A Linguistic Resource for Discovering Event Structures and Resolving Event Coreference

Cosmin Adrian Bejan and Sanda Harabagiu

Human Language Technology Research Institute
The University of Texas at Dallas
Richardson, TX 75083-0688, USA
{ady, sanda}@hlt.utdallas.edu

Abstract

In this paper, we present a linguistic resource that annotates *event structures* in texts. We consider an event structure as a collection of events that interact with each other in a given situation. We interpret the interactions between events as event relations. In this regard, we propose and annotate a set of six relations that best capture the concept of event structure. These relations are: *subevent*, *reason*, *purpose*, *enablement*, *precedence* and *related*. A document from this resource can encode multiple event structures and an event structure can be described across multiple documents. In order to unify event structures, we also annotate inter- and intra-document event coreference. Moreover, we provide methodologies for automatic discovery of event structures from texts. First, we group the events that constitute an event structure into event clusters and then, we use supervised learning frameworks to classify the relations that exist between events from the same cluster.

1. The Problem

Text and human communication abounds with reference to events and their interrelations. Events can determine other events or states, they may culminate in accomplishments or they may cause states to terminate. In texts, events are commonly described by verbs or nominalizations. But verbal and nominalized predicates do not refer only to events. Vendler (1967) has derived a seminal categorization of predicates into states, processes or activities, accomplishments, and achievements. The events, processes, achievements and states that occur in certain interrelations constitute an *event structure*.

Knowledge about event structure is extremely important. Based on this knowledge, different forms of inference, including temporal, causal, and intentional inference, may be produced. Currently, such knowledge, although very needed, is not readily available in any linguistic resource. FrameNet (Baker et al., 1998) and WordNet (Fellbaum, 1998) annotate event relations at conceptual level. Prop-Bank (Kingsbury et al., 2002) and VerbNet (Kipper et al., 2000) encode only predicate argument structures in texts. TimeBank (Pustejovsky et al., 2003b) contains annotations with all the events from a given text, but only temporal relations are considered, whereas the ACE corpus (LDC, 2005) focuses on a limited class of events that are annotated as complex structures involving entities, time expressions and values.

In order to discover event structures from texts, we have to determine: (1) which events belong to the same event structure; and (2) what relations exist between the events from the same structure. To address the first problem, we consider that the discovery of events from the same event structure is similar to the problem of event clustering. For example, given the event describing the *arrest* of a criminal, we discover that events such as the *search* for the criminal, the *accusation* of a crime, his *capture* following by his *interrogation* are part of the same event structure. Another approach for building event structures is by mapping events to semantic frames and linking them by using the frame relations that are encoded in FrameNet. However, since the frame relations are defined at conceptual level, it is difficult to capture all the events from an event structure in a given context.

To address the identification of relations between events, we have surveyed the literature for lists of relations between events, processes, states and achievements within the same event structure. Based on the theory of discourse relations (Hobbs, 1985; Mann and Thompson, 1988), the theory of cognitive semantics (Talmy, 2000), the theory of frame semantics (Fillmore, 1982) and the event ontology defined in (Sinha and Narayanan, 2005), we propose a set of six event relations that exist in an event structure. We also consider the temporal interpretation of each relation, such that we can employ event structures for temporal inference applications.

In this paper, we describe a linguistic resource that encodes event structure annotations as well as methodologies for automatic discovery of event structures from texts. In our representation, we consider that a document can encode multiple event structures. Also, the information about an event structure may be scattered across multiple documents and the unification of such information is granted whenever event coreference can be established. The result is a collection of documents annotated with (1) events, which may belong to different event structures, (2) event relations that hold between events from the same event structure, and (3) inter- and intra-document event coreference.

The remainder of this paper is organized as follows. Section 2 describes the event clustering model and a method for evaluating this model. Section 3 details the methods employed for recognizing relations that hold between the events from the same event structure. Section 4 discusses the event coreference framework. Section 5 presents the resulting resource and its usage for temporal inference. Section 6 summarizes the conclusions.

2. Discovering Event Clusters

A first step for discovering event structures and relations between events is the identification of event clusters. The task of clustering events that belong to the same event structure was cast as a generative probabilistic model that encodes latent event structures and makes possible recovering these structures using statistical inference. In particular, the generative model that we employed is called Latent Dirichlet Allocation (LDA), which was introduced in (Blei et al., 2003). In our adapted model, the basic idea is that a document is expressed as a probabilistic mixture of event structures and an event structure has assigned a probability distribution over events.

