Using Clustering Methods for Discovering Event Structures

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
http://www.hlt.utdallas.edu/~{ady, sanda}

Introduction

In the process of understanding specific situations, domain experts manually build event models that provide the infrastructure for reasoning with events and for simulating event scenarios. Such kind of models are also encoded in document collections in which each document is represented as a sequence of events. When describing situations in texts, events do not operate independently, but rather they are inter-related with other events from the same scenario. For example, in a CRIME scenario, a person is *accused* of a crime, then that person is *arrested* and *interrogated* after which a *trial* is held. We define an *event structure* as a collection of events that interact with each other in a specific situation.

Extracting event structures from texts allows us to perform various forms of inference over events. For example, given an event e from an event structure, we can determine which events are likely to happen after e happened. Another example is to compute the probability that an event e from an event structure s is likely to happen given the fact that a set of events from s and disjoint from e already happened in a text t. In this specific example, the event e is not required to be present in t. In general, if we know what events interact with each other in an event structure, we can build more complex inference models dealing with *causality*, *intentionality* or *temporality* of events.

Our goal is to provide a method that automatically extracts event structures from texts. In order to build event structures we need (1) to determine the set of events that belong to the same event structure and (2) to establish what relations exist between the events from the same structure. In this abstract, we describe the theoretical frameworks for solving both tasks, but we detail more the first task for which we also show our preliminary results.

The motivation of this work is in the same spirit with the work performed for solving Topic Detection and Tracking tasks (Allan 2002). However, instead of considering clusters as topically related bag of words like in a classic topic modeling approach, our goal is to build structured event representations and to interpret the event interactions that exist in these representations.

Discovering Event Clusters

A first step for discovering event structures is to group events that belong to the same structure into clusters. We cast this task as a generative probabilistic model that encodes latent event structures and makes possible recovering this structures using statistical inference. In particular, the model that we employed is called Latent Dirichlet Allocation (LDA) and was introduced in (Blei, Ng, & Jordan 2003).

The basic idea in our event clustering model is that documents are expressed as probabilistic mixtures of event structures, while each event structure in a document has assigned a probability distribution over the events mentioned in the document. In this model, the events are considered discrete random variables, a document contains a fixed number of events, and each event e_i, $i \in \{1, \ldots, E\}$, is an element from an event collection $\mathcal{E}$. A document d is represented as a sequence of N_d events $d{=}(e_1, e_2, \ldots, e_{N_d})$, while a corpus $\mathcal{C}$ is represented as a collection of M documents $\mathcal{C}{=}\{d_1, d_2, \ldots, d_M\}$ having the total number of events $N{=}\sum_{d=1}^{M} N_d$.

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 the total number of event structures considered, S. If $P(z)$ denotes the probability distribution over event structures and $P(e|z)$ is the probability distribution over the events e given the event structure z, the probability of the ith event from a document d is given by:

$$P(e_i) = \sum_{j=1}^{S} P(e_i|z_i = j)P(z_i = j)$$

In this formula, $P(e_i|z_i = j)$ is the probability of e_i given the jth event structure and represents how significant the event e_i is for the jth structure, while $P(z_i = j)$ is the probability that the jth structure was chosen for the ith event from d. In our event clustering model, the multinomial distribution associated to each structure, $P(e|z)$ is parameterized by an $E{\times}S$ matrix Φ such that $P(e|z = j) = \phi^{(j)}$ is the multinomial distribution over events for the jth structure ($\sum_{i=1}^{E} \phi_i^{(j)} = 1$). Similarly, the distribution of event structures associated to each document d is parameterized by an $S{\times}M$ matrix Θ such that $P(z) = \theta^{(d)}$ is the multinomial distribution of the event structures corresponding to d ($\sum_{j=1}^{S} \theta_j^{(d)} = 1$). Using these notations, the generative process for each document $d \in \mathcal{C}$ is described as follows:

1. Choose $\theta^{(d)} \sim$ Dirichlet(α).
2. For each event $e_i^{(d)}$, $i \in \{1 \ldots N_d\}$:
   1. Choose a structure $z_i^{(d)} \sim$ Multinomial($\theta^{(d)}$).
   2. Choose an event $e_i^{(d)} \sim$ Multinomial($\phi^{z_i^{(d)}}$) conditioned on the event structure $z_i^{(d)}$.

Aside from the advantages of being unsupervised, the event clustering model has the benefit that all the events from each structure can be interpreted separately. Another advantage is related to how general or how specific we want our event structures. Setting a lower value for the number of structures in the model will derive more general structures whereas a higher value for the number of structures will produce very specific events in each structure.

