Unsupervised Discovery of Event Scenarios from Texts

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

Abstract

We propose a new unsupervised learning approach for discovering *event scenarios* from texts. We interpret an event scenario as a collection of related events that characterize a specific situation. The approach uses the Latent Dirichlet Allocation (LDA) probabilistic model described in (Blei, Ng, & Jordan 2003) to automatically learn the probability distribution of events corresponding to event scenarios.

We performed experiments on an event annotated corpus and compared the automatically extracted event scenarios with frame scenarios defined in FrameNet (Baker, Fillmore, & Lowe 1998). The results show a better coverage for those event scenarios that are described in more detail in the event annotated corpus. When compared with a smoothed unigram model, the event scenario model achieves a perplexity reduction of 93.46%.

Columbia **captures** top cocaine boss **wanted** by U.S.
A reputed drug lord on the FBI's top 10 most-wanted list was **captured** in western Colombia, the interior minister **said** Monday, in a major **blow** to the country's largest remaining drug cartel.
Diego Montoya, who **sits** alongside Osama bin Laden as the only major alleged drug trafficker on the FBI list, is **accused** of **leading** the Norte del Valle cartel and **exporting** tons of cocaine to the United States.
The FBI had **offered** a reward of $5 million for information **leading** to his **arrest**.
Aware of the **arrest**, FBI officials Monday were **checking** fingerprint databases and otherwise **trying** to **confirm** that it was, in fact, Montoya who was **captured**, **said** spokesman Richard Kolko.
Montoya **put** up no **resistance** when the army finally **cornered** him in the cartel's stronghold of Valle del Cauca state on the Pacific Coast, officials **said**.
He is to be **questioned** before being **extradited** to the U.S. for **trial**, a process that Santos **said** would take at least two months.

Figure 1: Excerpt describing the arrest of Diego Montoya.

Introduction

With the avalanche of electronic text collections descending from all over the web, new forms of document processing that facilitate automatic extraction of useful information from texts are required. One approach for understanding the key aspects of a document or of a set of documents is to analyze the events in the document(s) and to automatically find scenarios of related events. We call an *event scenario* a set of events that can interact with each other in specific situations. For example, Figure 1 shows an excerpt from a web article describing the arrest of a Columbian drug dealer, Diego Montoya, on September, 2007. The events from this excerpt, shown in boldface in Figure 1, capture the event scenario describing the arrest of a criminal. The *accusation* of a crime, the *capture* of the criminal followed by his *interrogation* and *trial* are typical events that happen in a CRIME scenario.

Clustering the interrelated events into scenarios constitutes the foundation of studying various forms of interactions between events. If we know what events can happen in a specific situation or if we know what events can interact with a given event or set of events, we can build more complex inference models for dealing with *causality*, *intentionality* and *temporality* of events. In addition, this approach can be used in applications on commonsense reasoning such as *statistical story extraction* or *automatic narrative comprehension*.

We propose an unsupervised learning method for discovering event scenarios using the LDA model. In this method, we represent the documents from the event annotated corpus as a collection of events. Following the LDA model, each document is expressed as a probabilistic mixture of event scenarios and each event scenario is defined as a probability distribution of events. We compare the scenarios extracted by the LDA-based model with several frame scenarios encoded in FrameNet (FN). The main advantage of the event scenario model is that it is able to extract the event scenarios in an unsupervised fashion, assuming that the events in the documents are already extracted. Therefore, this approach can model large collections of documents without prior analysis or special initializations and can be adapted to different document collections without requiring additional efforts. Moreover, since a document can express multiple event scenarios, this model also captures the association that exists between these scenarios.

The rest of the paper is structured as follows. In section

2, we present a set of relations that hold in event scenarios. In this section, we also describe how the event scenarios, together with these event relations, form structured representations that can encode complex semantic and temporal structures. In section 3, we describe how we can build event scenarios using FrameNet frame relations and discuss the limitations of these frame-based scenarios. We present the event scenario model based on the Latent Dirichlet Allocation generative model in section 4. In section 5, we show the experimental results and compare the event scenario model with a smoothed unigram model. Section 6 summarizes the conclusions.

Structured Event Representations

The best way to understand our motivation for automatically discovering event scenarios from texts is to present "the big picture" of what we want to achieve. Our ultimate goal is to automatically extract structured representation of events from documents, where every event from such an event structure interacts with one or multiple events from the same structure.

- The **SUBEVENT** relation holds between an event A that is part of a composite event B. A composite event can have multiple subevents, which can be involved in complex semantic and temporal structures.
- The **REASON** relation is a causal relation that happens between a reason event A and a consequence event B. When multiple reason events cause one consequence event, this relation is applied successively.
- The **PURPOSE** relation is a causal relation which represents the intention of an event A to achieve a goal event B.
- The **ENABLEMENT** relation is a causal relation for which an event A allows an event B to happen, but does not necessarily cause B.
- The **PRECEDENCE** relation determines a sequential ordering of two events belonging to the same event structure. When the events are explicitly anchored to non overlapping time intervals, but they are not linked by any event structure relation, the PRECEDENCE relation does not apply.
- The **RELATED** relation refers to event structures 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.