In this model, an event is the basic unit of discrete data. We considered as events all the predicates that fulfill the TimeML specifications (Pustejovsky et al., 2003a). According to TimeML annotation guidelines, events are classified as: (1) REPORTING, (2) PERCEPTION, (3) ASPECTUAL, (4) I_ACTION (intentional action), (5) I_STATE (states that refer to possible worlds), (6) STATE and (7) OCCURENCE events.

More formally, we represent a document d as a sequence of N_d events, $d = (e_1, e_2, \ldots, e_{N_d})$, and a *corpus* $\mathcal{C}$ as a collection of M documents, $\mathcal{C} = \{d_1, d_2, \ldots, d_M\}$. We assume that we have S event structures and E unique events in the corpus $\mathcal{C}$. The assignment of an event e_i to an event structure is facilitated by a hidden variable z_i, which takes values from 1 to S. We use an $E \times S$ matrix Φ to denote the distribution of events associated to each event structure and an $S \times M$ matrix Θ to denote the distribution of event structures in each document. With these notations, the generative process can be described as follows:

---

1. For each document $d \in \{d_1 \ldots d_M\}$:

   1. Choose $\theta^{(d)} \sim \text{Dirichlet}(\alpha)$.
   2. For each event $e_i^{(d)}$, $i \in \{1 \ldots N_d\}$:
      1. Given the document d, choose an event structure $z_i^{(d)} \sim \text{Multinomial}(\theta^{(d)})$.
      2. Given the event structure $z_i^{(d)}$, choose an event $e_i^{(d)} \sim \text{Multinomial}(\phi^{z_i^{(d)}})$.

Therefore, for each document d, the generative process is performed in three steps. First, a distribution over event structures, $\theta^{(d)}$, is sampled from a prior Dirichlet distribution with parameter α. Next, an event structure is assigned to each event in the document according to the sampled distribution. Finally, an event is chosen from the multinomial distribution over events given the sampled event structure chosen in the previous step. This process repeats until convergence.

We experimented with this new method on 50 event structures and trained the LDA model such that we could derive the probability of any new event to be assigned to any of the initial event structures. We also experimented with this event clustering method using the event annotations from TimeBank corpus. The event clusters were obtained by employing the `lda-c` tool, which is an implementation of the LDA model and is available at *http://www.cs.princeton.edu/~blei/lda-c/*. To assess the quality of the model, we considered FrameNet-based scenarios as baseline event clusters. We automatically built these scenarios by mapping the TimeBank events to semantic frames and by connecting them through frame-to-frame relations defined in FrameNet. The frame relations annotated in FrameNet are listed in Table 1. Each frame relation defined in FrameNet is asymmetric and holds between a more abstract frame, called the Super-Frame, and a more specific frame, called the Sub-Frame (Ruppenhofer et al., 2005).

Inheritance	Perspective On	Causative Of	See Also
Subframe	Using	Inchoative Of	Precedes

Table 1: The FrameNet frame relations.

Figure 1 illustrates a common employment scenario that is built using the FrameNet frame relations. For instance, the SUBFRAME relation holds between a frame representing a complex process and frames that characterize sub-processes of the complex process. In this example, *Employee's scenario* is a complex frame that has the following sub-frames: *Get a job*, *Being employed* and *Quitting*. A sub-frame can also be a complex frame for other frames. The PERSPECTIVE ON relation is used when at least two points of view are expressed with respect to a situation. For instance, in Figure 1, the *Get a job* and *Hiring* frames refer to the same situation, which corresponds to the *Employment start* frame, but *Get a job* is considered from an employee's perspective whereas *Hiring* is considered from an employer's perspective.
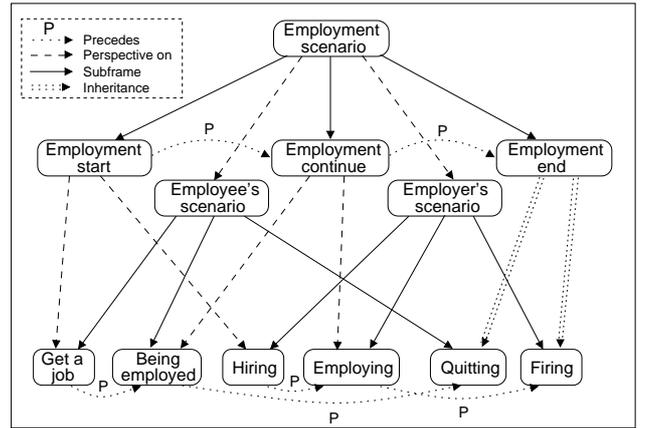

Figure 1: Building FrameNet scenarios using frame relations.

