Natural Language Engineering **21** (4): 607–652. © Cambridge University Press 2014
doi:10.1017/S1351324914000060

Construction and evaluation of event graphs

GORAN GLAVAŠ and JAN ŠNAJDER

University of Zagreb, Faculty of Electrical Engineering and Computing,
Text Analysis and Knowledge Engineering Lab, Unska 3, 10000 Zagreb, Croatia
email: {goran.glavas,jan.snajder}@fer.hr

(*Received 26 July 2013; revised 10 March 2014; accepted 14 March 2014; first published online 1 May 2014*)

Abstract

Events play an important role in natural language processing and information retrieval due to numerous event-oriented texts and information needs. Many natural language processing and information retrieval applications could benefit from a structured event-oriented document representation. In this paper, we propose *event graphs* as a novel way of structuring event-based information from text. Nodes in event graphs represent the individual mentions of events, whereas edges represent the temporal and coreference relations between mentions. Contrary to previous natural language processing research, which has mainly focused on individual event extraction tasks, we describe a complete end-to-end system for event graph extraction from text. Our system is a three-stage pipeline that performs anchor extraction, argument extraction, and relation extraction (temporal relation extraction and event coreference resolution), each at a performance level comparable with the state of the art. We present EvExtra, a large newspaper corpus annotated with event mentions and event graphs, on which we train and evaluate our models. To measure the overall quality of the constructed event graphs, we propose two metrics based on the tensor product between automatically and manually constructed graphs. Finally, we evaluate the overall quality of event graphs with the proposed evaluation metrics and perform a headroom analysis of the system.

1 Introduction

Real-world events are situations that *happen* or *occur* (Pustejovsky *et al.* 2003b). As our society becomes increasingly digital, there are a growing number of textual information sources (e.g., breaking news, investigative stories, police reports, tweets, historical texts, electronic health records) that are filled with descriptions of events. The ability to automatically extract and analyze events from text is now more important than ever, with applications that range from security and intelligence to journalism, media analysis, and historical research. In text, real-world events are represented by the so-called *linguistic events*, or *event mentions*, as more frequently referred to in computational linguistics (Rosen 1999). Due to ambiguity and vagueness of natural language, the mapping of real-world events and their relations (temporal, causal, *etc.*) to their linguistic counterparts introduces a loss of information. Real-world events may – at least in principle – be individuated by

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

their unique spatiotemporal extent (i.e., it is not possible for two real-world events to happen at exactly the same location and at exactly the same time) (Quine 1985). In contrast, event mentions are often referentially ambiguous because the exact spatiotemporal location of an event can rarely be inferred from text. Similarly, whereas two real-world events always stand in some temporal relation (e.g., *before*), the relationship is often not inferable from text. The inherent ambiguity of event mentions makes event extraction a particularly challenging task.

Although events and their roles in natural language have been studied extensively in linguistics (Vendler 1957; Davidson 1967; Pustejovsky 1991; Tenny and Pustejovsky 2000), these topics have received less attention in computational linguistics. Information extraction (IE) has traditionally focused more on entities and entity-related tasks, such as named entity recognition (NER) (McCallum and Li 2003; Nadeau and Sekine 2007) and entity coreference resolution (Soon, Ng and Lim 2001; Lee *et al.* 2011). In the past decade, however, research on event extraction has gained considerable momentum, boosted in particular by standards, such as TimeML (Pustejovsky *et al.* 2003b), and datasets, such as TimeBank (Pustejovsky *et al.* 2003a), as well as by shared evaluation tasks, such as Automatic Content Extraction (ACE 2005) Evaluation and TempEval (Verhagen *et al.* 2007; Verhagen *et al.* 2010). ACE and TempEval led to the establishment of standard terminology and event extraction tasks. Event mentions are considered to consist of event anchors, words that convey the core meaning of an event (e.g., '*killed*' or '*bombing*'), and event arguments, the protagonists and circumstances of events (e.g., agent, time, location). The most typical event extraction tasks include extraction of event anchors and arguments and the recognition of temporal relations between events. These tasks are typically considered in isolation (Ahn 2006; Bethard 2008; Bejan and Harabagiu 2010; Grover *et al.* 2010; Llorens, Saquete and Navarro 2010; UzZaman and Allen 2010; Jun and Min 2012); attempts to construct more complete structured event-based representations from event-oriented texts have been relatively rare (Bramsen *et al.* 2006; Kolomiyets, Bethard and Moens 2012).

Whereas research in natural language processing (NLP) has focused on the extraction of sentence-level event mentions, the second strand of research, carried out within the topic detection and tracking (TDT) program (Allan 2002), has focused on document-level events. In TDT, an event is defined as something that occurs in a certain place at a certain time (Yang *et al.* 1999), whereas a topic is defined as a set of news stories strongly related by some real-world event (Allan 2002). To group the news stories by topics, TDT has mostly relied on the traditional vector space model (SaltonWong and Yang 1975). More recent approaches extend the vector space model by differentiating between particular entity classes (e.g., named entities, noun phrases, collocations), among which named entities have been considered particularly important because they often identify the participants of an event (Hatzivassiloglou, Gravano and Maganti 2000; Kumaran and Allan 2004; Makkonen, Ahonen-Myka and Salmenkivi 2004). At the same time, the TDT community has recognized the need for a more structured representation of events (Makkonen 2003). Although traditional TDT approaches predominantly group news into flat clusters, structuring stories into hierarchical event threads has been proposed (Nallapati *et al.* 2004), as

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

well as tracking the evolution of events (Wei and Chang 2007; Yang, Shi and Wei 2009).

In spite of the obvious similarities, the two perspectives of events – sentence-level events in NLP and document-level events in TDT – have largely remained isolated from one another. In particular, no attempts have been made to bridge the two representation levels by introducing some intermediate structured representation of events. Such a representation could be grounded in linguistic events, but extended to cover more complex events at different levels of granularity and abstraction. A structured representation would enable the modeling of events in an event space that spans from linguistic events at the lowest level to topics at the highest level, effectively bridging the representation gap between the NLP and TDT notions of an event. For tasks related to event-oriented information needs, using such a structured document representation may lead to substantial performance improvements. For example, much event-oriented information needs involve structure that cannot be expressed with keyword-based queries (e.g., '*What are the countries that President Obama has visited and in which has his visit triggered protests*?'). Similarly, structured event-based document representation can be beneficial for temporal question answering (Ahn *et al.* 2006; Saquete *et al.* 2009) (e.g., '*Who won both the Boston and New York marathons three years in a row*?') and can allow for robust multi-source information extraction, in which missing and incorrectly extracted information can be compensated for by merging the information from multiple sources.

In this work, we aim to bridge the gap between the two notions of events: we represent document-level events as structures built from sentence-level events. More precisely, we structure the news stories into *event graphs* built from the individual event mentions extracted from the text. An event graph is a labeled graph in which nodes represent individual event mentions, and edges represent semantic relations between event mentions (e.g., temporal relations). Once represented as graphs of sentence-level event mentions, document-level events can be analyzed and compared using a wide variety of existing graph-based algorithms. For example, algorithms can be used for exact or approximate subgraph matching (Cordella *et al.* 2004; Tong *et al.* 2007) to identify common subevents or graph kernels (Gärtner, Flach and Wrobel 2003; Menchetti, Costa and Frasconi 2005) to measure the extent to which documents report on the same events (Glavaš and Šnajder 2013).

The main contribution of our work is the development of an end-to-end system for constructing event graphs from natural language text. Although the idea of graph-based event modeling is, in itself, language-independent, we focus on the processing English language texts, for which we achieve state-of-the-art performance on individual extraction tasks. Other important contributions of our work are novel evaluation metrics for evaluating the overall quality of the event-related information captured by event graphs and EvExtra, a large newspaper corpus annotated with factual event mentions and event graphs.

Constructing event graphs from raw text is a challenging task, as it requires (1) combining machine learning and rule-based methods for several individual tasks, with each requiring dataset annotation, model construction, and intrinsic evaluation,

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

and (2) pipelining the individual models into a seamless workflow. Our system combines several state-of-the-art NLP components into an integrated pipeline; we use both off-the-shelf components (e.g., for parsing, named entity recognition, entity coreference resolution) and components we developed on our own (extraction of event mentions, temporal relation extraction, and event coreference resolution). The final pipeline consists of three stages: (1) Supervised extraction of factual event anchors, (2) robust rule-based extraction of event arguments, and (3) supervised extraction of the relations between events (temporal relations and coreference of event mentions).

Following the standard practice, we conduct an intrinsic evaluation of each individual component of our pipeline, obtaining state-of-the-art results. While standard evaluation methodologies have been established for the individual tasks, such as anchor extraction (e.g., precision, recall, and F-score), there have been very few proposals for the joint evaluation of two or more extraction tasks (UzZaman and Allen 2011). Our work also makes a contribution in this respect by proposing and validating metrics that measure the overall quality of the extracted event-oriented information represented by event graphs. Two proposed metrics are based on the graph product (Hammack, Imrich and Klavžar 2011) between automatically constructed event graphs and their respective manually created reference graphs, considering all elements of an event graph (i.e., anchors, arguments, and relations). We evaluate our three-stage graph construction pipeline using the proposed metrics, providing insight into the overall accuracy of the event-oriented information extracted from the text. Finally, we perform a headroom analysis for each of the pipeline steps, and identify the pipeline components whose improvement would result in the largest improvements in overall graph accuracy.

The remainder of the paper is structured as follows. In Section 2, we describe the related research on event extraction and the structuring of event-oriented information from text. In Section 3, we describe the construction of event graphs. In Section 4, we present the metrics for evaluating the overall quality of event graphs. Section 5 describes the EvExtra corpus. In Section 6, we focus on the experimental evaluation of each step of the construction of an event graph. Finally, we draw our conclusions and outline the future work in Section 7.

2 Related works

Most of the work on event extraction focuses on individual tasks (e.g., temporal relation extraction). In this section, we describe the related works on the tasks relevant for our graph construction pipeline and evaluation metrics: event anchor and temporal relation extraction (Section 2.1), argument extraction (Section 2.2), event coreference resolution (Section 2.3), and temporal evaluation (Section 2.4).

2.1 Event anchor and temporal relation extraction

The first event extraction tasks were organized as part of the Message Understanding Conferences (MUC) (Grishman and Sundheim 1996), where events were viewed as

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

scenario templates defined for specific domains. The scenarios were predetermined (e.g., a shuttle launch) and the task was to fill the information slots for a given scenario. Early approaches and systems for extracting event-based information were predominantly rule-based (Humphreys *et al.* 1998; Aone and Ramos-Santacruz 2000). The improvements were mostly gained by including additional extraction patterns for scenarios from new domains (Aone and Ramos-Santacruz 2000). However, more recent approaches focus on automatically acquiring extraction templates (Yangarber *et al.* 2000; Surdeanu *et al.* 2003).

The emergence of standards for annotating event and temporal information in text, such as TimeML (Pustejovsky *et al.* 2003b), and the creation of the accompanying datasets, such as TimeBank (Pustejovsky *et al.* 2003a), triggered the transition from template-based event extraction to more robust and general approaches. The organization of shared evaluation tasks, such as ACE (2005) and TempEval (Verhagen *et al.* 2007; Verhagen *et al.* 2010) was another step forward. Similar to Message Understanding Conferences and early work on event extraction, ACE (2005) focused on domain-specific event extraction. The tasks included the extraction of event anchors and arguments (both tasks are framed as classification into one of the predetermined domain classes) as well as event coreference resolution. Some of the evaluated systems approached the tasks with machine learning methods (e.g., Ahn 2006). The first TempEval campaign (Verhagen *et al.* 2007) focused on three temporal relation extraction tasks: extraction of relations between events and temporal expressions, between events and document creation time, and between the main events of adjacent sentences. The second TempEval campaign (Verhagen *et al.* 2010), TempEval-2, introduced three additional tasks: extraction of event anchors, extraction of temporal expressions, and recognition of temporal relations between events from the same sentence, where one syntactically dominates the other. On the anchor extraction task, the best results were achieved by two rather different systems: (1) a supervised machine-learning approach using conditional random fields with a rich set of linguistic features (Llorens *et al.* 2010), and (2) a rule-based approach filtering head verbs and head nominalizations with WordNet-based attributes (Grover *et al.* 2010). On the temporal relation extraction task, the best-performing system used supervised machine learning with Markov logic networks (UzZaman and Allen 2010). Recently, Llorens, Saquete and Navarro-Colorado (2013) have demonstrated that semantic knowledge (lexical semantics and semantic roles) improves the recognition of event anchors and temporal relations.

The same tasks addressed in TempEval-2, although in a different domain, were also addressed in the i2b2 NLP challenge for clinical records (Sun, Rumshisky and Uzuner 2013), which focused on the temporal relations within clinical narratives. The participating systems extracted (1) clinically significant events (e.g., problems, tests, admissions, transfers), (2) temporal expressions (e.g., dates, durations, frequencies), and (3) temporal relations between the events and temporal expressions from a corpus of clinical discharge summaries. Machine learning-based approaches performed best on event detection, and rule-based methods performed best on temporal expression extraction and normalization, whereas hybrid methods that combined machine learning with heuristics performed best on the classification of

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

temporal relations. In contrast to the work done in the i2b2 challenge, we aim to determine the temporal relations between pairs of events instead of between events and temporal expressions.

Unlike the aforementioned approaches, which consider temporal relations between events in isolation (i.e., by extracting relations between pairs of events), other approaches have focused on constructing a more complete temporal structure from documents. Kolomiyets *et al.* (2012) extract narrative timelines of children's stories as dependency trees using two different parsing models – a deterministic non-projective shift-reduce parser and a graph-based maximum spanning tree parser. However, children's stories tend to have a simpler (i.e., more linear) temporal structure than other genres (e.g., news), which is also confirmed by their experiments in which a linear temporal sequence was found to be a very competitive baseline. Bramsen *et al.* (2006) represent the temporal structure of a document as directed acyclic graphs (DAGs). They consider only temporal precedence (relations *before* and *after*) and do not consider temporal overlap. Globally optimal directed acyclic graphs are induced using integer linear programming optimization. Compared with our work, their graphs represent a more coarse temporal structure because the nodes in their directed acyclic graphs represent textual segments rather than individual event mentions.

Extracting more complete temporal structures has also been the focus of the most recent shared temporal evaluation campaign, TempEval-3 (UzZaman *et al.* 2013). TempEval-3 featured the same tasks as TempEval-2 but provided an overall evaluation metric called *temporal awareness* as a unified measure of a system's ability to extract temporal expressions, event mentions, and temporal relations between them. TempEval-3 also introduced new corpora: a small platinum corpus for evaluation, manually annotated by task organizers, and a large silver corpus, automatically annotated by merging the predictions of the best performing systems from TempEval-2. ClearTK (Bethard 2013), the most temporally aware system, employed a pipeline of linear machine-learning models in combination with simple morphosyntactic features.

