events in the Semantics of English: A Study in Subatomic Semantics*. The MIT Press, 1990.
[54]B. Partee. Some Structural Analogies between Tenses and Pronouns in English. *Journal of Philosophy*, 70:601–609, 1973.
[55]B. Partee. Nominal and Temporal Anaphora. *Linguistics and Philosophy*, 7:243–286, 1984.
[56]J. Pustejovsky, J. Castano, R. Ingria, R. Sauri, R. Gauzauskas, A. Setzer, and G. Katz. TimeML: Robust Specification of Event and Temporal Expression in Text. *IWCS-5, Fifth International Workshop on Computational Semantics.*, 2003.
[57]J. Pustejovsky, P. Hanks, R. Sauri, A. See, R. Gaizauskas, A. Setzer, D. Radev, B. Sundheim, D. Day, L. Ferro, and M. Lazo. The TIMEBANK corpus. In *Proceedings of Corpus Linguistics 2003*, pages 647–656, Lancaster, 2003.
[58]J. Pustejovsky and M. Verhagen. 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, 2010.
[59]M. Richardson and P. Domingos. Markov logic networks. In *Machine Learning*, 2006.
[60]S. Riedel, L. Yao and A. McCallum. Modeling relations and their mentions without labeled text. In ECML-PKDD, 2010.
[61]N. Schlaefer, J. Ko, J. Betteridge, G. Sautter, M. Pathak, and E. Nyberg. Semantic extensions of the Ephyra QA system for TREC 2007. In *Proc. TREC 2007*, 2007.
[62]S. Schockaert, M. D. Cock, D. Ahn, and E. Kerre. Supporting Temporal Question Answering: Strategies for Offline Data Collection. In *Proc. 5th International Workshop on Inference in Computational Semantics (ICoS-5)*, 2006.
[63]M. Verhagen, R. Gaizauskas, F. Schilder, G. Katz, and J. Pustejovsky. Semeval2007 task 15: Tempeval temporal relation identification. In *SemEval-2007: 4th International Workshop on Semantic Evaluations*, 2007.
[64]J. Pustejovsky, and M. Verhagen. SemEval-2010 Task 13: Evaluating Events, Time Expressions, and Temporal Relations (TempEval-2). In *SemEval-2010: 5th International Workshop on Semantic Evaluations*, 2010.
[65]R. Snodgrass. Of duplicates and septuplets. *Database Programming and Design*, 1998.
[66]M. Surdeanu, J. Tibshirani, R. Nallapati and C. D. Manning. Multi-instance multi-label learning for relation extraction. In *Proc. EMNLP*, 2012.
[67]S. Takamatsu, I. Sato and H. Nakagawa. Reducing wrong labels in distant supervision for relation extraction. In *Proc. ACL*, 2012.
[68]P. P. Talukdar, D. Wijaya, and T. Mitchell. Coupled temporal scoping of relational facts. In *Proc. WSDM*, 2012.
[69]P. P. Talukdar, D. Wijaya, and T. Mitchell. Acquiring temporal constraints between relations. In *Proc. CIKM*, 2012.
[70]S. Tamang and H. Ji. Adding smarter systems instead of human annotators: Re-ranking for slot filling system combination. In *Proc. CIKM2011 Workshop on Search and Mining Entity-Relationship data*, 2011.
[71]M. Tatu and M. Srikanth. Experiments with reasoning for temporal relations between events. In *COLING*, pages 857–864, 2008.
[72]B. Taylor. Tense and continuity. *Linguistics and Philosophy*, 1:199–220, 1977.
[73]C. Tenny and J. Pustejovsky. A History of events in Linguistic Theory. In *Events as Grammatical Objects*, pages 3–38. CSLI Publications, 2000.
[74]R. Tibshirani. Optimizing reinsertion: regression shrinkage and selection via the lasso. *J. R. Statist. Soc. B*, 58, No. 1, 267-288, 1996.
[75]R. Tibshirani. Regression shrinkage and selection via the lasso: a retrospective. *J. R. Statist. Soc. B*, 73, No. 3, 273-282, 2011.

[76]M. Trautwein. *The Time Window of Language.* Walter de Gruyter, 2011.
[77]Z. Vendler. *Linguistics in Philosophy*. Cornell University Press, 1967.
[78]M. Verhagen. *Times between the Lines*. PhD thesis, Brandeis, University, 2004.
[79]M. Verhagen. Temporal Closure in an Annotation Environment. *Language Resources and Evaluation*, 39, No. 2-3, 211-241, 2005.
[80]M. Verhagen, R. Sauri, T. Caselli and J. Pustejovsky. Semeval-2010 task 13: Tempeval 2. In *Proceedings of International Workshop on Semantic Evaluations (SemEval 2010)*, 2010.
[81]Y. Wang, B. Yang, L. Qu, M. Spaniol, and G. Weikum. Harvesting Facts from Textual Web Sources by Constrained Label Propagation. In *CIKM2011*, 2011.
[82]K. Yoshikawa, S. Riedel, M. Asahara, and Y. Matsumoto. Jointly identifying temporal relations with markov logic. In *Proc. the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 405–413, 2009.