The evaluation of the LDA-based event clustering method was performed by measuring the coverage of the event clusters over the FrameNet scenarios. Bejan (2008) details promising results obtained by this method.

3. Recognizing Relations in Event Structures

After determining which events belong to the same event structure, the next step is to define a set of relations that

hold between the events from the same cluster. Undoubtedly, the inter-event relations encoded in FrameNet are useful. However, they do not cover all the relations that may define an event structure. For example, no specific relations of ENABLEMENT/DISABLEMENT exist to capture PRECONDITIONS, no EFFECTS are considered and no distinction between CAUSALITY, PURPOSE and REASON is considered. Also, because they are defined at conceptual level, the scenarios defined by frame relations are too generalized and therefore they cannot capture all the events happening in specific situations.

In our effort, we consider the following set of relations that capture the concept of event structure:

- SUBEVENT is similar with the SUBFRAME relation from FrameNet. It holds between an event A that is part of a composite event B. The composite event B can have multiple subevents and an subevent can be a composite event for other events. The hierarchy of events resulted using the SUBEVENT relation can encode complex semantic and temporal structures.

- REASON is a causal relation which happens between a reason event and a consequence event. When multiple reason events cause one consequence event, this relation is applied repetitively. For example, in the following sentence

  (S_1) *Diego Montoya is* **accused** *of* **leading** *Norte del Valle cartel and* **exporting** *tons of cocaine to the United States.*

  there is one REASON relation between *leading* and *accused* and another one between *exporting* and *accused*. In this context, the reason for the *accused* event to happen is caused by the *leading* and *exporting* events.

- PURPOSE is a causal relation which represents the intention of an event A to achieve a goal event B. In multiple cases, the presence of the signal word *to* between two consecutive events in the same sentence implies the existence of a PURPOSE relation between the two events:

  (S_2) *FBI officials Monday were* **checking** *fingerprint databases to* **confirm** *that it was, in fact, Montoya who was* **captured**.

  In this sentence, the *checking* event happened with the purpose of *confirming* by the FBI officials Montoya's personal identity.

- ENABLEMENT relation is a causal relation for which an event A allows an event B to happen, but does not necessarily cause B. An example of ENABLEMENT relation holds between *captured* and *checking* from sentence S_2. In this example, the *checking* event will happen only if the *captured* event happened.

- PRECEDENCE is similar with the PRECEDES relation from FrameNet. This relation determines a sequential ordering of two events belonging in the same event structure and does not hold between events from different event structures.

- RELATED refers to events between which there is a weak connection. For example, a related relation exists between a reporting event and an event mentioned in the reported statement.

It is worth mentioning that in many cases these relations depend on the context within which their corresponding events are used. For example, the two REASON relations exist in S_2 only because leading a drug cartel and exporting drugs are illegal activities. This is also the reason that this particular relation examples cannot be recovered using frame-to-frame relations. In our example, there is no connection between the *Leadership* frame, which is evoked by the *leading* event, and the *Notification of charges* frame, which is evoked by the *accused* event. Similarly, there is no connection between *Exporting*, which is evoked by the *exporting* event, and *Notification of charges*.

Table 2 lists various properties that hold for these binary relations. This table also shows the temporal interpretation associated for every event relation. For instance, if a relation $A \xrightarrow{\text{SUBEVENT}} B$ exists, then the time interval of the event A must be included in the time interval associated to the composite event B. The temporal relations corresponding to event relations allows us to build a chronological order of the events that belong to the same event structure.