A related stream of research addresses the learning of narrative chains from the text (Chambers and Jurafsky 2008; Chambers and Jurafsky 2009; Jans *et al.* 2012). Chambers and Jurafsky (2008) define a narrative chain as a 'partially ordered set of events related by a common protagonist'. They consider only verbal events and, similar to Bramsen *et al.* (2006), only the relation of temporal precedence between two events. The set of events that constitute a chain is determined based on the pointwise mutual information between pairs of events estimated from a large corpus, after which the temporal ordering is determined using a binary classifier (*Before* versus *Other*) trained on TimeBank. In their subsequent work, Chambers and Jurafsky (2009) introduce *typed event chains*, in which shared arguments are constrained to a certain semantic role. In contrast to FrameNet or PropBank, the semantic roles are not predetermined but are instead represented as sets of words induced from a corpus in an unsupervised fashion. For example, in the narrative chain (*suspect* X) → (*hunt* X) → (*accuse* X) → (*convict* X), the shared argument X should have the semantic role of a 'criminal person,' defined by certain

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

words {*criminal*, *murderer*, *suspect*, *thief*}. Chambers and Jurafsky (2009) go on to define *event schemas* as sets of typed event chains as a means of modeling all the actors participating in a set of events. For example, if (*hunt* X) is part of one narrative chain, then (*Y suspect*) → (Y *hunt*) has to be a part of another chain, but both chains belong to the same narrative schema. Although related to our work, the extraction of narrative chains and event schemas differ from the extraction of event graphs in two important aspects. First, only verbal events are considered for participation in narrative chains, while nominal event mentions (e.g., '*elections*', '*murder*', '*match*') – often informationally very relevant – are ignored. Second, because event schemas are unordered sets of narrative chains, while narrative chains model only the temporal relations between events that share a protagonist, the temporal relations between event mentions with no common protagonists (e.g., '*Obama and Putin met only a week after the meeting between Sarkozy and Merkel in Athens*') are also ignored.

2.2 Argument extraction

In ACE, argument extraction was defined as a separate task and framed as a supervised classification problem. Ahn (2006) considers the named entities, temporal expressions, and numerical expressions as argument candidates. His system pairs all argument candidates with all event anchors within the same sentence and uses supervised classification to determine whether a candidate is indeed the argument of the anchor. However, in general, event arguments may belong to classes other than named entities, temporal expressions, or numerical expressions. Furthermore, Ahn uses domain-specific argument types, not allowing for cross-domain argument extraction.

As events anchors are very often predicates, argument extraction is related to the task of semantic role labeling (SRL), which seeks to induce shallow semantic representations of sentences by extracting predicate-argument structures (Gildea and Jurafsky 2002). SRL has been proven beneficial for many NLP tasks (Surdeanu *et al.* 2003; Melli *et al.* 2006; Moreda *et al.* 2011; Llorens *et al.* 2013). Approaches to SRL differ in the role sets they use; the two main role sets are the one based on FrameNet (Baker, Fillmore and Lowe 1998) and PropBank (Palmer, Gildea and Kingsbury 2005). The two role sets are of different granularity: FrameNet, which is based on semantic frame theory (Fillmore 1976), defines a very detailed representation of situations (frames) along with their conceptual roles (frame elements). In contrast, PropBank is theory-neutral and defines role labels on a per-verb basis.

Bejan and Hathaway (2007) use FrameNet frames to recognize the arguments of events. They identify arguments and their semantic roles through several processing steps, each of which introduces non-negligible performance loss: (1) identification (disambiguation) of the frame satisfied by the predicate; (2) identification of phrases that represent the arguments of a predicate; and (3) assignment of semantic roles to previously identified arguments. Predicates are typically verbs, and frame identification is similar to word sense disambiguation (WSD) (Bejan and Hathaway 2007). Since word sense disambiguation approaches generally perform significantly

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

worse on verbs than on nouns or adjectives (Navigli and Lapata 2007; Agirre and Soroa 2009), the correct frame is often not selected, although it is covered by the resource. The selection of an incorrect frame in turn typically triggers incorrect argument extraction and role labeling. This error is further exaggerated by the fine-grainedness of semantic roles in FrameNet.

The lack of coverage and robustness limit the effectiveness of FrameNet-based SRL approaches for event extraction. PropBank-based SRL approaches may be more adequate as they have a wider lexical coverage. Apart from the predicate-specific semantic roles, PropBank defines a set of modifier adjuncts, i.e., general arguments that any verb can optionally take (e.g., *Location*, *Time*, *Purpose*, *Manner*). PropBank-based SRL was the topic of the CoNLL-2005-shared task, in which most participants employed supervised machine-learning models that used phrase-structure features (Haghighi, Toutanova and Manning 2005; Surdeanu and Turmo 2005). Unlike CoNLL-2005 SRL systems, which label the predicate arguments based on constituency parsing information, our event argument extraction is rule-based and relies on dependency parsing information.

2.3 Event coreference resolution

Coreferent event mentions are mentions that refer to the same real-world event. By identifying coreferent event mentions, we can gather all of the information about the real-world event to which these mentions refer, even though they are scattered throughout the text. The task of event coreference resolution was first addressed by Humphreys, Gaizauskas and Azzam (1997) who arrange mentions from text in a hierarchical semantic network according to an event ontology, and coreference was determined pairwise by computing scores based on the distance of events in the ontology. Bagga and Baldwin (1999) build summaries for each event anchor in the document by selecting all sentences in which the same anchor or one of its nominalizations appears. To determine coreference, they analyze the overlap in named entities and temporal expressions between the events' summaries. Ahn (2006) addresses event coreference resolution using a maximum entropy classifier to determine the coreference of paired event mentions. Bejan and Harabagiu (2010) propose a clustering approach based on non-parametric Bayesian models for both within- and cross-document event coreference resolution. Their models are based on a rich set of linguistic features, including features that compare event arguments. However, they extract arguments according to semantic frames defined in PropBank (Kingsbury and Palmer 2002) and FrameNet (Baker *et al.* 1998), effectively limiting the approach to the predicates covered by those resources. Chen *et al.* (2011) propose a framework that integrates five different event coreference resolvers, arguing that coreferent pairs with different parts of speech (POS) exhibit different behaviors (e.g., the coreference of verb–verb event pairs differs from coreference of verb–noun event pairs). They train one binary classifier for each pair of POS-tags and use a rich set of linguistic features, including argument-matching features. Their argument extraction approach is limited to the extraction of arguments from anchor premodifiers and prepositional attachments.

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

2.4 Temporal evaluation

The work on temporal evaluation originates from the need to measure inter-annotator agreement on the annotations of temporal relations (Setzer, Gaizauskas and Hepple 2003; Tannier and Muller 2008). Setzer *et al.* (2003) propose performing transitive closure for a set of annotated temporal relations prior to measuring the inter-annotator agreement. Precision and recall are defined as the number of relations from the transitive closure of one annotator's annotations that can be found in the closure of another annotator's annotations. Tannier and Muller (2008) argue that relations introduced transitively should not be matched the same way as *core relations* (which cannot be inferred from other relations). Thus, they evaluate only the overlap in the core relations between the gold standard temporal annotations and the system-extracted relations. UzZaman and Allen (2011) propose a middle-ground approach: They define the precision as the proportion of the system-extracted relations that can be found in the transitive closure of the gold-annotated relations and the recall as the proportion of gold-annotated relations that can be found in the transitive closure of the system-extracted relations. The proposed *temporal awareness* metric has already been used for the TempEval-3-shared task (UzZaman *et al.* 2013).

The above-mentioned metrics, each in its own way, account for the temporal structure induced by temporal relations. However, these metrics are still inadequate for the evaluation of event graphs. First, we find these metrics to be biased toward evaluating temporal relations. An event graph that contains no correct temporal relations but contains all and only correct event mentions (anchors and their arguments) would be assigned a temporal awareness score of zero, which seems inappropriate. Furthermore, because temporal awareness essentially compares the sets of temporal relations, it cannot be used to measure the degree to which the structures of two event graphs correspond. In Section 4, we introduce two evaluation metrics to address these issues.

3 Construction of event graphs

In this section, we describe the event graph construction pipeline in detail. We begin with a formal definition of an event graph, describe the underlying assumptions, and proceed with the description of the individual components of the graph construction pipeline. The overall architecture of the graph construction pipeline is depicted in Figure 1.

3.1 Event graph

A graph is the representation of choice used when modeling relations between concepts. In our case, the concepts of interest are the individual real-world events, realized in text as event mentions. We define an *event graph* as a vertex- and edge-labeled mixed graph in which vertices represent the individual event mentions and edges represent the temporal and coreference relations between event mentions.

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

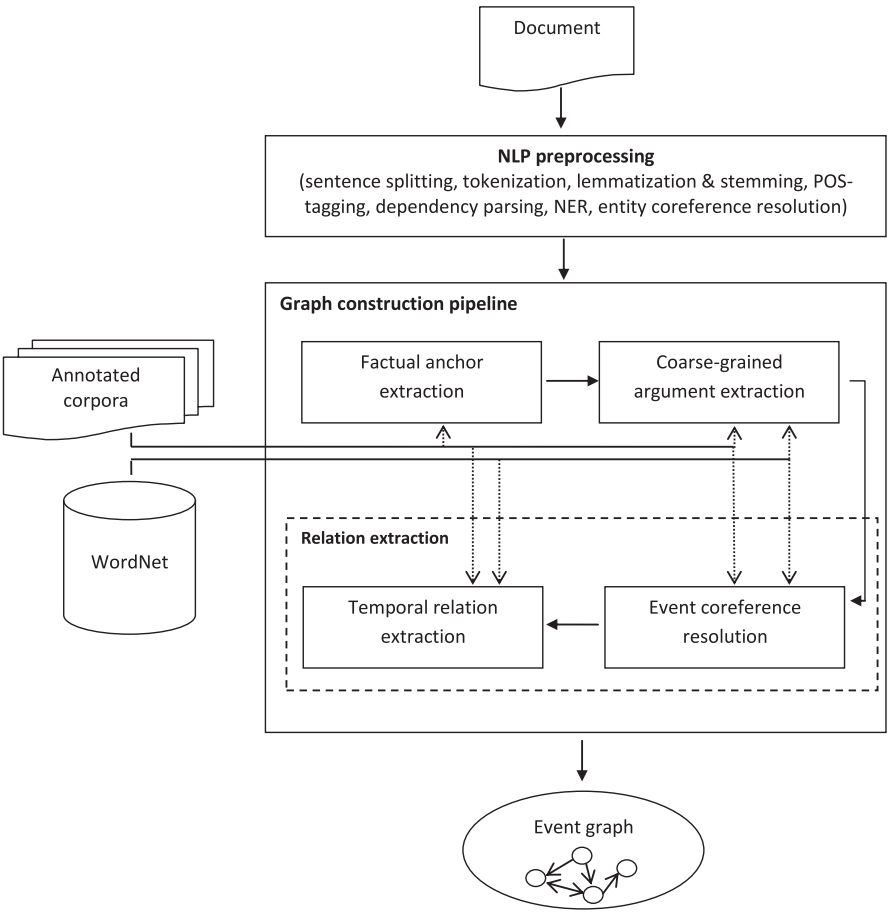


Fig. 1. Overall architecture of the event graph construction pipeline.

Given that some relations between events are symmetric (temporal *overlap*, temporally *equal*, or *coreference*), whereas others are not (temporally *before* or temporally *after*), an event graph may contain both directed and undirected edges; thus, we define an event graph as a mixed graph. Figure 2 shows an example of an event graph and the corresponding news story.[1]

Let M denote a set of extracted event mentions and R denote the set of event relations. An event graph G is a tuple

$$G = (V, E, A, m, r)$$

where V is the set of vertices, E is the set of undirected edges, A is the set of directed edges (arcs), $m : V \rightarrow M$ is a vertex labeling function mapping vertices to mentions from M, and $r : E \cup A \rightarrow R$ is an edge labeling function mapping edges to relations

[1] The figure is provided only as an illustrative example. The issue of optimal visual representation in terms of the effort required to understand the temporal structure of a document falls outside the scope of this paper. For a discussion of scalable visual representations of temporal information, the reader is referred to Verhagen (2007).

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

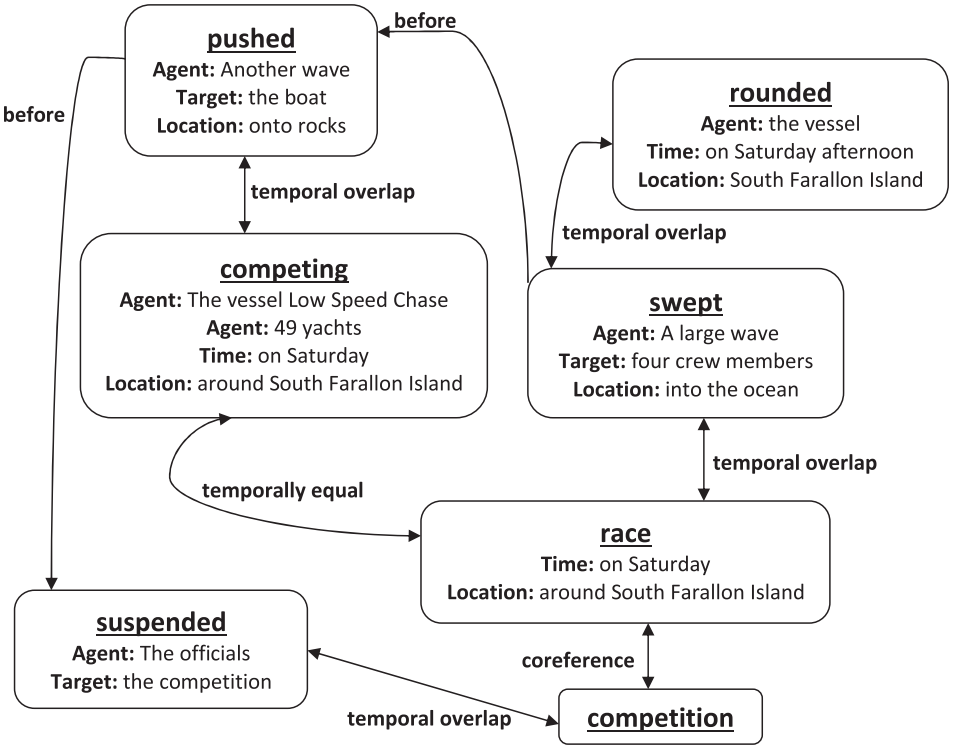


Fig. 2. Example of a reference event graph built from a news story.

from R. Each event mention from M is assigned a type (e.g., *Occurrence*, *Reporting*) and consists of an event anchor and a set of typed event arguments.

The above definition is general in that it does not specify any concrete sets for M and R; in practice, of course, these sets need to be specified in detail. The concrete definitions are likely to be application and domain-specific. In our work, we define the set of events M and the set of relations R based on the following assumptions.

Factuality of events. In principle, event-oriented information needs concern factual, real-world events. For example, in question answering a question about a factual event (e.g., '*Who killed Osama bin Laden*?') is much more likely to appear than a question about a non-factual event (e.g., '*Who did not win a medal in 2012* Olympics?' or '*When might Obama resign*?'). Similarly, in answering temporal questions, complex queries have been addressed by decomposing them into facts, which, in most cases, relate to factual event mentions, and by retrieving the answers for each fact individually (Saquete *et al.* 2009). Since we aim to represent real-world events as they actually occurred, in this work we focus exclusively on *factual event mentions* – mentions for which we can ascertain that the event to which they refer has indeed occurred or is occurring in the real world (Polanyi and Zaenen 2006; Saurí and Pustejovsky 2012). Thus, we do not consider negative (e.g., '*He did not win*'), hypothetical (e.g., '*He can win*'), future (e.g., '*He will win*'), and counterfactual event mentions ('*If he had won…*'), nor do we consider mentions referring to uncertain

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

events (e.g., '*Maybe he won*'). Event factuality is also essential for the temporal ordering of events (Karttunen and Zaenen 2005) because non-factual events can hardly be placed on a timeline.