Table 1: Event relations in structured scenarios.

We interpret event interactions as event relations. In order to propose the relations that best define the concept of an event structure, we surveyed the literature on the theory of discourse relations (Hobbs 1985; Mann & Thompson 1988), frame semantics (Fillmore 1982), and event ontologies (Sinha & Narayanan 2005) and concluded with the set of relations listed in Table 1. As an example, for the article illustrated in Figure 1, the *leading* and *exporting* events represent REASONS for *accusation*, the *accusation* is the ENABLER for *arrest*, the PURPOSE of *trying* is *confirm*, *extradited* PRECEDES *trial*, etc.

The first step for discovering event structures is to group the events from the same structure into clusters. In our approach, each cluster corresponds to an event scenario. This approach diverge from a classic topic modeling approach because we constrain the clusters to contain only events instead of words and because the event relations allows us to consider structured event representations instead of topically related bag of words. The event representations can be further extended by using semantic parsers to extract the semantic and temporal information associated to events from event structures.

Scenarios Using Semantic Frames

The Berkeley FrameNet project (Baker, Fillmore, & Lowe 1998) is an ongoing effort to build a semantic lexicon for English based on the theory of *frame semantics* (Fillmore 1982). The frame semantics theory models the meaning of words or word expressions, also called target words or predicates, into conceptual structures that characterize scenes or situations called *semantic frames*. A semantic frame can contain multiple target words while a target word can evoke multiple frames. Therefore, for polysemous target words a frame disambiguation task is required. The FrameNet lexicon also encodes frame-to-frame relations that allow us to group frames into frame scenarios. The list of frame relations is presented in Table 2.

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

Table 2: The FrameNet frame relations.

Based on the theory of frame semantics, Sinha & Narayanan (2005) have built an event ontology in which each event is associated to a semantic frame. Following the same intuition, we can build event scenarios by mapping the events extracted from texts to semantic frames and by grouping them using the frame relations. For instance, Figure 2 illustrates the frames and frame relations that encode the CRIME scenario derived from FrameNet. In this scenario, only the shaded frames contain target words that can be mapped to events from Figure 1. After the mapping

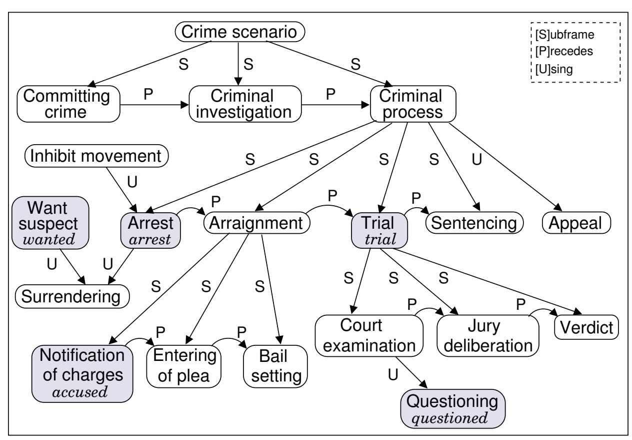

Figure 2: Representing event scenarios using frame relations.

method is applied, the resulting event scenario is composed of the following events: *wanted*, *arrest*, *trail*, *accused* and *questioned*.

There is no doubt that frame relations are useful for discovering event scenarios. However, the method of extracting scenarios using frames has several limitations. First, a well known problem of the FrameNet project is related to the coverage of the lexical database. In spite of being a large resource, there are still many target words and semantic frames to annotate. For instance, the *resistance*, *offered*, and *cornered* events from our example are not yet covered in FrameNet, and hence they will never be assigned to a frame scenario.

Second, the method suffers from poor scalability and domain adaptability. For any document collection, the method will always extract only the scenarios corresponding to those frames that are evoked by the events from the collection. Moreover, since it does not take into account how many times a specific event occurs in the document collection, or how frequently a frame scenario occurs in a document, the method is not able to predict how important an event is for a scenario or how important a scenario is in a document.

The last problem we have noticed is related to the flexibility of the frame-based scenarios. Because they are defined at the conceptual level, the scenarios defined by frame relations are too generalized, and therefore they cannot capture all the events that happen in specific situations. Also, because FrameNet has a fixed number of frame scenarios, the method does not have the capability to "zoom in" or "zoom out" the event scenario space in order to extract more specific or more general event scenarios. All these limitations are surmounted by the event scenario model that we describe in the next section.