Relation	Relation Properties	Temporal Interpretation
$A \xrightarrow{\text{Subevent}} B$	irreflexive asymmetric transitive	$A \xrightarrow{\text{During}} B$
$A \xrightarrow{\text{Reason}} B$	irreflexive asymmetric antitransitive	$A \xrightarrow{\text{Any relation}} B$
$A \xrightarrow{\text{Purpose}} B$	irreflexive asymmetric antitransitive	$A \xrightarrow{\text{Before}} B$
$A \xrightarrow{\text{Enablement}} B$	irreflexive asymmetric antitransitive	$A \xrightarrow{\text{Before}} B$
$A \xrightarrow{\text{Precedence}} B$	irreflexive asymmetric transitive	$A \xrightarrow{\text{Before}} B$
$A \xrightarrow{\text{Related}} B$	irreflexive asymmetric transitive	$A \xrightarrow{\text{Any relation}} B$

Table 2: Properties of event relations.

In order to detect event relations from the same event structure, we trained support vector machine multi-class classifiers in a supervised learning framework. In the data processing phase of this framework, we extracted syntactic parse trees for the annotated sentences using Collins' parser (Collins, 1997). For example, Figure 2 shows the syntactic parse tree representation associated to the sentence S_1 with the annotated events highlighted. The syntactic information encoded in these representations helps us explore the properties of event relations. In the learning phase, we considered a learning instance as a pair of events, where each event is mapped to a constituent from a syntactic parse tree. The features that we extracted for an event pair (e_1, e_2) in our preliminary experiments are:

- e WORD: The head word of the constituent associated to event e.

- $e_1 + e_2$ WORD: The combination of the e_1 WORD and e_2 WORD features.

S
NP VP
NNP NNP VBZ VP
Diego Montoya is VBN PP
accused IN S
of VP
VP CC VP
VBG NP and VBG NP PP
leading NNP NNP NNP NN **exporting** NP PP TO NP
Norte del Valle cartel NNS IN NP to DT NNP NNPS .
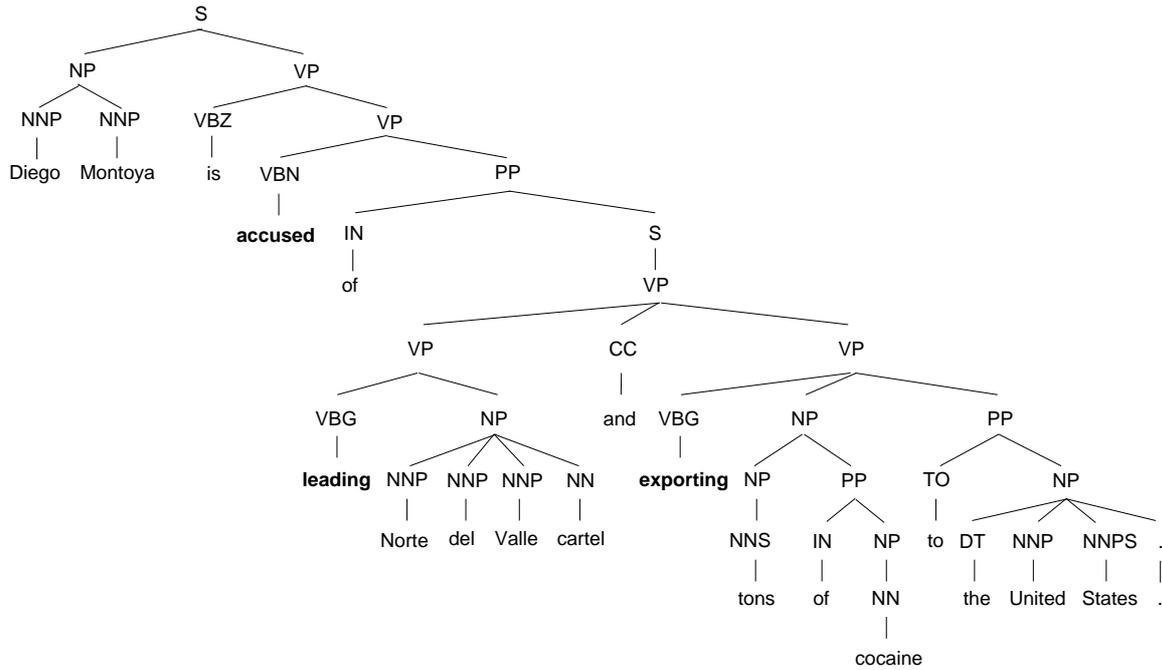tons of NN the United States .
cocaine

Figure 2: Syntactic parse tree representation of the sentence S_1.

- e STEM: The stem word of the e WORD feature.