Argument types. Aiming for robustness, we focus on extracting four generic types of event arguments: agent, target, location, and time. These types of arguments are informationally most salient as they directly answer four main wh-questions: 'Who did what to *whom*, *where*, and *when*?'. By considering only these four argument types, we can use a more generic, and thus a more robust approach to argument extraction. The approach is arguably more robust as it eliminates the coverage issues present in FrameNet-based approach of Bejan and Hathaway (2007), involves fewer processing steps, and thus reduces the chance of an error. Note that an event mention may have more than one argument of the same generic type (e.g., in '*Low Speed Chase and Stallion rounded the South Farallon Island…* ' both '*Low Speed Chase*' and '*Stallion*' are the agents of the mention '*rounded*').

Relation types. In principle, it is possible to identify many different semantic relations between events (e.g., temporal, causal, containment, semantic relations). In this work, we focus on two types of relations that intuitively appear to be most relevant for a wide range of possible applications: *temporal relations* (because events are temporally determined concepts; Pustejovsky *et al.* 2003b) and *event coreference relations* (because a single real-world event may be referred to by multiple mentions, potentially conveying complementary information). When resolving within-document event coreferences, we do not merge the vertices representing coreferent mentions. In other words, each graph vertex represents a single event mention and not a real-world event referred to by the mentions. For coreference resolution, we do not consider mentions other than event anchors (e.g., we do not consider pronominal anaphoric mentions).

3.2 Anchor extraction

Extraction of factual event mentions is the first step in the construction of event graphs. Apart from considering only factual event mentions, our definition of an event anchor differs from TimeML (Pustejovsky *et al.* 2003b) in three important respects, which we summarize below.

(1) We do not consider TimeML *states* as events. TimeML defines states as 'situations or predicates describing states or circumstances in which something obtains or holds true' (Pustejovsky *et al.* 2003b). Similarly, Smith (1999) characterizes states as situations that do not change over time, i.e., situations that hold over an undifferentiated interval. They are typically situations that hold true for a long period of time (e.g., '*used*' in '*It's not something we're not used to in the industry*'). We adopt this definition of a state and have chosen not to extract predicates describing states. We argue that in texts reporting on real-world events (e.g., breaking news, police reports, biographies, electronic health records), states are less informative than actions.

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

(2) We consider state changes, i.e., the transitions from one state to another (e.g., '*The CBI index rose 3.2%*') as a distinct class of events. We define a state change as an event that alters some intrinsic property of an entity participating in that event. For example, we consider the event 'fell' in '*The girl fell and started crying*' to be an instance of the *Occurrence* class because falling does not change any intrinsic property of the girl (her position is not an intrinsic property). In contrast, we consider the event 'fell' in '*The P&G stocks fell 5% in early trading*' to be an instance of the *State change* class because it describes a change in an intrinsic property of the stock – its value has changed. We note that state changes are mostly telic events (completed in some sense), whereas occurrences can be telic or atelic. In the TimeML standard (Pustejovsky *et al.* 2003b), state changes are not distinguished from other instances of the more general *Occurrence* class. However, we find this distinction to be useful in domains where the intrinsic properties of entities change frequently (e.g., electronic health records).

(3) Our event anchors are single words. Although in TimeML standard event anchors can be multiword expressions, a vast majority of event anchors (99.63%) in the TimeBank corpus are single words.

The definition of an event mention determines the complexity of the anchor extraction task. Due to the first point above, when compared with tasks based on the TimeML definition of an event (Bethard 2008; Verhagen *et al.* 2010), our task is arguably more demanding, as it also requires us to discriminate between events and states. Furthermore, our models need to discriminate between factual and non-factual events.

We address the task of event anchor extraction in two steps: anchor identification and event type classification. For each token, anchor identification predicts whether it is an event anchor. Note that we do not merge adjacent event tokens. Instead, we treat adjacent event tokens as two independent event mentions. We then classify the tokens identified as event anchors into one of the following five classes: *Occurrence*, *Intentional action*, *Perception*, *Reporting*, and *State change*. We adopted the first four classes from the TimeML standard and added the *State change* class for event mentions that refer to transitions from one state to another.

We use supervised machine learning models with the same rich set of lexical, POS, and syntactic features for both subtasks. We compute most of the features using the basic linguistic and preprocessing tools from the Stanford Core NLP library.[2] We use the following features:

- *Token features*: These consider only current token information. We use the following token features: *word*, *lemma*, *stem*, *ending* (the suffix of the word from the last vowel, or the penultimate vowel if the last letter of the word is a vowel), *POS tag*, and *coarse POS tag* (noun, verb, adjective, or others). Similar sets of features have been used in most previous approaches to event

[2] `http://nlp.stanford.edu/downloads/corenlp.shtml`

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

anchor extraction (Bethard 2008; Llorens *et al.* 2010; UzZaman and Allen 2010).

- *Context features*: These are all of the token features (e.g., *word*, *lemma*, *stem*) computed for all the tokens in a two-token symmetric window (two tokens to the left and two tokens to the right of the current token). Feature windowing for the anchor extraction task has also been proposed in previous works (Bethard 2008; Llorens *et al.* 2010; UzZaman and Allen 2010).
- *Syntactic features*: These include all syntactic dependency relations of a token, the chunk types (e.g., NP, VP, PP) of the token chunk and its adjacent chunks to the left and the right. Similar dependency-based syntactic features have been proposed by Bethard (2008), while Llorens *et al.* (2010) based their syntactic features on phrase-structure trees. We additionally employ a feature indicating whether the token has a nominal subject (i.e., it governs a *nominal subject* dependency relation), a feature indicating whether the token is a direct object (i.e., dependent of a *direct object* dependency relation), and a feature indicating whether the token has a direct object (governs a *direct object* dependency relation). The Stanford dependency parser (De Marneffe, MacCartney and Manning 2006) was used to compute the values of syntactic features.
- *Modifier features*: These aim to discriminate between factual and non-factual mentions by describing different modifiers for the token: the modality modifier (e.g., 'may' in '*Stallion may have rounded*...'), the auxiliary verb (e.g., '*had*' in '*Low Speed Chase had already finished the race*'), the determiner of the token (e.g., 'The' in '*The race around South Farallon Island*...'), and a feature indicating whether the current token is negated (e.g., true for mention '*finish*' in '*Stallion did not finish the race*'). We also use dependency relations to compute the values for modifier features (the relation *mod* for modality modifiers, *neg* for negations, *aux* for auxiliaries, and *det* for determiners).
- *Other features*: These include a feature indicating whether the word is capitalized (named events, such as '*World Cup*' are capitalized), features indicating whether the word and lemma of the current token are more frequently annotated as an *event* or *not event* on the training set, and features indicating the most frequent event type assigned to the word and lemma on the training set (e.g., *Reporting* for word '*said*' and lemma '*say*').

Considering the large size of the feature space (approximately 100K), induced by the large number of lexical features, we chose to use linear models, L2-regularized logistic regression and linear support vector machine (SVM). We used the LibLinear implementation of the algorithms (Fan *et al.* 2008).

3.3 Argument extraction

We use a rule-based approach for event argument extraction. As we are considering a small number of generic argument types, we can use a relatively small number of linguistically motivated extraction rules to reliably identify the argument and its type by considering the dependency relations between the argument and anchor.

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

The underlying assumption is that certain types of arguments tend to be realized in certain syntactic roles. Each extraction rule specifies a dependency pattern and a set of possible argument types, whereas a dependency pattern defines the set of possible dependency relations between the event anchor and head word of the argument.

Argument extraction consists of two steps: (1) Argument detection, which syntactically identifies argument candidates, and (2) argument type classification. Some of the syntactic patterns are semantically unambiguous, i.e., they can be used to both identify the argument and in most cases unambiguously identify its type. For example, the syntactic role of a nominal subject is typically associated with the semantic type of an agent. Other patterns are semantically ambiguous, in the sense that the argument type is not uniquely determined by the syntactic role. For example, the prepositional object of an event anchor may be a target (e.g., *'…competing against 48 opponents'*), location (e.g., *'…competing around South Farallon Island'*), or a time specification (e.g.,*'…competing on Saturday'*) of the event. To disambiguate the argument type in such cases, we use additional rules based on named entity recognition (Finkel, Grenager and Manning 2005), temporal expression extraction (Chang and Manning 2012), and WordNet-based semantic similarity (Wu and Palmer 1994). Next, we describe the two extraction steps in more detail.

3.3.1 Argument detection

We define the syntactic patterns used within extraction rules in terms of non-collapsed dependency relations provided by the Stanford dependency parser (De Marneffe *et al.* 2006). We designed the patterns based on the analysis of a manually annotated development set (cf. Section 6.2) by incrementally adding the rules that increased the recall without significantly decreasing the precision. To prevent overfitting, we preferred more general rules (rules whose patterns have fewer dependency relations) over the more complex rules. For the same reason, we chose to use only unlexicalized patterns.

We use two types of syntactic patterns: *extractive patterns* and *distributive patterns*. Extractive patterns identify head words of arguments of interest (agent, target, time, and location) and assign them to event anchors. A list of extractive patterns is provided in Table 1. Extractive patterns identify only the head words of event arguments. Given that arguments are often multiword expressions, we identify the full extent of the argument as the chunk containing the argument head word. For this task we use the chunker from the OpenNLP library.[3]

Distributive patterns consider the cases in which one argument is shared between two or more events (*argument sharing*), or when one event has two or more arguments of the same type (*argument conjunction*).

Argument sharing considers cases in which two event anchors occur in a certain syntactic relation that indicates that an argument is being shared, although it is syntactically linked to only one of the anchors. For example, in '*Low Speed Chase hit*

[3] `http://opennlp.apache.org/`

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

Table 1. *Extractive patterns for event argument extraction (anchors are underlined, arguments are shown in bold)*

Pattern name	Dependencies	Example	Argument type
Nominal subject	$nsubj(x, y)$	'**A wave** swept four crew members' (x = 'swept', y = 'wave')	Agent
Passive nominal subject	$nsubjpass(x, y)$	'**The boat** was pushed on the rocks' (x = 'pushed', y = 'boat')	Target
Direct object	$dobj(x, y)$	'A wave swept **four crew members**' (x = 'swept', y = 'members') 'Stallion rounded **Farallon Island**' (x = 'rounded', y = 'Island')	Target Location
Indirect object	$iobj(x, y)$	'Stallion allowed **Chase** a head start' (x = 'allowed', y = 'Chase')	Target
Prep. object	$prep(x, z)$, $pobj(z, y)$	'Stallion raced **on Sunday**' (x = 'raced', y = 'Sunday', z = 'on') 'Stallion raced **around Farallon Island**' (x = 'raced', y = 'Island', z = 'around') 'Stallion competed against **Chase**' (x = 'competed', y = 'Chase', z = 'against') 'The conflict was initiated by **Chase**' (x = 'initiated', y = 'Chase', z = 'by')	Time Location Target Agent
Participial modifier	$partmod(x, y)$	'**The boat** suffering damage' (x = 'boat', y = 'suffering') '**The boat** sunk on the shore' (x = 'boat', y = 'sunk')	Agent Target
Relative clause modifier	$rcmod(x, y)$, $nsubj(y, z)$	'**The boat** which won the race' (x = 'boat', y = 'won', z = 'which') '**The crewmen** storm swept' (x = 'crewmen', y = 'swept', z = 'storm')	Agent Target
Infinitival modifier	$infmod(x, y)$, $nsubj(z, x)$	'The first boat to beat Stallion was **Chase**' (x = 'boat', y = 'beat', z = 'Chase')	Agent
Possession modifier	$poss(x, y)$	'**Yesterday**'s race' (x = 'race', y = 'Yesterday') '**Stallion**'s victory' (x = 'victory', y = 'Stallion')	Time Agent
Noun compound	$nn(x, y)$	'**Farallon Island** race' (x = 'race', y = 'Island') '**Sunday** competition' (x = 'competition', y = 'Sunday') '**Stallion** initiative' (x = 'initiative', y = 'Stallion')	Location Time Agent
Open clausal complement	$xcomp(x, y)$, $nsubj(x, z)$	'**Stallion** managed to get past Chase' (x = 'managed', y = 'get', z = 'Stallion')	Agent

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

and 'damaged the Stallion', '*Low Speed Chase*' is the agent for both event mentions 'hit' and '*damaged*'. However, the *Nominal subject* pattern (cf. Table 1) would identify '*Low Speed Chase*' as the agent of only the anchor '*hit*'. Argument sharing is indicated with conjunction (e.g., *conj(hit, damaged)* in '*Low Speed Chase hit* and '*damaged the Stallion*') and open clausal complement (e.g., *xcomp(managed, avoid)* in '*Stallion managed to avoid the disaster*'). An argument can only be shared with events that have not already been assigned an argument of the same type. For instance, in '*The Stallion provoked and the Low Speed Chase answered*', we would not distribute an agent '*The Stallion*' to the event '*answered*' nor an agent '*Low Speed* '*Chase*' to the event '*provoked*', even though the anchors are conjoined.

The *argument conjunction* rule handles the cases in which one event has multiple arguments of the same type that are mutually syntactically conjoined. For instance, in '*Stallion and Low Speed Chase were seriously damaged in the crash*' both '*Stallion*' and '*Low Speed Chase*' are the targets of '*damaged*'. However, the *Passive nominal subject* pattern (cf. Table 1) would identify only '*Low Speed Chase*' as a target of '*damaged*'. By distributing the anchor '*damaged*', we also assign '*Stallion*' as its target.

It should be noted that the proposed extraction rules are designed specifically for the English language, and thus adapting the pipeline to a new language would require rewriting or at least adapting these rules. The development of the rules for English was a three-person-week effort.

3.3.2 Argument type disambiguation

Some extraction rules require an additional disambiguation step because the argument types cannot be uniquely determined from their syntactic roles. For example, variously typed arguments (e.g., time, location, agent, target, manner, cause) may be syntactically realized as a prepositional object of an event anchor. Our goal is to identify arguments denoting time, location, agent, or target. We define a set of disambiguation rules for each ambiguous extraction pattern. Similar to the extraction rules, we designed the disambiguation rules based on the analysis of the annotated development corpus (cf. Section 5.2), preferring more general rules over more specific rules.

We present the rules for determining the semantic type of a prepositional object, given that the pattern can be satisfied by arguments of all four semantic types. The rules apply in the order in which they are listed.

(1) If an argument is a named entity (NE) of type *Location*, it is assigned the location type; if it is a temporal expression, it is considered to be a temporal argument of the event.
(2) If an argument is a named entity of type *Person* or *Organization* and the preposition between the anchor and the argument is '*by*', then the argument is an agent. If the preposition is '*against*', then the argument is a target.
(3) If the prepositional object is introduced by a temporal preposition ('*before*', '*after*', '*during*', '*since*', and '*until*'), the argument is considered to be of temporal type. If it is introduced by a locative preposition ('*above*', '*below*', '*behind*',

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

'*inside*', '*around*', '*along*', '*across*', '*near*', and '*through*'), the argument is a location of the event. As noted by Derczynski and Gaizauskas (2013), some temporal signals – some of which we use here, such as '*before*' and '*near*' – are ambiguous and can have both temporal and spatial reading. However, our choice of temporal and locational signals is rather pragmatic – we choose the prepositions based on the number of times they occur in the corpus (cf. Section 5.2) as locative and temporal signals. For example, in our development set, '*before*' has never occurred as a locational signal, nor has '*near*' occurred as a temporal signal. We leave a more systematic treatment of this issue for the future work.