Recognizing Relations in Event Structures

In our work, we interpret the interactions between events as event relations. In order to define the set of relations that best capture the concept of event structure, we surveyed the literature in the theory of discourse relations (Hobbs 1985; Mann & Thompson 1988), cognitive semantics (Talmy 2000), frame semantics (Fillmore 1982) and event ontologies (Sinha & Narayanan 2005) and decided to tackle the following set of relations that hold between the events from the same structure:

- SUBEVENT holds between an event that is part of a composite event. A subevent can be a composite event for other events.
- REASON is a causal relation which happens between a reason event and a consequence event.
- PURPOSE is a causal relation which represents the intention of an event A to achieve a goal event B.
- ENABLEMENT is a causal relation for which an event A allows an event B to happen, but does not necessarily cause B.
- PRECEDENCE determines a sequential ordering of two events belonging in the same event structure.
- 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.

We cast the recognition of event relations as a multi-class classification problem. To be able to train classifiers that automatically detect the relation for a given pair of events, we started to annotate each type of event relation from a collection of web articles. We first annotated the events using Callisto (an annotation tool which is available at *http://callisto.mitre.org*) and then the event relations using a modified version of Tango (an annotation tool which is available at *http://timeml.org/site/tango/tool.html*). In the annotation process, we annotated the events in accordance with the TimeML specifications (Pustejovsky *et al.* 2003).

Preliminary Results

For the task of event clustering, we chose web articles from Google News and annotated them by following the procedure described in the previous section. The first version of our corpus contains documents from the domain of terrorist attacks (17 articles), natural disasters (7 articles), death of famous personalities (14 articles) and arrests of reputed criminals (7 articles) with a total of 3964 event annotations from an event collection of 1262 unique events.

Cluster 9		**Cluster 13**		**Cluster 31**		**Cluster 47**	
event	$\phi^{(9)}$	event	$\phi^{(13)}$	event	$\phi^{(31)}$	event	$\phi^{(47)}$
said	.1088	died	.0544	said	.1020	said	.0489
attack	.0476	said	.0476	sang	.0612	wanted	.0217
attacks	.0272	earthquake	.0408	funeral	.0408	captured	.0190
explosions	.0204	reported	.0272	died	.0204	alleged	.0190
reported	.0204	damage	.0272	buried	.0204	armed	.0163
bombings	.0204	destroyed	.0204	ovation	.0204	extraditions	.0163
act	.0204	tsunamis	.0204	eulogized	.0204	trafficking	.0135
claimed	.0136	according	.0204	applauded	.0204	extradited	.0135
bombing	.0136	swept	.0204	grieves	.0204	capture	.0135
packed	.0136	disaster	.0204	honored	.0204	operation	.0108
came	.0136	told	.0136	ceremony	.0204	reported	.0108
called	.0136	forcing	.0136	accompanied	.0204	killed	.0108
ripped	.0136	quake	.0136	opened	.0204	take	.0108
explosion	.0136	washed	.0136	came	.0204	arrested	.0108
placed	.0136	injured	.0136	read	.0204	brought	.0108
blasts	.0068	flooding	.0136	captivated	.0204	arrest	.0081
detonated	.0068	missing	.0136	dressed	.0204	escorted	.0081
killed	.0068	declared	.0136	clutched	.0204	testify	.0081
exploded	.0068	recovered	.0136	recalled	.0204	pledging	.0081

Table 1: Examples of learned event clusters.

For extracting the event scenarios we used the `lda-c` tool, which is an implementation of the LDA model and is available at *http://www.cs.princeton.edu/∼blei/lda-c/*. In our experiments, the number of clusters S was set to 50. Table 1 lists four event clusters learned by the LDA-based model, which was trained on the annotated corpus. Every column of this table lists top 19 events sorted by their relevance to the corresponding event cluster. As can be observed from these preliminary results, our model successfully reproduced the event clusters that correspond to the four domains we chose for this experiment.

Conclusions

Discovering event structures from texts constitutes a stepping stone in the process of understanding and reasoning about texts. We introduced a novel method for extracting event structures from a collection of documents. Our preliminary results proved that the LDA-based model is a valid approach for extracting clusters of events that belong to the same event structure.

References

Allan, J. 2002. Introduction to Topic Detection and Tracking. In Allan, J., ed., *Topic Detection and Tracking, Event-based Information Organization*, 1–16. Kluwer Academic Publishers.

Blei, D.; Ng, A.; and Jordan, M. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research* 993–1022.

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

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

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

Pustejovsky, J.; Castano, J.; Ingria, B.; Sauri, R.; Gaizauskas, R.; Setzer, A.; and Katz, G. 2003. TimeML: Robust Specification of Event and Temporal Expressions in Text. In *Fifth International Workshop on Computational Semantics (IWCS-5)*.

Sinha, S., and Narayanan, S. 2005. Model Based Answer Selection. In *Inference for Textual Question Answering Workshop, AAAI 2005*.

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