- $e_1 + e_2$ STEM: The combination of the e_1 STEM and e_2 STEM features.

- e CLASS: The lexical class associated to the event e. This feature can take three values: verb, noun and adjective.

- $e_1 + e_2$ CLASS: The combination of the e_1 CLASS and e_2 CLASS features.

- e POS: The head part of speech associated to e.

- $e_1 + e_2$ POS: The combination of the e_1 POS and e_2 POS features.

- SIGNAL WORDS: A set of binary features testing for the presence of words between the e_1 and e_2 events such as: *'as a result', 'and', 'because', 'while', 'next'*, etc. This feature is extracted only for pairs of events that belong in the same sentence.

- e PREVIOUS WORD: The first word preceding the event expression.

- e NEXT WORD: The first word following the event expression.

- IDENTICAL: Binary feature indicating whether the e_1 STEM and e_2 STEM features are identical.

- SYNONYMOUS: Binary feature indicating whether the e_1 STEM and e_2 STEM features are synonymous.

- NOMINALIZATION: Binary feature indicating whether one of the events is a nominalization of the other event.

- ADJACENT EVENTS: In case when they belong in the same sentence, this feature tests whether e_1 and e_2 are adjacent.

- SAME CLAUSE: Binary feature testing whether the two events belong in the same clause. This feature is extracted only for pairs of events from the same sentence.

- DIRECTED PARSE TREE PATH: The path in the syntactic parse tree between the constituents associated to the e_1 and e_2 events. For example, the path between the *leading* and *accused* events from the syntactic parse tree illustrated in Figure 2 is VBG↑VP↑VP↑S↑PP↑VP↓VBN. When the events belong to different sentences, this feature is computed from the combination of the partial paths associated to e_1 and e_2. A partial path corresponding to an event is the path in the parse tree between the event constituent and its root constituent.

- UNDIRECTED PARSE TREE PATH: The same syntactic path as DIRECTED PARSE TREE PATH without preserving the movement direction.

- SAME SENTENCE: Indicates whether the two events belong in the same sentence.

- SAME DOCUMENT: Indicates whether the two events belong in the same document.

- DISTANCE IN WORDS: If the events belong in the same sentence, this feature counts the number of words between the two events.

- DISTANCE IN SENTENCES: If the events belong in the same document, this feature counts the number of sentences between the two events.

- POSITION: Binary feature indicating whether e_1 occurs before or after e_2. This feature is extracted only for pairs of events that belong in the same document.

- SUBFRAME: When both events can evoke semantic frames, this feature indicates whether their corresponding frames are connected through a SUBFRAME relation. For those events that can evoke multiple semantic frames, a frame disambiguation task is applied (Bejan and Hathaway, 2007).

- SAME SEMANTIC ROLES: In order to establish whether the two events share the same participating entities, happen at the same time and take place in the same location, we extracted a set of binary features from the semantic roles corresponding to these events. Therefore, we first extracted the semantic structures for the two events using our semantic parser (Bejan and Hathaway, 2007), and then test whether these structures share the same semantic roles or not.

4. Event Coreference

We have noticed that several portions of an event structure are mentioned in different documents or in different parts of the same document. In order to unify such information, we have to have the ability to perform event coreference.
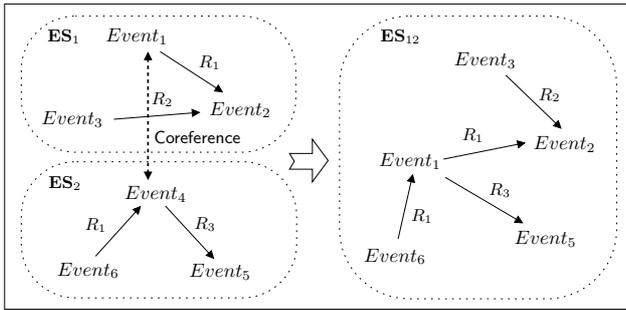

Figure 3: Unifying event structures.