If none of the previous rules apply, we compare the semantic similarity of the argument's head word against three sets of concepts from WordNet. The sets T, L, and G represent temporal, locative, and general (non-locative and non-temporal) concepts respectively:

$$L = \{location, geographical_area, vehicle, building, transport, facility\}$$
$$T = \{time, duration, time_unit, clock_time, time_period\}$$
$$G = \{object, concept, person, event\}$$

We defined the sets of locative and temporal concepts by identifying the most common WordNet hypernyms of the temporal and locative argument heads, manually annotated on the development set. For example, we observed that vehicles and facilities often make locative event arguments (e.g., '*in the car*', '*at the discotheque*'). We defined the concept set G by selecting the most general non-temporal and non-locative concepts from WordNet.

The semantic similarity between a word w and a concept set S is the maximum of the Wu and Palmer (1994) similarity between w and the concepts from S:

$$ssim(w, S) = \max_{s \in S}(wupalmer(w, s))$$

The semantic type of the argument head word w is then determined based on its semantic similarity to the concept sets L, T, and G. The argument is considered locative if $ssim(w, L) > ssim(w, T)$ and $ssim(w, L) > ssim(w, G)$. Similarly, the argument is considered temporal if $ssim(w, T) > ssim(w, L)$ and $ssim(w, T) > ssim(w, G)$.

3.4 Relation extraction

In this work, we restrict our extraction to two main types of relations between events: temporal relations and event coreference relations. We use supervised machine learning models for both tasks, which we describe below.

3.4.1 Temporal relations

Most of the related approaches to temporal relation extraction are restricted in one of two ways: they either consider a subset of the possible event pairs (e.g., salient events of adjacent sentences or events from the same sentence in which

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

Table 2. *Temporal relation types mapping*

TimeML type	EvExtra coarse-grained type
I_BEFORE, BEFORE	*Before*
I_AFTER, AFTER	*After*
SIMULTANEOUS, IDENTITY	*Equal*
INCLUDES, IS_INCLUDED, BEGINS, BEGUN_BY, ENDS, ENDED_BY , DURING	*Overlap*

one syntactically dominates the other) (Bethard 2008; Verhagen *et al.* 2010) or they assume a partial temporal ordering between events and represent the temporal structure as trees or directed acyclic graphs (Bramsen *et al.* 2006; Kolomiyets *et al.* 2012). Both restrictions may easily lead to some temporal information remaining uncovered. Syntactic and discourse constraints narrow down the set of pairs for which temporal relations can be determined. Furthermore, imposing partial ordering implies that every temporal relation is reflexive, transitive, and antisymmetric. Since this assumption does not hold for every temporal relation of interest (specifically, *overlap* is not transitive, whereas *equal* is not antisymmetric), it can lead to spurious relations being extracted (e.g., spurious *overlap* relation due to incorrectly assumed transitivity).

In contrast, there is a rationale for restricting the pairs of events to be considered: determining the temporal relations between very distant pairs of event mentions in a document is impractical due to the large span of text between them (the information that needs to be 'understood' to determine the relation). Consequently, the effort required to annotate the temporal relations between all pairs of event mentions is simply too large. Moreover, due to discourse coherence (Wolf and Gibson 2005) of narratives, temporally related event mentions tend to occur close to each other. Thus, we chose to consider all pairs of events from the same sentence and adjacent sentences. However, we make no further restrictions by assuming certain syntactic relations between events or by assuming partial temporal ordering.

Our temporal relation extraction procedure consists of two steps. In the first step we identify pairs of event mentions between which temporal relation can be established. In the second step we classify the type of relations identified as establishable in the first step. Aiming for robustness rather than fine-grained temporal semantics, we differentiate between four coarse-grained types of temporal relations – *before*, *after*, *overlap*, and *equal*. It has been observed that distinguishing between semantically fine-grained temporal relations is hard and that is why previous evaluations have also employed coarse-grained temporal relation types (Verhagen *et al.* 2007; Verhagen *et al.* 2010). The mapping between the TimeML relation types and our coarse-grained types is shown in Table 2. We use the same rich set of features to both identify and classify temporal relations.

- **Positional features** measure the distance between and relative position of two event anchors. They include a feature indicating whether the event anchors are in the same sentence, a feature indicating whether the anchors are in

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

adjacent sentences, a feature indicating whether the events are adjacent (no other anchor is in between), and the distance between anchors (in number of tokens).

- **Lexical features** include the word, lemma, stem, and POS of both event anchors. In addition, we use a feature indicating whether the word forms of event anchors are the same, a numerical feature indicating the semantic similarity of the anchors (measured on WordNet using the Wu and Palmer's algorithm (1994)), and the bag-of-words (BoW) between the two anchors (Bethard 2008). Similar basic lexical and POS features have been proposed in previous works (Bethard 2008; Llorens *et al.* 2010; UzZaman and Allen 2010).
- **Syntactic features** are computed only for event pairs from the same sentence and include the syntactic path between the anchors (concatenation of the dependency relation labels on the path between the anchors in the dependency tree) and binary features indicating whether the first event anchor syntactically dominates the second and *vice versa.* Similar features for capturing the syntactic path between the events have been proposed previously by Llorens *et al.* (2010). In additionally, we use the *adverbial complement marks* of both anchors as features. An adverbial complement mark is the adverbial modifier of an event anchor that is in an *adverbial complement* relation to another event anchor (i.e., one of the events is a predicate of an adverbial clause governed by the other event), which often relates to a *heading preposition* feature proposed by Llorens *et al.* (2010). For example, in '*The storm began before Stallion rounded South Farallon Island*', the preposition '*before*' is the adverbial complement mark of the anchor '*rounded*'.
- **Modifier features** include all features that describe the modal, auxiliary, negation, and determination modifiers for both event anchors. These are the same features used for anchor extraction models (cf. Section 3.2).

Due to the large number of features (approximately 200K), for temporal relation extraction, we also chose to use linear discriminative models (logistic regression and linear SVM) using the LibLinear implementation (Fan *et al.* 2008).

3.4.2 Coreference relations

The edges of event graphs representing coreference relations are established by resolving pairwise within-document coreference of event mentions (e.g., mentions '*race*' and '*competition*' in the example in Figure 2). We use a supervised model with a set of features that we believe best capture the similarity between parts of event mentions, i.e., between factual event anchors and event arguments of coarse-grained semantic types. All the features were designed so that they indicate high similarity if the event mentions corefer.

- *The greedy BoW overlap* is computed by aligning the semantically similar lemmas from the BoW of the first event with lemmas from the BoW of

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

the second event. The BoW of an event mention contains lemmas from its event anchor and all arguments. This amounts to a weighed bipartite graph matching problem, which we solve as follows: In each iteration, we select the pair of lemmas (l_1, l_2) (where l_1 and l_2 are the lemmas from the BoW set of the first and second event mention respectively) with the highest semantic similarity. The semantic similarity between lemmas is measured on WordNet using the algorithm of Wu and Palmer (1994). Let P be the set of all pairs constructed with this greedy alignment. The final value for this feature is computed as the weighted sum of the semantic similarities between the lemmas for all pairs in P, normalized by the cardinality of the larger of the two BoWs:

$$alignedOverlap(e_1, e_2) = \frac{\sum_{(l_1,l_2)\in P} w(l_1, l_2) \cdot wupalmer(l_1, l_2)}{\max\left(|bow(e_1)|, |bow(e_2)|\right)}$$

where $w(l_1, l_2)$ is the weighting function that equals the larger of the *information contents* of lemmas l_1 and l_2. The information content (Resnik 1999) allows us to place more importance on lemmas occurring less frequently in a large corpus. The information content of lemma l in corpus C is computed as follows:

$$ic(l) = -\ln P(l) = \ln \frac{\sum_{l' \in C} freq(l')}{freq(l)}$$

Similar alignment strategies have been successfully applied for measuring semantic textual similarity (Šarić *et al.* 2012) and evaluating machine translation (Lavie and Denkowski 2009).

- *The named entity features* measure the overlap in the named entity occurrences between the two events, looking for named entity matches regardless of the argument roles occupied by those named entities. We extracted named entities using the recognizer from the Stanford CoreNLP library (Finkel *et al.* 2005). The *capital letter overlap* feature measures the overlap of capitalized words (as one form of rudimentary named entity recognition) between the two events by counting the number of the same capitalized words in both mentions. *Person*, *location*, *organization*, and *temporal expression overlap* features measure the overlap of named entities of the respective types by counting the number of overlapping named entities of the respective types. The *numbers overlap* feature measures the overlap in numbers between the two events. Numbers often occur as part of dates, temporal expressions, money, or percentages. Two event mentions differing in occurring numbers are a strong indicator that the mentions do not corefer (e.g., '*Stallion won two competitions in a row*' versus '*Stallion won seven competitions in a row*').
- *The anchor and argument comparison features* measure the overlap between anchors and arguments of two mentions respectively. As event anchors convey the core meaning of the event, coreferent mentions will have semantically closely related anchors. The same observation holds for arguments of matching types. The BoW and named entity overlap indicate close semantic relatedness, but not necessarily coreference. Consider, for example, the events

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

'*The Low Speed Chase* defeated *Stallion*' and '*The Stallion* defeated *Low Speed Chase*'. These mentions are identical if observed only through lexical overlap features, but do not corefer. We use one feature to measure the semantic similarity of the two anchors on WordNet, using the algorithm of Wu and Palmer (1994). For each of the four main argument types (*agent*, *target*, *location*, *time*) we use two comparison features: *argument coreference match* and *argument head word similarity*. The *argument coreference match* determines whether the arguments belong to the same entity coreference chain. To this end, we use the entity coreference chains provided by the Stanford coreference resolution tool (Lee *et al.* 2011). The *argument head word similarity* measures the semantic similarity of argument heads, using the Wu and Palmer (1994) algorithm. The comparison of event arguments has also been proposed by Bejan and Harabagiu (2010), but instead of comparing arguments of coarse-grained semantic types, they focus on comparing the frame elements with matching semantic roles. Chen *et al.* (2011) also use argument matching features, but they only consider arguments from verb premodifiers and prepositional attachments.

- *Other features* include a binary feature indicating whether the mentions are from the same sentence, a binary feature indicating whether the mentions belong to adjacent sentences, and a concatenation of the coarse POS tags of mention anchors (e.g., VV, VN, NN, VA, *etc.*). Similar features have been proposed in related works on event coreference resolution (Bejan and Harabagiu 2010; Chen *et al.* 2011).

We use SVM with radial basis kernel as our supervised model. The use of the kernel is motivated by the fact that, in contrast to the previous two models, for this model, the number of features is considerably smaller (42 features in total) than the number of training instances. We used the LibSVM implementation for the SVM with radial basis kernel (Chang and Lin 2011).

Note that all relations between factual event mentions except the temporal *overlap* are transitive relations. Thus, we can apply transitive closure to introduce additional relations into the graph. However, the resulting quality of the event graph directly depends on the accuracy of the initially extracted relations. By applying transitive closure, a single incorrectly classified relation (e.g., a relation recognized as temporally *before* instead of temporal *overlap*) may introduce a chain of spurious relations into the graph. We comment on this issue in Section 6.4, where we evaluate the overall quality of event graphs.

4 Evaluation of event graphs

In this section we consider the evaluation of event graphs. Each individual extraction task in the graph construction pipeline can be evaluated intrinsically using standard evaluation metrics such as precision, recall, and F-score. However, measuring the performance of the individual steps does not reflect the overall quality of an event graph as errors from earlier steps propagate down the pipeline. For example, if a

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

single event anchor is omitted in the anchor extraction step, all relations between that anchor and other anchors will be omitted as well.

We propose two evaluation metrics to measure the overall quality of event graphs. The proposed metrics quantify the overlap between the extracted event graphs and their respective manually created reference graphs based on the notion of a tensor graph product. The metrics take into account all components that comprise an event graph – the event anchors, event arguments, and relations (temporal and coreference). The proposed evaluation metrics exhibit desirable properties of standard metrics, as we demonstrate below.

4.1 Tensor product for labeled graphs

We use the tensor product of graphs as a mechanism to capture exact overlaps between the constructed event graphs and reference event graphs. A tensor product for the labeled graphs (Borgwardt 2007; Hammack *et al.* 2011) is a binary operation that takes two labeled graphs G and G' as input and results in a product graph, denoted as $G \times G'$, with vertices from the set $V(G) \times V(G')$.

Let $\mathscr{L}(v)$ and $\mathscr{L}(v_1, v_2)$ denote the labels of vertex v and edge (v_1, v_2) respectively. The vertices $V(G \times G')$ and edges $E(G \times G')$ of the product graph are defined as follows. The set of vertices comprises all pairs of vertices from G and G' whose labels match:

$$V(G \times G') = \{(v, v') \mid v \in V(G), v' \in V(G'), \mathscr{L}(v) = \mathscr{L}(v')\}$$

Analogously, the set of edges comprises all edges between pairs of matching vertices that are equally labeled in both input graphs:

$$E(G \times G') = \{\big((v_1, v_1'), (v_2, v_2')\big) \in V(G \times G') \times V(G \times G') \mid (v_1, v_2) \in E(G), (v_1', v_2') \in E(G'), \mathscr{L}(v_1, v_2) = \mathscr{L}(v_1', v_2')\}$$

We apply the above definition of the tensor product of graphs to the automatically extracted graph $G_C = (V, E, A, m, r)$ and reference graph $G_R = (V', E', A', m', r')$. Since we wish to identify the exact overlap between G_C and G_R, we require a strict match between vertex labels and edge labels of the two graphs. The matching function is defined as follows:

(1) The labels of vertices $v \in V(G_C)$ and $v' \in V(G_R)$ match, denoted $\mathscr{L}(v) = \mathscr{L}(v')$, iff the anchors and all arguments of event mentions $m(v)$ and $m'(v')$ match. The anchors of $m(v)$ and $m'(v')$ match iff they point to the same token from text and have the same event type (e.g., *Occurrence*). Sets of arguments of $m(v)$ and $m'(v')$ match iff they are identical, i.e., have the same head word and the same type.
(2) The labels of edges $(v_1, v_2) \in E(G_C) \cup A(G_C)$ and $(v_1', v_2') \in E(G_R) \cup A(G_R)$ match iff the corresponding relations are identical, i.e., $\mathscr{L}(v_1, v_2) = \mathscr{L}(v_1', v_2') \Leftrightarrow r(v_1, v_2) = r'(v_1', v_2')$.

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

4.2 Evaluation metrics

The tensor product of graphs measures the exact overlaps between G_C and G_R; intuitively, the product graph will be larger as G_C becomes more similar to G_R. Thus, we define two measures for the overall quality of an event graph: *relative overlap size* (ROS) metric and *largest connected component* (LCC) metric. The ROS metric measures the size of the overlap between the constructed graph and the reference graph regardless of the structure of that overlap, whereas the LCC metric accounts for the structure of the overlap by considering the largest connected components of event graphs. In the context of news stories, ROS measures the ability of the system to capture all event-oriented information regardless of whether it belongs to the main narrative or to the side narratives of the story, whereas LCC measures the ability of the system to capture the structure of the main narrative (assuming that the largest connected component of the event graph corresponds to the main narrative of the news story). Both metrics abide by the following requirements:

(1) The scores assigned by the metric reflect the overall quality of the automatically constructed event graph well.
(2) The metric is easily interpretable and can be related to the standard precision and recall metrics.

To satisfy the latter, the metrics were designed to penalize for information from G_R missing in G_C and spurious information in G_C not existing in G_R.