Figure 3 shows how two event structures ES_1 and ES_2 can be unified in ES_{12} if $Event_1$ and $Event_4$ are identified as co-referent. Moreover, the unification of two event structures can identify additional event coreference, if the semantics of the relations between events is considered. If, for example, the PURPOSE relation is considered, an event cannot have more than one purpose/goal, and after unifying two event structures from one event E_i we have two distinct PURPOSE relations pointing to E_j and E_k, then we may conclude that E_j and E_k must be coreferring or must belong in the same event structure as well. Furthermore, some relations from the event structure may be also consolidated. If from E_i we have two distinct PRECEDENCE relations pointing to E_j and E_k, we need to resolve their temporal precedence and consolidate the PRECEDENCE relations accordingly. In the same way, contradictory information in the event structure may be detected and resolved. In order to identify event coreference between two events we used the following list of requirements: (a) both events are expressed by the same predicate or their synonyms or hyponyms, and (b) whenever specified, both predicates have the same arguments.

An example that shows how event coreference relation helps in identifying event structure relations is illustrated in Figure 4. In this example, a coreference relation exists
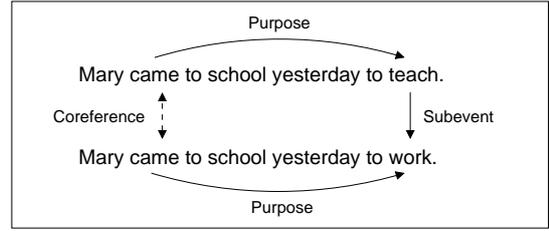

Figure 4: Contribution of coreference relation for detecting event structure relations.

between *came* from the first sentence and *came* from the second sentence because the requirements mentioned above are fulfilled. This coreference relation together with the two PURPOSE relations impose a relation to hold between the *teach* and *work* events. In this context, a SUBEVENT relation is applied.

5. A Linguistic Resource for Event Structure and Event Coreference

In order to use these methodologies for clustering events, for learning event relations and for solving event coreference, we created a collection of documents with: (1) event annotations, (2) inter- and intra-document event coreference annotations and (3) event relation annotations. The annotation procedure for this resource is performed in five steps:

STEP 1: *Theme selection.* The first question for building a linguistic resource is which documents to choose for annotation. We decided to choose web articles from Google News because it has the facility to cluster the articles by subject matter, and, therefore, it allow us to annotate multiple parallel articles describing the same event scenarios. In consequence, this option has multiple advantages: (1) the cross-document identical events can be easily identified; (2) the event structures mentioned in one document can be unified with other events by using inter- and intra-document coreference annotations (see Figure 3); and (3) various topics about specific events can be selected by using Google's news archive search service. For our linguistic resource, we extracted articles using search keywords such as *attack*, *death*, *earthquake*, *catastrophe*, etc.

STEP 2: *Web documents processing.* For each theme, we extracted on average seven parallel articles. These web documents are manually cleaned and saved into files. The text files are further tokenized and splitted into sentences (one sentence per line).

STEP 3: *Event annotation.* We annotated events in texts in accordance with TimeML specification (Pustejovsky et al., 2003a). For this task we used an annotation tool called Callisto, which is available at *http://callisto.mitre.org*.

STEP 4: *Event coreference annotation.* Callisto has the ability to convert annotated documents into TimeML

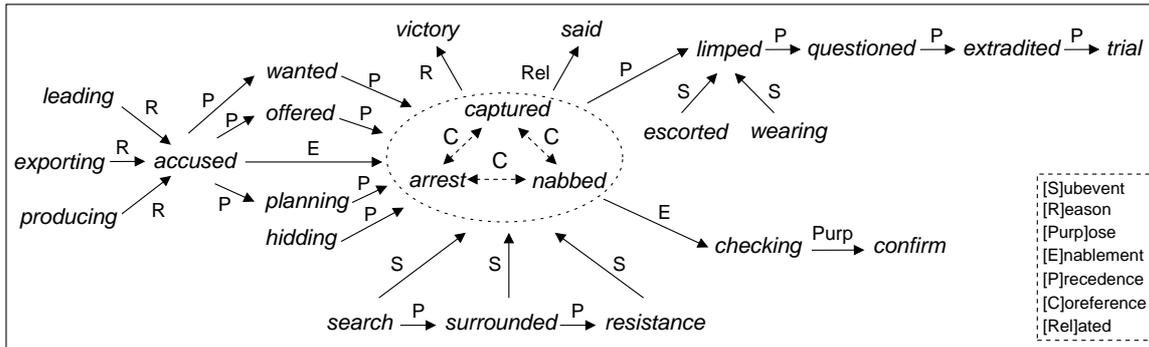Figure 5: Example of event structure annotations.

Figure 6: Building event structures using FrameNet relations.

format documents. This allows us to use for annotating coreferring events belonging in the same TimeML document a tool called Tango, which is available at *http://timeml.org/site/tango/tool.html*. For annotating cross-document coreferring events we used a modified version of Tango on generated document pairs between files with the same theme.

STEP 5: *Event relation annotation.* To be able to annotate the event relations defined in section 2, we modified Tango and performed a similar method as the one mentioned in the previous step for inter-document event coreference.

Figure 5 illustrates an example of event structure annotations on several articles describing the arrest of a Colombian drug leader, Diego Montoya, on September 10, 2007. As can be seen in this figure, most of the events are linked to the *captured* event, which is the main event of the story. To describe the events depicted in Figure 5, we summarize the story as follows. Diego Montoya was *captured* because he was *accused* of *leading* an important drug cartel in Columbia and because he was *producing* and *exporting* drugs. Between the *accusation* and his *arrest* a series of events happened: Montoya was *wanted* by FBI that *offered* a reward for information leading to his *arrest*. Aware of the *arrest*, FBI officials were *checking* fingerprint databases to *confirm* that it was, in fact, Montoya who was *captured*. The article also describes in detail how Montoya was *captured*: first, a *search* operation was initiated, and, when the drug leader was *surrounded* by the Colombian forces, he put up no *resistance*. After his *arrest*, Montoya was *escorted* to be *questioned* before being *extradited* to the United States for a *trial*. Colombian officials *said* that Montoya's *arrest* was a big drug war *victory*.

The mapping of this example in the FrameNet crime scenario is illustrated in Figure 6. In this figure, only the shaded frames contain events from our example. As can be noticed, the frame scenario contains only approximately 20 percent of all the events depicted in Figure 5. As we mentioned in section 3, one reason for a weak coverage of event structures on frame scenarios is that frame relations map concepts and not specific events. Another well know problem of the FrameNet resource is related to the coverage of the lexical units that are assigned to frames. For instance, the events *resistance*, *surrounded* and *offered* from the example are not yet covered in FrameNet. The
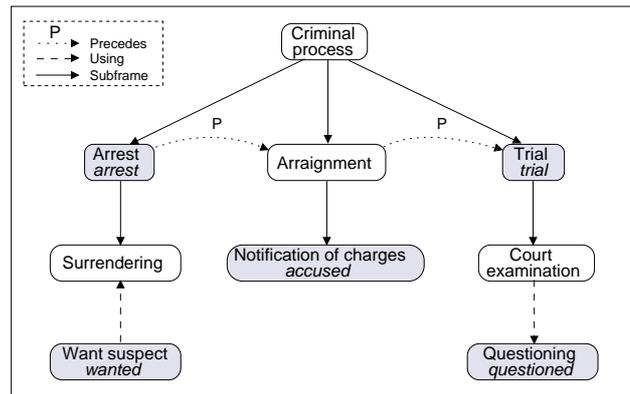
last problem that we noticed when mapping our example to the FrameNet crime scenario refers to the temporal order of *accused* and *captured* events: in FrameNet a $captured \xrightarrow{\text{BEFORE}} accused$ relation can be inferred, whereas, in this context, an AFTER temporal relation exists between these two events.

Based on the temporal interpretation associated to every event relation, we can derive a strict partial order of the events that belong in the same event structure. For example, Figure 7 illustrates the chronologies of events associated to the event structure depicted in Figure 5. Furthermore, these chains of events from the same event structure can be linked to other events from the document collection if the TimeML temporal relations and the semantic roles that encode a temporal expression are considered.

6. Conclusions

We created a novel linguistic resource for encoding event structures in texts, which are defined by a set of six event relations, as well as for capturing inter-document and cross-document event coreference. We argued that this resource is useful for performing several forms of inference and showed in an example how to use the event structures for deriving a chronological order of events.