Relative overlap size. This metric measures the relative size of the product graph $G_C \times G_R$ compared with both G_C and G_R. The size of a graph is computed as the sum of number of vertices and number of edges, i.e., $|V(G)| + |E(G)|$. We define the relative overlap size in terms of ROS-based precision and recall defined as follows:

$$ROS_P(G_C, G_R) = \frac{|V(G_C \times G_R)| + |E(G_C \times G_R)|}{|V(G_C)| + |E(G_C)|}, \tag{1}$$

$$ROS_R(G_C, G_R) = \frac{|V(G_C \times G_R)| + |E(G_C \times G_R)|}{|V(G_R)| + |E(G_R)|} \tag{2}$$

ROS is computed as the harmonic mean between the precision and recall:

$$ROS(G_C, G_R) = 2 \cdot \frac{ROS_P(G_C, G_R) \cdot ROS_R(G_C, G_R)}{ROS_P(G_C, G_R) + ROS_R(G_C, G_R)} \tag{3}$$

ROS measures the overall amount of overlapping information that the automatically extracted graph G_C shares with the reference graph G_R. However, this metric does not consider the structure of the overlap (the structure of the graph product G_P). The same ROS score may be produced for a constructed graph that shares several fragmented components with the reference graph as for one that shares a single larger component with the reference graph.

Largest connected component. This metric considers the structure of the overlap between the graphs. It first computes the tensor product between the first graph and the largest weakly connected component of the second graph, then compares

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

the size of the largest weakly connected component of that product with the size of the largest weakly connected component of the second graph. Let $lcc(G)$ denote the largest weakly connected component of graph G. We define the metric in terms of LCC-based precision and recall defined as follows:

$$LCC_P(G_C, G_R) = \frac{|V(lcc(G_R \times lcc(G_C)))|}{|V(lcc(G_C))|} \tag{4}$$

$$LCC_R(G_C, G_R) = \frac{|V(lcc(G_C \times lcc(G_R)))|}{|V(lcc(G_R))|} \tag{5}$$

LCC is computed as the harmonic mean between the precision and recall:

$$LCC(G_C, G_R) = 2 \cdot \frac{LCC_P(G_C, G_R) \cdot LCC_R(G_C, G_R)}{LCC_P(G_C, G_R) + LCC_R(G_C, G_R)} \tag{6}$$

The intuition behind the LCC metric is that a narrative in a story (e.g., a news story) will be informationally coherent and will result in a connected event graph. If the main story is accompanied by additional stories (sidebar stories), it is likely that the event graphs of the sidebar stories will not be connected to the event graph of the main story (i.e., the temporal relations between the anchors of different narratives will be unidentifiable). This phenomenon is also observed in most gold graphs and automatically constructed event graphs in our corpus (cf. Section 5.3), where the ratio between the size (in the number of nodes) of the largest connected component and the whole graph is, on average, 0.83. Accordingly, we assume that the largest connected component of the event graph corresponds to the main narrative of the story. By comparing only the largest connected components, we emphasize the overlap between the main narratives and diminish the influence of the side narratives.

As an example of how event graph evaluation metrics are computed, consider the construction of an event graph from the story snippet given in Figure 2 (cf. Section 3). Let the graph given in Figure 2 be the reference graph G_R and the graph in Figure 3 be the automatically constructed graph G_C. The graph construction process omitted one reference event mention ('*competing*') and also falsely extracted one event mention ('*members*'). Figure 4 presents the tensor product $G_C \times G_R$ of graphs G_C and G_R. The product graph contains six event mentions from G_R for which the anchor and all arguments have been correctly extracted in G_C and three correctly extracted relations from G_R. The largest weakly connected component in the product graph contains the vertices '*pushed*', '*swept*', and '*rounded*'. Note that both G_C and G_R in this example are weakly connected graphs (i.e., $lcc(G_C) = G_C$ and $lcc(G_R) = G_R$), and hence the product is the same for both recall and precision components of LCC (i.e., $lcc(G_C) \times G_R = lcc(G_R) \times G_C = G_C \times G_R$). The number of vertices and edges in the three graphs and their largest connected components are as follows:

$$\begin{array}{rrr} |V(G_R)| = 7 & |V(G_C)| = 7 & |V(G_C \times G_R)| = 6 \\ |E(G_R)| = 8 & |E(G_C)| = 6 & |E(G_C \times G_R)| = 3 \\ |V(lcc(G_R))| = 7 & |V(lcc(G_C))| = 7 & |V(lcc(G_C \times G_R))| = 3 \end{array}$$

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

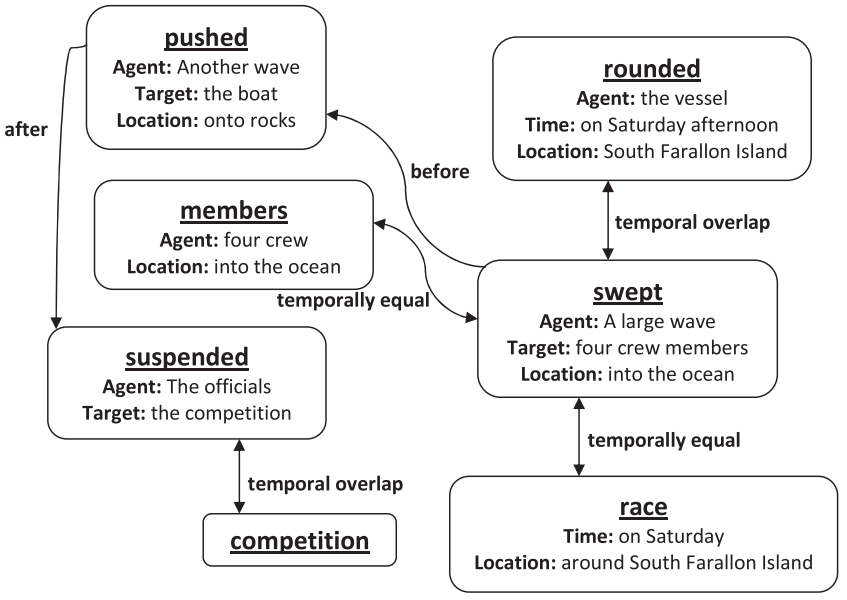


Fig. 3. Example of an automatically constructed event graph.

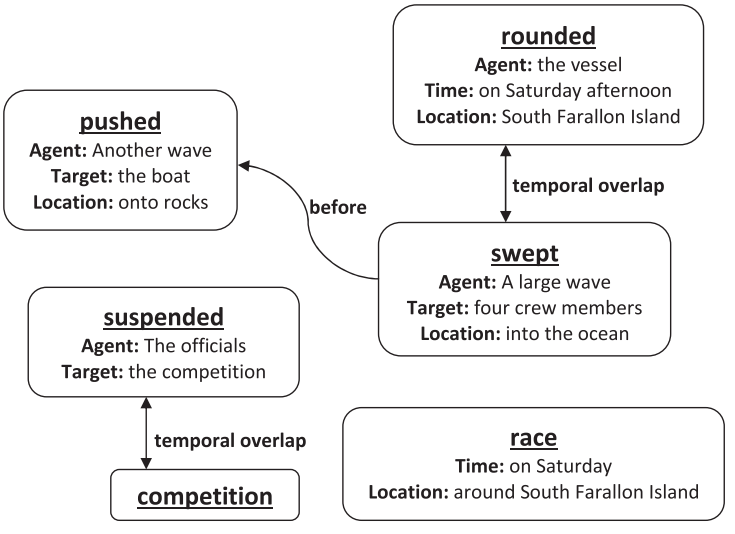


Fig. 4. Example of the tensor graph product between the reference event graph (Figure 2) and automatically constructed event graph (Figure 3).

Substituting these values into (1)–(6) gives the ROS and LCC scores:

$$ROS_P(G_C, G_R) \approx 0.69 \qquad LCC_P(G_C, G_R) \approx 0.43$$
$$ROS_R(G_C, G_R) \approx 0.60 \qquad LCC_R(G_C, G_R) \approx 0.43$$
$$ROS(G_C, G_R) \approx 0.64 \qquad LCC(G_C, G_R) \approx 0.43$$

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

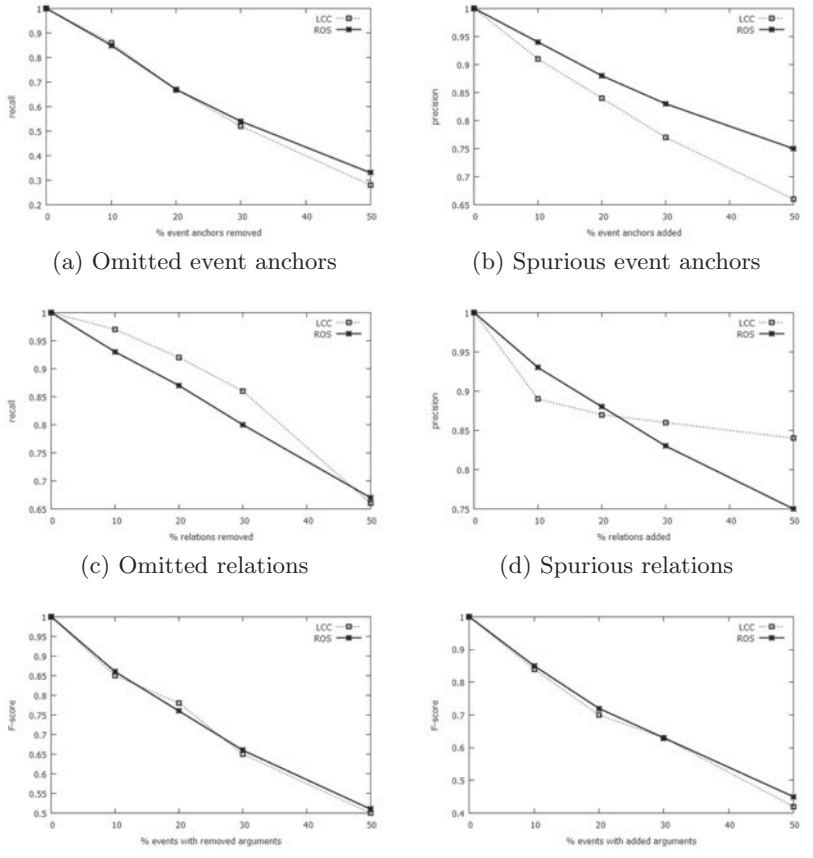
4.3 Validation of evaluation metrics

For ROS and LCC to be useful, they must have some desirable properties. We demonstrate that both ROS and LCC have properties of standard metrics. Both measures are harmonic means of components that relate to standard precision and recall metrics used in numerous NLP tasks. We demonstrate that the precision component of both ROS and LCC decreases with an increase of spurious information in automatically extracted event graphs. Analogously, the recall component decreases with an increase in missing information. We also demonstrate that decrease in performance is caused by errors in the extraction of any event graph component (anchors, arguments, and relations).

Following the idea of UzZaman and Allen (2011), we simulate the construction of an event graph G_C by randomly adding and removing information from the reference graph G_R, and then computing the ROS and LCC scores on the product graph G_P of the modified graph G_C and the original reference graph G_R. The scores are averaged over 105 reference graphs that are manually created from the set of 105 news stories (cf. Section 6.4). On average, reference graphs have 42 vertices, 62 edges, and two weakly connected components. The largest weakly connected component has an average of 39 vertices. The results are shown in Figure 5.

We first remove the vertices (with their corresponding edges) from the reference graphs to simulate the case in which some event anchors are not extracted. We observe (Figure 5(a)) that for both ROS and LCC, the recall is inversely proportional to the number of removed vertices. Next, we add vertices to the reference graphs to simulate the case in which some false event anchors are extracted. Since the average degree of vertices in reference graphs equals approximately one, for each added vertex we also randomly add one relation. We observe (Figure 5(b)) that for both ROS and LCC, the precision is inversely proportional to the number of added vertices. Similarly, we observe that precision and recall decrease with the addition (Figure 5(c)) and removal (Figure 5(d)) of relations respectively. However, the precision drop for the LCC score becomes negligible after adding a small number of relations because many of the reference graphs G_R in our experiments are close to being connected. Thus, when deriving G_C from G_R by adding a small number of spurious relations, we quickly obtain a connected graph. Consequently, adding relations does not increase the size of the largest connected component of G_C. However, in a realistic setting, extracted graphs will be more fragmented and some spurious relations may significantly increase the size of the largest connected component and thus decrease the LCC precision score. Finally, we demonstrate that the removal and addition of event arguments (Figures 5(e) and (f)) also decrease ROS and LCC scores. Given that the vertex in the tensor product G_P is created only for vertices from G_C and G_R that match in all arguments, both removal of existing arguments and addition of spurious arguments affect both precision and recall. In other words, the vertex of G_C to which a false argument has been added or from which a true argument has been removed will not match any vertices in the reference graph G_R and thus will not appear in the graph product G_P.

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press



(a) Omitted event anchors (b) Spurious event anchors

(c) Omitted relations (d) Spurious relations

(e) Omitted arguments (f) Spurious arguments

Fig. 5. Validation of ROS and LCC evaluation metrics: examination of metrics behavior when omitting true information ((a) anchors, (c) relations, and (e) arguments) or adding spurious information ((b) anchors, (d) relations, and (f) arguments).

5 EvExtra corpus

To train and evaluate the supervised extraction models, and to develop the rule-based argument extraction model, we need annotated corpora. Furthermore, to evaluate the overall quality of constructed event graphs with the proposed ROS and LCC metrics, we require a collection of manually constructed event graphs. Acknowledging the differences between our definition of factual events and TimeML's definition of events (cf. Section 3.2), we had to either annotate our own corpus or adjust some existing corpus such as TimeBank (which conforms to the TimeML standard) to fit our definition of factual events (e.g., by removing all states and non-factual event mentions). Considering that the limited size of the TimeBank corpus was identified as one of its major shortcomings (Boguraev and Ando 2005), we decided to annotate a new larger corpus of factual events. The new corpus, which we call the EvExtra corpus, contains 759 articles (330K tokens) collected from EMM

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

Table 3. *Comparison of event-annotated datasets*

Event class	#TimeBank	#EvExtra
Occurrence	4,452	12,895
Reporting	1,010	5,263
I_Action	668	4,610
Perception	51	250
State change	–	961
State	1,181	–
I_State	586	–
Aspectual	295	–
Total	8,243	23,979

NewsBrief, an online service for collecting and clustering breaking news stories.[4] All 759 articles were manually annotated for factual event mentions (anchors and arguments). Complete event graphs (event mentions, temporal relations, and event coreferences) were annotated for a subset of 105 articles. In the following text we describe the annotation process in greater detail.

5.1 Anchors annotation

Anchor annotation on the EvExtra corpus was conducted by six annotators. All annotators initially annotated a calibration set consisting of 10K tokens, after which the disagreements were analyzed and resolved by consensus. Out of 330K tokens, a common subset of 30K tokens was annotated independently by each of the six annotators. On this subset, we measured the inter-annotator agreement (IAA) as an pairwise-averaged F-score (Table 4). The IAA was computed after calibration and before resolving the disagreements. The remainder of the corpus (300K tokens) was split into six portions; each portion was independently annotated by one pair of annotators, and the third annotator resolved the disagreements. A comparison between the TimeBank corpus and the EvExtra corpus reveals that EvExtra is approximately three times larger than TimeBank (Table 3) in terms of annotated event anchors.[5]

While annotators achieved high agreement on event anchor identification, they achieved modest agreement on event type classification (except for the *Reporting* class) even after two rounds of calibration. This level of agreement indicates that determining event type is a challenging task even for humans. The agreement was particularly low for *Perception* and *State change* because annotators frequently confused these two classes with the more frequent *Occurrence* class. For instance, in '*The child fell and scratched its knee*', the anchor '*fell*' is within the *Occurrence* class, whereas in '*The stocks fell 3% in early trading*' it is within the *State change* class. Nonetheless, we believe that the resolving of disagreements substantially improves

[4] `http://emm.newsbrief.eu/NewsBrief/clusteredition/en/latest.html`
[5] The EvExtra corpus is available under the CC BY-SA-NC 3.0 license from `http://takelab.fer.hr/grapheve/`

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

Table 4. *Inter-annotator agreement for factual event anchors*

Identification	Classification				
Factual anchor	Occurrence	I_Action	Reporting	Perception	State change
87.4	65.7	59.0	86.0	45.1	42.5

the quality of annotations, which is also indicated by the performance of the supervised model (cf. Section 6.1).

The IAA we obtained for anchor identification on the EvExtra corpus (0.87 F-score) is much higher than the agreement on the TimeBank corpus (0.78 P&R avg.),[6] especially given that on TimeBank, IAA is measured as an average of precision and recall, which is a less conservative estimate than the harmonic mean computed by the F-score. Moreover, we believe that our estimate of the human agreement is more reliable as it has been averaged over 15 pairs (all combinations of two out of six annotators), whereas the TimeBank IAA is measured from a single pair of annotators. This result seems to suggest that annotating non-factual events is more demanding for humans than annotating factual events.

5.2 Arguments annotation

EvExtra corpus contains 105 articles manually annotated with event arguments to enable the development of argument extraction model. Two annotators first independently annotated 25 articles, after which the disagreements were resolved by consensus. Each annotator then separately annotated another 40 news articles. We used the resolved set of 25 documents as the development set for our rule-based argument extraction, and the remaining 80 documents as the test set. The annotators were instructed to annotate four types of event arguments: agents, targets, locations, and temporal arguments. In line with the guidelines for annotating mentions in entity coreference resolution (Pradhan *et al.* 2011), we instructed the annotators to annotate the largest possible extent for each event argument. For example, in '*The vessel Low Speed Chase, a 12 m racing sailboat with a crew of eight, competed on Saturday…*', the phrase '*The vessel Low Speed Chase, a 12 m racing sailboat with a crew of eight*' would be annotated as the agent of the mention '*competed*'. Table 5 provides the statistics of the dataset, including the average length of argument expressions, broken down by argument type. The total average length of argument expressions is three tokens.

Table 6 presents the IAA measured in terms of the F-score on the 25 documents annotated by both annotators. We measured the agreement on the argument head words (allows for extent mismatches as long as the head words match) and the strict agreement on the full argument (exact match of full argument extent). We believe that the average head word agreement of 86% is a fair agreement. As expected, mismatches in the extent annotation occur somewhat more frequently.

[6] http://timeml.org/site/timebank/documentation-1.2.html#iaa

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

Table 5. *Statistics of the argument extraction dataset*

	Dataset	Agent	Target	Time	Location
Counts	Development set	978	706	218	205
	Test set	2,063	1,589	511	551
	Total	3,041	2,295	729	756
Expression length (in num. of tokens)		2.78	3.03	4.28	2.41

Table 6. *Inter-annotator agreement for argument annotation*

	Agent	Target	Time	Location	All (micro-averaged)
Head agreement	89.2	79.4	76.0	75.8	86.2
Extent agreement	77.6	57.3	55.7	57.0	68.1

5.3 Annotation of event graphs

In order to evaluate automatically extracted event graph using the proposed ROS and LCC evaluation metrics, we first need to create a set of reference graphs. To this end, we had two annotators annotate reference graphs for 105 EvExtra news articles.[7] Both annotators independently annotated 25 documents, and then each annotator annotated a separate set of 40 documents. For final evaluation, we use reference graphs from 80 documents; graphs from the remaining 25 documents annotated by both annotators were not used for evaluation because they were already used as a development set for the argument extraction model. The manual annotation of reference graphs mimicked the stages of the construction pipeline – annotation of anchors, arguments, temporal relations, and event coreference. We measured the inter-annotator agreement on the set of 25 articles annotated by both annotators using the LCC and ROS metrics. The observed agreement is 0.47 ± 0.13 of LCC and 0.69 ± 0.08 of ROS, which we consider to be in reasonable agreement.

6 Experimental evaluation

In this section, we experimentally evaluate the graph construction pipeline. We describe the experiments for each pipeline component in detail and discuss the results obtained. We also evaluate the overall quality of the constructed event graphs using the evaluation metrics proposed in Section 4.

6.1 Anchor extraction

We evaluate the performance of our supervised models for both anchor extraction tasks: anchor identification and event type classification. We select 70% of articles from the EvExtra corpus as the training set and leave the remaining 30% for

[7] The set of 105 manually constructed event graphs is available under the CC BY-SA-NC 3.0 license from `http://takelab.fer.hr/grapheve/`

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

Table 7. *Anchor extraction and classification performance on the* EvExtra *corpus*

Anchor identification	P	R	F1			
Baseline (*memorize*)	77.6	67.2	72.0			
EvGraph-Anchors	83.4	76.8	79.9			
Type classification	**Occ.**	**IAct.**	**Rep.**	**Per.**	**SCh.**	**Macro F1**
Baseline (*memorize*)	83.5	47.4	89.5	52.3	38.9	62.3
EvGraph-Anchors	84.0	62.4	90.6	83.5	55.1	75.2

Occ.: Occurrence; IAct.: I_Action; Per.: Perception; Rep.: Reporting; SCh.: State change.

testing. Here and henceforth (for all supervised models), we use grid search and 10-fold cross-validation to optimize the model hyperparameters and report the performance of the optimal model on the test set. All results were obtained with the logistic regression model, which we found to slightly outperform the linear SVM in all experiments. As a competitive baseline, we use the *memorize* model proposed by Bethard (2008). The baseline model predicts for each word the most frequent class assigned to that word on the training set. If the word was never observed in the training set, it is assigned the most frequent class in the training set: class *Not event* and class *Occurrence* for the anchor identification and event type classification task respectively.

Table 7 shows the performance of anchor extraction on the EvExtra corpus. The upper part of the table shows the performance of the model (EvGraph-Anchors) on the anchor identification task. Our model substantially outperforms the *memorize* baseline. The overall anchor identification performance is close to the IAA scores, suggesting that the performance cannot be substantially improved with the current annotation quality.

The lower part of Table 7 shows the performance on the event type classification task. The logistic regression model again outperforms the competitive baseline, this time by a wider margin. The performance gain is most discernible for the classes with the least number of instances in the training set – *Perception* and *State change*. We obtain the best results for the *Reporting* class, which is expected because reporting events exhibit little lexical variation in news stories (i.e., only a few verbs such as '*say*', '*report*', '*tell*', and '*add*' account for the majority of reporting events). The performance of the model is above the IAA scores for all classes, indicating that resolving annotator disagreements (which took place after measuring the IAA score and before model training) considerably improved the corpus quality. This improvement is especially notable for small classes (such as the *Perception* class), where the adjudication of only a handful of examples leads to large improvements in the annotation consistency and, consequently, improvements in the classifier performance.

In contrast to previous approaches based on the TimeBank corpus, we do not extract non-factual event mentions or mentions describing states. However, for the sake of comparison with the state-of-the-art models, we evaluate our model on the

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

Table 8. *Performance of the anchor identification and classification models on the TimeBank corpus Occ.: Occurrence; ISt.: I_State; St: State; Rep.: Reporting; Per.: Perception; IAc.: I_Action; As.: Aspectual*

Anchor identification	P	R	F1					
Llorens *et al.* (2010)	81.0	**86.0**	**83.0**					
Grover *et al.* (2010)	75.0	85.0	80.0					
EvGraph-Anchors-TimeBank	**87.0**	75.1	80.6					
EvGraph-Anchors	84.2	61.0	70.8					
Type classification	**Occ.**	**ISt.**	**St.**	**Rep.**	**Per.**	**IAc.**	**As.**	**F1**
Llorens *et al.* (2010)	–	–	–	–	–	–	–	**79.0**
UzZaman and Allen (2010)	–	–	–	–	–	–	–	77.0
EvGraph-Anchors-TimeBank	84.7	62.0	50.2	91.4	72.0	44.1	68.5	78.3

TimeBank corpus. We perform two experiments: in the first experiment, we both train and test our model on the TimeBank corpus; we refer to this model as the EvGraph-Anchors-TimeBank model. In the second experiment, we train the model on the EvExtra corpus but test it on the TimeBank corpus (EvGraph-Anchors model). The results are shown in Table 8, along with the state-of-the-art results from TempEval-2 (Verhagen *et al.* 2010). Our anchor identification and classification models achieve a performance comparable with state of the art, even though the features that aim to distinguish between factual and non-factual events have little impact when used on the TimeBank corpus. On anchor identification task, the EvGraph-Anchors-TimeBank model has a significantly better precision than the best performing model at TempEval-2 (Llorens *et al.* 2010). However, our model has a lower recall, presumably because it lacks the features specifically designed to recognize non-factual events and states. The EvGraph-Anchors model (trained on the EvExtra corpus) also has good precision (as all factual events are events) but significantly lower recall (as the model fails to recognize states and non-factual event anchors because such anchors were not present in the training set).

Given that the EvExtra corpus is approximately three times larger than TimeBank, it is interesting to see how the size of the training set affects the performance on both tasks. To this end, we trained models on 25%, 50%, 75%, and 100% of the training set. Table 9 presents the performance of the four models on the test set. The performance improved considerably when the size of the training set was doubled from 25% to 50% of the full size. However, with a further increase in the training set size, the performance gains diminish. Thus, we therefore doubt that substantial performance gains can be obtained from annotating more data; improving the annotation quality would result in larger gains.

A closer inspection of the anchor identification misclassifications reveals that most errors relate to one of the following causes:

(1) Noun event anchors that are most often mentioned generically (e.g., *violence*, *war*) are not extracted even when referring to concrete events (e.g., '*at least*

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

Table 9. *Influence of the training set size on performance*

	Identification	Classification					
Train set size	Factual anchor	Occ.	I_Act.	Rep.	Per.	St. ch.	Macro F1
25%	75.0	78.5	54.3	87.8	76.3	39.2	67.2
50%	77.8	82.7	60.2	89.1	75.9	51.0	71.8
75%	79.2	83.6	61.8	90.5	79.5	55.0	74.1
100%	79.9	84.0	62.4	90.6	83.5	55.1	75.2

41 people, mostly civilians, have been killed in violence that prompted UN…'). Such cases result in false negatives.

(2) Verbs in the infinitive form may refer to both something that actually occurred (e.g., '*Violence forced the Arab League to end its own Syrian monitoring mission…*') and something that did not occur in the real world (e.g., '*She underlined that the United States want to see continuing progress*'). The classifier is often unable to distinguish between the two cases, producing both false positives and false negatives.

(3) Non-factual mentions occasionally have distant and implicit indicators of factuality (e.g., '*That raises the possibility that its fighters are infiltrating across the border*'). In such cases the model does not recognize that the anchors are not factual, which results in false positives.

6.2 Argument extraction

For the argument extraction task, we compared our rule-based system with the baseline method that extracts all nominal subjects of an event anchor as agents, all direct objects as targets, all locational named entities in the vicinity of the anchor as locative arguments, and all temporal expressions in the vicinity of the anchors as temporal arguments. In Table 10, we present the performance of the argument extraction method (EvGraph-Args) and the baseline on the test set of 80 manually annotated documents, broken down by argument type. The results are given separately for the argument heads (columns 3–5) and argument extent recognition (columns 6–8). In the latter case, only strict matches of the full argument extent are considered to be correct.

The results indicate that our extraction rules are suitable for the recognition of argument heads, whereas the chunk-based detection of argument extent is moderately successful in identifying the full extent of arguments. This moderate performance is mostly due to the fact that arguments often extends beyond the boundaries of the chunk containing its head word, as extracted by the employed OpenNLP chunker. For example, in '*The vessel Low Speed Chase, a 12 m racing sailboat with a crew of eight, competed on Saturday…*', the chunk of the argument head '*Chase*' is '*The vessel Low Speed Chase*', whereas the full extent of the argument is '*The vessel Low Speed Chase, a 12 m racing sailboat with a crew of eight*'. We could have resolved this mismatch by instructing the annotators to annotate argument

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

Table 10. *Argument extraction performance*

		Head extraction			Extent recognition		
	Argument type	P	R	F1	P	R	F1
Baseline	Agent	93.1	58.5	71.8	72.8	45.7	56.2
	Target	88.2	55.5	68.1	63.6	40.0	49.1
	Time	78.1	23.6	36.2	50.0	15.2	23.3
	Location	79.1	16.7	27.6	62.8	13.4	22.0
	Macro avg.	84.6	38.6	53.0	62.3	28.6	38.8
EvGraph-Args	Agent	85.7	79.8	82.6	63.7	59.4	61.5
	Target	81.3	74.6	77.8	58.3	54.0	56.1
	Time	80.0	64.2	71.2	50.0	43.5	46.5
	Location	66.3	56.6	61.1	34.8	34.1	34.5
	Macro avg.	78.3	68.8	73.3	51.7	47.8	49.6

extents as chunks provided by the employed chunker. However, this step would introduce a bias toward using chunking to identify argument extent, as well as a bias toward the specific chunker used. The fact that we use strict matching also lowers the results.

Overall, the EvGraph-Args model outperforms the baseline by a wide margin, justifying the use of a rich set of syntactic patterns and argument type disambiguation rules. The baseline model is better in terms of precision but is much worse in terms of recall and, consequently, F-score. This is expected considering that we optimized the argument extraction rules for the F-score. Performance on the locative arguments is substantially worse than on temporal arguments because while most temporal arguments are temporal expressions, recognized as such by the temporal expression tagger, the majority of locative arguments are not locational named entities. Thus, to identify locative arguments in such cases, we rely on measuring the WordNet-based semantic similarity with the locative concepts (cf. Section 3.3.2), which does not appear to be particularly robust.

As our argument extraction approach is primarily based on syntactic patterns, most extraction errors are caused by parsing errors. A single incorrect sentence parse may trigger several argument extraction errors.

6.3 Relation extraction

In this section, we describe the experiments on the extraction of temporal relations and within-document event coreference relations, i.e., relations forming the edges of an event graph.

6.3.1 Temporal relation extraction

We conducted all experiments on the combined TimeBank/AQUAINT corpus used for TempEval-2 (Verhagen *et al.* 2010). The dataset consists of 256 documents and

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

Table 11. *Results for temporal relation identification and classification*

Relation identification	P	R	F1		
Adjacency baseline	27.5	58.9	37.5		
EvGraph-TRels-Ident	79.3	71.3	75.1		
Relation classification	Equals	Overlap	Before	After	Macro F1
Majority class baseline	–	–	51.6	–	12.9
Tense-based baseline	11.4	46.2	28.4	7.9	23.5
EvGraph-TRels-Class	67.5	47.0	49.8	57.7	55.5

3,721 temporal relations annotated between events, 3,190 of which are between events from the same or adjacent sentences. We selected 180 documents for training and the remaining 76 documents for testing. To avoid including *post hoc* temporal knowledge in training, we ensured that all documents from the training set were created before the documents from the test set. Furthermore, some of the documents from the AQUAINT portion of the dataset contain passages that are verbatim (or almost verbatim) copies of passages from earlier created documents. To avoid having the same passages in both training and test sets, we eliminated all duplicate passages from the AQUAINT portion of the corpus.

In the TimeBank/AQUAINT corpus only the temporal relations inferable from text were annotated. Since we perform temporal relation extraction in two steps (relation identification followed by relation classification), we first needed to generate negative training instances for the relation identification task. We used all pairs of events from adjacent sentences for which no temporal relation was annotated in the dataset as negative instances.

The first part of Table 11 presents the performance of the model (EvGraph-TRels-Ident) on the temporal relation identification task. As the baseline, we use a model that connects adjacent event mentions (i.e., predicts that temporal relations are identifiable for all pairs of adjacent event mentions). Our model outperforms the baseline by a very wide margin, with balanced precision and recall.