In addition, we described how the event structures can be recovered through a generative probabilistic model and how the event relations can be automatically classified in a supervised learning framework.
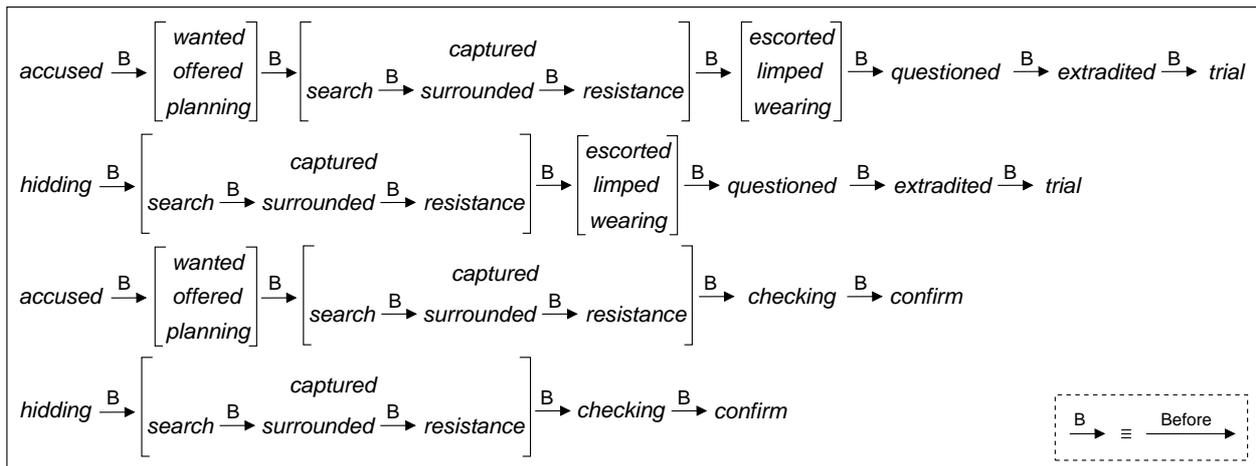
Figure 7: Deriving chronologies of events from event structures.

7. References

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *Proceedings of the COLING-ACL*.

Cosmin Adrian Bejan and Chris Hathaway. 2007. UTD-SRL: A Pipeline Architecture for Extracting Frame Semantic Structures. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 460–463.

Cosmin Adrian Bejan. 2008. Unsupervised Discovery of Event Scenarios from Texts. In *Proceedings of the 21st Florida Artificial Intelligence Research Society International Conference (FLAIRS), Applied Natural Language Processing track*.

David Blei, Andrew Ng, and Michael Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, pages 993–1022.

Michael Collins. 1997. Three generative, lexicalized models for statistical parsing. In *Proceedings of the ACL–EACL*, pages 16–23.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Charles J. Fillmore. 1982. Frame Semantics. In *Linguistics in the Morning Calm*.

Jerry R. Hobbs. 1985. On the coherence and Structure of Discourse. In *CSLI Technical Report 85–37*.

Paul Kingsbury, Martha Palmer, and Mitch Marcus. 2002. Adding Semantic Annotation to the Penn TreeBank. In *Proceedings of Human Language Technology (HLT-2002)*.

Karin Kipper, Hoa Trang Dang, and Martha Palmer. 2000. Class-Based Construction of a Verb Lexicon. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*, pages 691–696.

LDC. 2005. ACE (Automatic Content Extraction) English Annotation Guidelines for Events, version 5.4.3 2005.07.01 edition.

William C. Mann and Sandra A. Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.

James Pustejovsky, Jose Castano, Bob Ingria, Roser Sauri, Rob Gaizauskas, Andrea Setzer, and Graham Katz. 2003a. TimeML: Robust Specification of Event and Temporal Expressions in Text. In *Proceedings of Computational Semantics Workshop (IWCS-5)*.

James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, David Day, Lisa Ferro, Robert Gaizauskas, Marcia Lazo, Andrea Setzer, and Beth Sundheim. 2003b. The TimeBank Corpus. In *Corpus Linguistics*, pages 647–656.

Josef Ruppenhofer, Michael Ellsworth, Miriam R. L. Petruck, and Christopher R. Johnson. 2005. FrameNet II: Extended Theory and Practice. Technical report, ICSI Berkeley.

Steve Sinha and Srini Narayanan. 2005. Model Based Answer Selection. In *Proceedings of Inference for Textual Question Answering Workshop, AAAI 2005*.

Leonard Talmy. 2000. *Toward a Cognitive Semantics, vol. I and II*. The MIT Press, Cambridge, Massachussets, USA.

Zeno Vendler. 1967. *Linguistics in Philosophy*. Ithaca:Cornell University Press.