In the second part of Table 11 we present the performance of the temporal relation classification model (EvGraph-TRels-Class). Here we use two baselines: a majority class baseline and a simple tense-based baseline. The majority class baseline predicts the most frequent class on the training set (*Before* class). The tense-based baseline predicts the temporal relation based on the tense and relative position of event anchors as follows: (1) An event anchor in the past tense is considered to occur *before* an event anchor in the present tense; (2) two event anchors in the same tense are considered to be temporally *equal* if they occur in the same sentence; (3) in all other cases, anchors are considered to be in temporal *overlap*. Our model outperforms both baselines by a very wide margin. The model performs worst on the *Overlap* class. This performance can be attributed to three possible causes: (1) Since most misclassifications occur between *Overlap* and *Before* instances as well as *Overlap* and *After* instances, we hypothesize that temporal overlap is less discernible from a text than other classes of temporal relations; (2) *Overlap* instances occur

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

Table 12. *Temporal relation classification in the TempEval-2 setting*

Model	Overlap	Before	After	F1
UzZaman and Allen (2010) (TempEval-2 Task E)	–	–	–	58.0
Llorens *et al.* (2010) (TempEval-2 Task E)	–	–	–	55.0
UzZaman and Allen (2010) (TempEval-2 Task F)	–	–	–	60.0
Llorens *et al.* (2010) (TempEval-2 Task F)	–	–	–	60.0
EvGraph-TRels-TimeBank	78.9	45.1	56.6	**60.2**

rarely in the training set; and (3) *Overlap* class as defined here includes several of Allen's (1983) basic temporal relations (X *overlaps* Y, X *starts* Y, X *finishes* Y, and X *during* Y), and these basic relations may manifest themselves differently in text. While the first two issues could be remedied by compiling a balanced training set with the most discernible examples for the *Overlap* class, the third issue would require a more detailed investigation, which we leave for the future work.

Our results cannot be directly compared with the results from TempEval-2 (Verhagen *et al.* 2010) because we use a different set of labels (adding the *Equal* class) and consider different pairs of events (all pairs of events from adjacent sentences). To enable a direct comparison with the state-of-the-art approaches for the extraction of temporal relations, we merge temporal equality and temporal overlap classes and consider only pairs of events where (1) both event mentions represent main events from two consecutive sentences (as in TempEval-2 Task E), or (2) one event mention syntactically dominates the other (as in TempEval-2 Task F). The performance of our model in this setting (EvGraph-TRels-TimeBank), compared with the best performing models from TempEval-2 (Verhagen *et al.* 2010), is shown in Table 12. The results confirm that the EvGraph-TRels-TimeBank model performs comparably to the state of the art.

6.3.2 *Event coreference*

We perform the event coreference resolution experiments on the EventCorefBank dataset from Bejan and Harabagiu (2008). EventCorefBank is annotated for both within- and cross-document event coreference. Since we build event graphs from the individual documents, here we only consider within-document event coreference resolution. We split the dataset into training and testing set (80%–20% split ratio). We employ the commonly used event coreference baseline (Ahn 2006; Bejan and Harabagiu 2010), which declares that two event mentions corefer if their anchor lemmas match or if one lemma is the nominalization of the other. This baseline is simple but competitive.

Table 13 presents the performance of our supervised model (EvGraph-Coref). Our model outperforms both baseline and unsupervised non-parametric Bayesian model (Bejan and Harabagiu 2010) by a wide margin. Although direct comparison with other supervised event coreference resolution systems is not feasible due to differences in the datasets, the reported results (Chen *et al.* 2011; Lee *et al.* 2012)

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

Table 13. *Within-document event coreference resolution performance*

Model	P	R	F1
Lemma-matching baseline	93.1	28.3	43.4
Unsupervised model (Bejan and Harabagiu 2010)	83.0	34.4	48.6
EvGraph-Coref	81.6	41.8	55.3

suggest that our mention comparison features achieve performance comparable with the state of the art. For all models, the recall is much lower than the precision. Closer inspection of the false negatives reveals two main causes for this result. First, in many cases the arguments of at least one of the mentions are implicit. For example, in '*Jackson was arrested on DUI charge early yesterday… General Manager A.J. Smith issued the following statement on the arrest…*' the arguments of the second mention (*arrest*') are implicit. In such cases the usefulness of features measuring the similarity between event arguments is diminished. Second, in some cases the entity coreference resolution from Lee *et al.* (2011) fails to recognize matches between mention arguments. For example, in '*Matt Smith has been cast as the next incarnation of the Doctor. Even the skeptics agreed that the right guy was chosen…*', entities '*Matt Smith*' and '*the right guy*' are not resolved as coreferent. Consequently, the match between the targets of mentions '*cast*' and '*chosen*' is not recognized.

6.4 Event graph evaluation

Finally, we evaluate the overall quality of the automatically extracted event graphs using the evaluation metrics proposed in Section 4.

6.4.1 Results

To compare our three-stage pipeline against a reasonable baseline, we use as a baseline system a combination of baselines from each pipeline stage. More precisely, the baseline pipeline consists of (1) the competitive *memorize* baseline for anchor extraction (cf. Section 6.1), (2) the rule-based baseline for argument extraction (cf. Section 6.2), (3) the tense-based baseline for temporal relations (cf. Section 6.3.1), and (4) the lemma-matching baseline for event coreference resolution (cf. Section 6.3.2). Table 14 presents the overall performance of the graph construction pipeline (EvGraph) along with the performance of the baseline system. We report the mean and standard deviation of ROS and LCC over the set of 80 documents. The results in Table 14 were obtained without applying transitive closure on the event graph. Applying transitive closure yielded consistently poorer results than when no transitive closure was applied. This result is not surprising because the temporal classification performance – although comparable to the state of the art – is only at about 60% of F1-score, and because transitive closure may induce several spurious relations due to a single incorrectly classified relation.

The full graph construction pipeline significantly outperforms the baseline. The differences are significant at $p < 0.05$ for LCC and $p < 0.01$ for ROS (Student's

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

Table 14. *Overall quality of the automatically constructed event graphs*

Model	LCC	ROS
Baseline	0.175 ± 0.103	0.184 ± 0.091
EvGraph	0.225 ± 0.098	0.287 ± 0.063

Table 15. *Results of the headroom analysis (gold standard annotations for one of the pipeline steps)*

Model	LCC	ROS
Gold anchors	0.269 ± 0.127	0.371 ± 0.084
Gold arguments	0.245 ± 0.095	0.345 ± 0.082
Gold temporal relations	**0.477 ± 0.170**	**0.437 ± 0.102**
Gold Coreference	0.249 ± 0.106	0.361 ± 0.074

two-tailed t-test). The obtained results indicate that the LCC metric is stricter than the ROS metric.

6.4.2 Headroom analysis

Our graph construction pipeline consists of multiple interdependent components. Thus, it is interesting to see how the performance of the individual components affects the performance of the entire system. To this end, we perform a headroom analysis of each component, which demonstrates the potential impact on overall system performance if the component in question would exhibit perfect performance. Although the so-obtained improvement estimates are overly optimistic, they are useful for comparing the potential benefits of enhancing different components in a resource-bounded setting. We perform the headroom analysis for the following components: (1) anchor extraction, (2) argument extraction, (3) temporal relation extraction, and (4) coreference resolution. In headroom analysis, the construction of an event graph for the component under test uses the gold standard annotations for that component and the automated output for all other components. For example, to construct an event graph for headroom analysis of the temporal relation extraction component, we automatically extract the anchors and arguments, copy temporal relations from the reference graph, and then automatically resolve event mention coreference.

In Table 15 we present the results of the headroom analysis. The results demonstrate that perfect temporal relation extraction would yield the largest improvements with respect to both evaluation metrics. At first, this result appears to contradict our earlier results, which indicated that in absolute terms, coreference resolution performs worse than temporal relation extraction (cf. Section 6.3). However, the number of temporal relations in the reference graphs is much larger than the number of coreferent event mention pairs, and thus the performance on temporal relation extraction has a greater effect on the overall event graph quality. Furthermore, the performance improvement is larger for the LCC metric (0.477 versus 0.225) than for the ROS

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

metric (0.437 versus 0.287): A single correctly extracted relation can considerably increase the size of the largest connected component, and thus the LCC score.

7 Conclusions and perspectives

Many NLP and information retrieval applications would benefit from structured event-oriented representations of a text due to the large number of both event-oriented information needs and event-oriented information sources (e.g., breaking news). Moreover, there are information needs that explicitly involve events and temporal structure (e.g., retrieving all documents in which one particular event precedes the other). In this paper we introduced *event graphs* as a novel way of structuring event-based information from text, bridging the gap that currently exists between sentence- and document-level events. In event graphs, individual event mentions (including event protagonists and circumstances) are connected through temporal relations and coreference links, providing a comprehensive representation of event-oriented information from a document. While most of the existing event extraction methods focus on isolated tasks (e.g., the extraction of event mentions or temporal relation extraction between gold-annotated event mentions), we developed and evaluated a complete end-to-end system for the construction of event graphs from text.[8]

We have described and intrinsically evaluated each component constituting the graph construction pipeline. The experimental results show that each of the individual graph construction components performs at a level comparable with the state of the art for the respective task. To measure the overall quality of the constructed event graphs, we proposed two metrics based on the graph product between the automatically constructed event graphs and their respective reference graphs created by human annotators. The headroom analysis indicates that improving temporal relation extraction would have the greatest effect on the overall event graph quality.

Finally, we have presented the EvExtra corpus, a large newspaper corpus annotated with factual event mentions and event graphs. We hope the EvExtra corpus will stimulate further research in this area.

This work lays the foundation for structured event-based document analysis. We anticipate four main directions for the future work. The first direction is to improve and extend the event graph construction pipeline, both technically (e.g., supervised argument extraction, exploration of other parameter settings) and conceptually. There are concepts of relatedness and interesting relations between event mentions other than temporal relations and coreference, such as causality and subordination, which could also be captured with event graphs.

The second research direction involves the adaptation of the pipeline to languages other than English. This adaptation would primarily require the annotation of a new corpus for the training of supervised extraction models (anchor extraction, temporal relation extraction, event coreference resolution) and the adaption of the syntax-based argument extraction rules. In addition, the following tools and resources would be required: WordNet (or a similar lexico-semantic resource), preprocessing tools (a sentence splitter, a tokenizer, a lemmatizer, and a stemmer), a POS tagger, a

[8] The system is freely available for research purposes from `http://takelab.fer.hr/grapheve/`

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

dependency parser, a named entity recognizer, and an entity coreference resolution tool. Our immediate focus will be on adapting the pipeline for the Croatian language.

The third direction for the future work is the application of event graphs in various NLP and information retrieval tasks. Graph-based representation of event-oriented documents opens up the possibility of formulating many event-based tasks in graph-theoretic terms and employing a variety of graph-based algorithms and tools. For instance, by locating overlaps (i.e., common subgraphs) between the event graphs of different documents we can measure the event-based similarity of documents. Other applications include, but are not limited to, temporal question answering, structured event-oriented information retrieval, new event detection in TDT, news summarization, and news simplification. We believe that the required quality of event graphs varies across different applications, and that the applications themselves will establish the reference values for graph evaluation metrics.

Finally, the fourth research direction would provide a more exhaustive and rigorous treatment of the representational and inferential aspects of event graphs. Event graphs already allow for a structured representation of event-oriented information, but the individual events are of different levels of abstraction and granularity. Furthermore, event graph is a rather static structure; being able to derive new event-graph representations by transforming an event graph or combining event graphs from different documents appears to be a logical next step. We envisage a formal framework, based on event graphs, that would enable the modeling of events in an event space that spans from linguistic events at the lowest level to topics at the highest level. This modeling could be accomplished by a comprehensive set of graph-based operators that transform and combine within- and cross-document event representations into representations at specified levels of abstraction and granularity, effectively inducing a hierarchy of events. Such an event graph-based framework would enable a uniform and elegant treatment of both events and topics for the purpose of event-based document analysis.

Acknowledgements

This work has been supported by the Ministry of Science, Education and Sports, Republic of Croatia under the Grant 036-1300646-1986. We thank the reviewers for their insightful comments and suggestions.

References

ACE. 2005. *Evaluation of the Detection and Recognition of ACE: Entities, Values, Temporal Expressions, Relations, and Events*. Gaithersburg, MD: NIST.

Agirre, E., and Soroa, A. 2009. Personalizing PageRank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL '09)*. Athens, Greece. Stroudsburg, PA: ACL, pp. 33–41.

Ahn, D. 2006. The stages of event extraction. In *Proceedings of COLING/ACL 2006 Workshop on Annotating and Reasoning about Time and Events*, Sydney, Australia. Stroudsburg, PA: ACL, pp. 1–8.

Ahn, D., Schockaert, S., De Cock, M., and Kerre, E. 2006. Supporting temporal question answering: strategies for offline data collection. In *Proceedings of the 5th International Workshop on Inference in Computational Semantics*. Buxton, UK: ACL, pp. 127–32.

Allan, J. 2002. *Topic Detection and Tracking: Event-Based Information Organization*, vol. 12. Dordrecht, Netherlands: Kluwer.

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

Allen, J. R. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM* **26**(11): 832–43.

Aone, C., and Ramos-Santacruz, M. 2000. REES: a large-scale relation and event extraction system. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*. Seattle, WA. Stroudsburg, PA: ACL, pp. 76–83.

Bagga, A., and Baldwin, B. 1999. Cross-document event coreference: annotations, experiments, and observations. In *Proceedings of the Workshop on Coreference and its Applications*. Stroudsburg, Pennsylvania. Stroudsburg, PA: ACL, pp. 1–8.

Baker, C. F., Fillmore, C. J., and Lowe, J. B. 1998. The Berkeley framenet project. In *Proceedings of the 17th International Conference on Computational linguistics (COLING '98)*, Montreal, Canada. Stroudsburg, PA: ACL, pp. 86–90.

Bejan, C., and Harabagiu, S. 2008. A linguistic resource for discovering event structures and resolving event coreference. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*, Marrakech, Morocco. Paris, France ELRA.

Bejan, C., and Harabagiu, S. 2010. Unsupervised event coreference resolution with rich linguistic features. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, Uppsala, Sweden. Stroudsburg, PA: ACL, pp. 1412–22.

Bejan, C. A., and Hathaway, C. 2007. UTD-SRL: a pipeline architecture for extracting frame semantic structures. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval 2007)*, Prague, Czech Republic. Stroudsburg, PA: ACL, pp. 460–63.

Bethard, S. 2008. *Finding Event, Temporal and Causal Structure in Text: A Machine Learning Approach*. PhD thesis. University of Colorado at Boulder, USA.

Bethard, S. 2013. ClearTK-TimeML: a minimalist approach to TempEval 2013. *Second Joint Conference on Lexical and Computational Semantics (*SEM)*, vol. 2, Atlanta, GA. Stroudsburg, PA: ACL, pp. 10–14.

Boguraev, B., and Ando, R. K. 2005. TimeBank-driven TimeML analysis. In *Annotating, Extracting and Reasoning about Time and Events, Dagstuhl Seminar Proceedings*, Dagstuhl, Germany.

Borgwardt, K. M. 2007. *Graph Kernels*. PhD thesis, Ludwig-Maximilians-Universität München, Munich, Germany.

Bramsen, P., Deshpande, P., Lee, Y. K., and Barzilay, R. 2006. Inducing temporal graphs. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP '06)*, Sydney, Australia. Stroudsburg, PA: ACL, pp. 189–98.

Chambers, N., and Jurafsky, D. 2008. Unsupervised learning of narrative event chains. In *Proceedings of 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT '08)*, pp. 789–97.

Chambers, N., and Jurafsky, D. 2009. Unsupervised learning of narrative schemas and their participants. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP* vol. 2, Singapore. Stroudsburg, PA: ACL, pp. 602–10.

Chang, C. C., and Lin, C. J. 2011. LibSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* **2**(27): 1–27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Chang, A. X., and Manning, C. D. 2012. *SUTIME: A Library for Recognizing and Normalizing Time Expressions*. Istanbul, Turkey: ELRA.

Chen, B., Su, J., Pan, S., and Tan, C. 2011. A unified event coreference resolution by integrating multiple resolvers. In *Proceedings of 5th International Joint Conference on Natural Language Processing (IJCNLP 2011)*, Chiang Mai, Thailand. New York, NY: Curran Associates.

Cordella, L. P., Foggia, P., Sansone, C., and Vento, M. 2004. A subgraph isomorphism algorithm for matching large graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**(10): 1367–72.

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

Davidson, D. 1967. The logical form of action sentences. *Essays on Actions and Events* **5**: 105–48.

De Marneffe, M. C., MacCartney, B., and Manning, C. D. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of 5th International Conference on Language Resources and Evaluation (LREC 2006)*, Genoa, Italy, vol. 6. Paris, France: ELRA, pp. 449–54.

Derczynski, L., and Gaizauskas, R. 2013. Temporal signals help label temporal relations. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics,* Sofia, Bulgaria, vol. 78. Stroudsburg, PA: ACL.

Fan, R. E., Chang, K., Hsieh, C. J., Wang, X. R., and Lin, C. J. 2008. LibLinear: a library for large linear classification. *Journal of Machine Learning Research* **9**: 1871–4.

Fillmore, C. J. 1976. Frame semantics and the nature of language*. *Annals of the New York Academy of Sciences* **280**(1): 20–32.

Finkel, J. R., Grenager, T., and Manning, C. D. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL '05)*, Ann Arbor, MI. Stroudsburg, PA: ACL, pp. 363–70.

Gärtner, T., Flach, P., and Wrobel, S. 2003. On graph kernels: hardness results and efficient alternatives. In B. Schölkopf and M. K. Warmuth (eds.), *Learning Theory and Kernel Machines*. Berlin, Germany: Springer-Verlag, pp. 129–43.

Gildea, D., and Jurafsky, D. 2002. Automatic labeling of semantic roles. *Computational Linguistics* **28**(3): 245–88.

Glavaš, G., and Šnajder, J. 2013. Recognizing identical events with graph kernels. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, Sofia, Bulgaria (accepted). Stroudsburg, PA: ACL.

Grishman, R., and Sundheim, B. 1996. Message understanding conference-6: a brief history. In *Proceedings of International Conference on Computational Linguistics (COLING 1996)*, Copenhagen, Denmark, vol. 96. Dresden, Germany: ICCL, pp. 466–71.

Grover, C., Tobin, R., Alex, B., and Byrne, K. 2010. Edinburgh-LTG: TempEval-2 system description. In *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval 2010)*, Uppsala, Sweden. Stroudsburg, PA: ACL, pp. 333–6.

Haghighi, A., Toutanova, K., and Manning, C. D. 2005. A joint model for semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, Ann Arbor, MI. Stroudsburg, PA: ACL, pp. 173–6.

Hammack, R., Imrich, W., and Klavžar, S. 2011. *Handbook of Product Graphs. Discrete Mathematics and Its Applications*. Boca Raton, FL: CRC Press.

Hatzivassiloglou, V., Gravano, L., and Maganti, A. 2000. An investigation of linguistic features and clustering algorithms for topical document clustering. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY: ACM, pp. 224–31.

Humphreys, K., Gaizauskas, R., and Azzam, S. 1997. Event coreference for information extraction. In *Proceedings of a Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*, Madrid, Spain. Stroudsburg, PA: ACL, pp. 75–81.

Humphreys, K., Gaizauskas, R., Azzam, S., Huyck, C., Mitchell, B., Cunningham, H., and Wilks, Y. 1998. University of Sheffield: description of the LaSIE-II system as used for MUC-7. In *Proceedings of the Seventh Message Understanding Conferences (MUC-7)*, San Diego, CA. Gaithersburg, MD: NIST.

Jans, B., Bethard, S., Vulić, I., and Moens, M. F. 2012. Skip n-grams and ranking functions for predicting script events. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, Avignon, France. Stroudsburg, PA: ACL, pp. 336–44.

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

Jun, P., and Min, Y. 2012. Improved temporal relation classication using dependency parses and selective crowdsourced annotations. In *Proceedings of the International Conference on Computational Linguistics (COLING 2012)*, Mumbai, India, pp. 2109–24.

Karttunen, L., and Zaenen, A. 2005. Veridicity. *Annotating, Extracting and Reasoning about Time and Events, Dagstuhl Seminar Proceedings*. Dagstuhl, Germany: IBFI.

Kingsbury, P., and Palmer, M. 2002. From Treebank to PropBank. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC '02)*, Las Palmas, Spain. Paris, France: ELRA, pp. 1989–93.

Kolomiyets, O., Bethard, S., and Moens, M. 2012. Extracting narrative timelines as temporal dependency structures. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012)*, Jeju Island, Korea. Stroudsburg, PA: ACL, pp. 88–97.

Kumaran, G., and Allan, J. 2004. Text classification and named entities for new event detection. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY: ACM, pp. 297–304.

Lavie, A., and Denkowski, M. J. 2009. The METEOR metric for automatic evaluation of machine translation. *Machine Translation* **23**(2–3): 105–15.

Lee, H., Peirsman, Y., Chang, A., Chambers, N., Surdeanu, M., and Jurafsky, D. 2011. Stanford's multi-pass sieve coreference resolution system at the CoNLL-2011 shared task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task (CoNLL 2011)*, Portland, OR. Stroudsburg, PA: ACL, pp. 28–34.

Lee, H., Recasens, M., Chang, A., Surdeanu, M., and Jurafsky, D. 2012. Joint entity and event coreference resolution across documents. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP 2012)*, Jeju Island, Korea, Stroudsburg, PA: ACL, pp. 489–500.

Llorens, H., Saquete, E., and Navarro, B. 2010. TIPSem (English and Spanish): evaluating CRFs and semantic roles in TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, Uppsala, Sweden. Stroudsburg, PA: ACL, pp. 284–91.

Llorens, H., Saquete, E., and Navarro-Colorado, B. 2013. Applying semantic knowledge to the automatic processing of temporal expressions and events in natural language. *Information Processing & Management* **49**(1): 179–97.

Makkonen, J. 2003. Investigations on event evolution in TDT. In *Proceedings of the Student Research Workshop at Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL-HLT '03)*, Edmonton, Canada. Stroudsburg, PA: ACL, pp. 43–48.

Makkonen, J., Ahonen-Myka, H., and Salmenkivi, M. 2004. Simple semantics in topic detection and tracking. *Information Retrieval* **7**(3): 347–68.

McCallum, A., and Li, W. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, Sapporo, Japan, Stroudsburg, PA: ACL, pp. 188–91.

Melli, G., Wang, Y., Liu, Y., Kashani, M. M., Shi, Z., Gu, B., Sarkar, A., and Popowich, F. 2006. Description of SQUASH, the SFU question answering summary handler for the DUC-2005 summarization task. In *Proceedings of the Document Understanding Conference*, New YorkNY. Gaithersburg, MD: NIST.

Menchetti, S., Costa, F., and Frasconi, P. 2005. Weighted decomposition kernels. In *Proceedings of the 22nd International Conference on Machine Learning (ICML '05)*, Bonn, Germany. New York, NY: ACM, pp. 585–92.

Moreda, P., Llorens, H., Saquete, E., and Palomar, M. 2011. Combining semantic information in question answering systems. *Information Processing & Management* **47**(6): 870–85.

Nadeau, D., and Sekine, S. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes* **30**(1): 3–26.

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

Nallapati, R., Feng, A., Peng, F., and Allan, J. 2004. Event threading within news topics. In *Proceedings of the 13th ACM International Conference on Information and Knowledge Management*. New York, NY: ACM, pp. 446–53.

Navigli, R., and Lapata, M. 2007. Graph connectivity measures for unsupervised word sense disambiguation. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, Hyderabad, India. Burlington, MA: Morgan Kaufmann, pp. 1683–8.

Palmer, M., Gildea, D., and Kingsbury, P. 2005. The proposition bank: an annotated corpus of semantic roles. *Computational Linguistics* **31**(1): 71–106.

Polanyi, L., and Zaenen, A. 2006. Contextual valence shifters. *Computing Attitude and Affect in Text: Theory and Applications* **20**: 1–10.

Pradhan, S., Ramshaw, L., Marcus, M., Palmer, M., Weischedel, R., and Xue, N. 2011. CoNLL-2011 shared task: modeling unrestricted coreference in OntoNotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task (CoNLL 2011)*, Portland, OR. Stroudsburg, PA: ACL, pp. 1–27.

Pustejovsky, J. 1991. The syntax of event structure. *Cognition* **41**(1): 47–81.

Pustejovsky, J., Castano, J., Ingria, R., Sauri, R., Gaizauskas, R., Setzer, A., Katz, G., and Radev, D. 2003b. TimeML: robust specification of event and temporal expressions in text. *New Directions in Question Answering* **2003**: 28–34.

Pustejovsky, J., Hanks, P., Sauri, R., See, A., Gaizauskas, R., Setzer, A., Radev, D., Sundheim, B., Day, D., and Ferro, L. 2003a. The TimeBank corpus. In *Proceedings of Corpus Linguistics*, vol. 2003. Lancaster, UK: UCREL, pp. 40.

Quine, W. 1985. Events and reification. In E. LePore and B. P. McLaughlin (eds.), *Actions and Events: Perspectives on the Philosophy of Donald Davidson*, Oxford UK: Blackwell, pp. 162–71.

Resnik, P. 1999. Semantic similarity in a taxonomy: an information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research* **11**: 95–130.

Rosen, S. 1999. The syntactic representation of linguistic events. *Glot International* **4**(2): 3–11.

Salton, G., Wong, A., and Yang, C. 1975. A vector space model for automatic indexing. *Communications of the ACM* **18**(11): 613–20.

Saquete, E., González, J. L. V., Martínez-Barco, P., Munoz, R., and Llorens, H. 2009. Enhancing QA systems with complex temporal question processing capabilities. *Journal of Artificial Intelligence Research* **35**(2): 775.

Šarić, F., Glavaš, G., Karan, M., Šnajder, J., and Bašić, B. D. 2012. TakeLab: systems for measuring semantic text similarity. In *Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval 2012) in Conjunction with the First Joint Conference on Lexical and Computational Semantics (*SEM 2012)*, Montreal, Canada. Stroudsburg, PA: ACL.

Saurí, R., and Pustejovsky, J. 2012. Are you sure that this happened? Assessing the factuality degree of events in text. *Computational Linguistics* **38**(2): 261–99.

Setzer, A., Gaizauskas, R., and Hepple, M. 2003. Using semantic inferences for temporal annotation comparison. In *Proceedings of the Fourth International Workshop on Inference in Computational Semantics (ICOS-4)*. Nancy, France: INRIA, pp. 25–6.

Smith, C. S. 1999. Activities: states or events? *Linguistics and Philosophy* **22**(5): 479–508.

Soon, W., Ng, H., and Lim, D. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics* **27**(4): 521–44.

Sun, W., Rumshisky, A., and Uzuner, O. 2013. Evaluating temporal relations in clinical text: 2012 i2b2 challenge. *Journal of the American Medical Informatics Association* **20**(5): 806–13.

Surdeanu, M., Harabagiu, S., Williams, J., and Aarseth, P. 2003. Using predicate-argument structures for information extraction. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, Sapporo, Japan, vol. 1. Stroudsburg, PA: ACL, pp. 8–15.

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press

Surdeanu, M., and Turmo, J. 2005. Semantic role labeling using complete syntactic analysis. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, Ann Arbor, MI. Stroudsburg, PA: ACL, pp. 221–4.

Tannier, X., and Muller, P. 2008. Evaluation metrics for automatic temporal annotation of texts. In *Proceedings of the Conference on Language Resources and Evaluation*, Marrakech, Morocco. Paris, France: ELRA, pp. 150–5.

Tenny, C., and Pustejovsky, J. 2000. A history of events in linguistic theory. *Events as Grammatical Objects* **32**: 3–37.

Tong, H., Faloutsos, C., Gallagher, B., and Eliassi-Rad, T. 2007. Fast best-effort pattern matching in large attributed graphs. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY: ACM, pp. 737–46.

UzZaman, N., and Allen, J. 2010. TRIPS and TRIOS system for TempEval-2: extracting temporal information from text. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, Uppsala, Sweden. Stroudsburg, PA: ACL, pp. 276–83.

UzZaman, N., and Allen, J. F. 2011. Temporal evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2011)*, Portland, OR. Stroudsburg, PA: ACL, pp. 351–6.

UzZaman, N., Llorens, H., Derczynski, L., Verhagen, M., Allen, J., and Pustejovsky, J. 2013. SemEval-2013 Task 1: TempEval-3: evaluating time expressions, events, and temporal relations. In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013), in Conjunction With the Second Joint Conference on Lexical and Computational Semantcis (* SEM 2013)*, Association for Computational Linguistics, June, Atlanta. Stroudsburg, PA: ACL.

Vendler, Z. 1957. Verbs and times. *The Philosophical Review* **66**(2): 143–60.

Verhagen, M. 2007. Drawing TimeML relations with TBox. In F. Schilder, G. Katz and J. Pustejovsky (eds.), *Annotating, Extracting and Reasoning about Time and Events*. Berlin, Germany: Springer-Verlag, pp. 7–28.

Verhagen, M., Gaizauskas, R., Schilder, F., Hepple, M., Katz, G., and Pustejovsky, J. 2007. SemEval-2007 Task 15: TempEval temporal relation identification. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval 2007)*, Prague, Czech Republic. Stroudsburg, PA: ACL, pp. 75–80.

Verhagen, M., Sauri, R., Caselli, T., and Pustejovsky, J. 2010. SemEval-2010 Task 13: TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval 2010)*, Uppsala, Sweden. Stroudsburg, PA: ACL, pp. 57–62.

Wei, C. P., and Chang, Y. H. 2007. Discovering event evolution patterns from document sequences. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* **37**(2): 273–83.

Wolf, F., and Gibson, E. 2005. Representing discourse coherence: a corpus-based study. *Computational Linguistics* **31**(2): 249–87.

Wu, Z., and Palmer, M. 1994. Verbs semantics and lexical selection. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL 1994)*, Las Cruces, NM. Stroudsburg, PA: ACL, pp. 133–8.

Yang, Y., Carbonell, J. G., Brown, R. D., Pierce, T., Archibald, B. T., and Liu, X. 1999. Learning approaches for detecting and tracking news events. *Intelligent Systems and their Applications* **14**(4): 32–43.

Yang, C. C., Shi, X., and Wei, C. P. 2009. Discovering event evolution graphs from news corpora. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* **39**(4): 850–63.

Yangarber, R., Grishman, R., Tapanainen, P., and Huttunen, S. 2000. Automatic acquisition of domain knowledge for information extraction. In *Proceedings of the 18th Conference on Computational Linguistics*, Hong Kong, vol. 2. Stroudsburg, PA: ACL, pp. 940–6.

https://doi.org/10.1017/S1351324914000060 Published online by Cambridge University Press