The Event Scenario Model

In recent years, the Latent Dirichlet Allocation model has been successfully applied in various applications such as topic modeling from a collection of documents (Blei, Ng, & Jordan 2003; Griffiths & Steyvers 2004), word sense disambiguation (Boyd-Graber, Blei, & Zhu 2007; Boyd-Graber & Blei 2007), object categorization from a collection of images (Sivic *et al.* 2005), and image classification into scene categories (Li & Perona 2005). In this section, we describe the event scenario model that uses the LDA model for a new application, namely finding event scenarios in a document collection.

The basic idea in the event scenario model is that documents are expressed as probabilistic mixtures of event scenarios, while each event scenario in a document has assigned a probability distribution over the events mentioned in the document. The purpose of this model is to find the best set of latent event scenarios that can explain the observed events and to make recovering these scenarios possible using statistical inference. In this model, we consider events as 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, $e{=}(e_1, e_2, \ldots, e_{N_d})$, 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 $N{=}\sum_{d=1}^{M} N_d$.

To present the event scenario model in a more formal way, we make additional notations. Assuming we have S scenarios, we indicate the assignment of an event to a scenario $s \in \{1, \ldots, S\}$ with an E-dimensional vector z. If $P(z)$ denotes the probability distribution over event scenarios z and $P(e|z)$ is the probability distribution over events e given the event scenario z, the distribution over the events 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 the event e_i given the jth event scenario and represents how significant the event e_i is for the jth scenario, while $P(z_i = j)$ is the probability that the jth scenario was chosen for the ith event from the document d. In our event scenario model, the multinomial distribution associated to each scenario, $P(e|z)$, is parameterized by an $E{\times}S$ matrix Φ such that $P(e|z = j) = \phi_e^{(j)}$ is the multinomial distribution over events for the jth scenario ($\sum_{i=1}^{E} \phi_i^{(j)} = 1$). Similarly, the distribution of event scenarios associated with 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 scenarios 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 scenario $z_i^{(d)} \sim$ Multinomial($\theta^{(d)}$).
    2. Choose an event $e_i^{(d)} \sim$ Multinomial($\phi_e^{z_i^{(d)}}$) conditioned on the scenario $z_i^{(d)}$.

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

For each document d, the model assigns the following probability:

$$P(e|\phi, \alpha) = \int P(e|\phi, \theta)P(\theta|\alpha)d\theta$$

Because the integral in this expression is intractable, several approximation techniques were proposed: mean field variational methods (Blei, Ng, & Jordan 2003), expectation propagation (Minka & Lafferty 2002), collapsed Gibbs sampling (Griffiths & Steyvers 2002), and collapsed variational inference (Teh, Kurihara, & Welling 2008).

Scenario 2		Scenario 6		Scenario 38		Scenario 41	
event	$P(e\|z=2)$	event	$P(e\|z=6)$	event	$P(e\|z=38)$	event	$P(e\|z=41)$
said	.0938	said	.1030	said	.0567	found	.0449
killed	.0285	offer	.0924	dropped	.0262	said	.0337
investigation	.0163	proposal	.0308	showed	.0218	took	.0337
bombing	.0163	closed	.0231	rose	.0174	negotiations	.0224
made	.0122	trading	.0231	demand	.0174	kidnapped	.0224
search	.0122	make	.0231	losses	.0131	promise	.0224
found	.0122	proposed	.0231	continue	.0131	dismembered	.0224
sent	.0122	acquire	.0154	expected	.0131	abandoned	.0112
ordered	.0122	transaction	.0154	declined	.0131	wrapped	.0112
discovered	.0081	sale	.0154	drop	.0131	kill	.0112
killings	.0081	comment	.0154	reported	.0087	report	.0112
arrested	.0081	own	.0154	closed	.0087	confirmed	.0112
attacks	.0081	provide	.0154	gains	.0087	hostage	.0112
evidence	.0081	talks	.0154	result	.0087	decapitated	.0112
find	.0081	respond	.0154	increased	.0087	ransom	.0112
manhunt	.0081	set	.0098	raise	.0087	kidnappings	.0112
arrests	.0081	rejected	.0091	posted	.0087	crime	.0112
bombings	.0081	agreement	.0077	earned	.0087	discover	.0112
slayings	.0081	acquisition	.0077	trend	.0087	demanded	.0112
murdered	.0081	sold	.0077	lead	.0087	cast	.0112

Table 3: Examples of learned event scenarios.

Table 3 lists four scenario examples learned by the event scenario model from an event annotated collection of news articles. For this example, the number of scenarios S was set to 50. The events from this example are related to commerce transaction, financial market, kidnapping, and war scenarios and are listed in order of their relevance to every scenario. As can be observed in this table, because news articles usually abound in reporting events, these type of events are highly related to all four scenarios.

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

Experimental Results

We trained the event scenario model on version 1.2 of TimeBank (Pustejovsky *et al.* 2003b), which is a corpus of news articles annotated with events, time expressions, and temporal relations between them. TimeBank contains 183 documents with 7935 event annotations. After removing 3 documents from the corpus because of naive event annotation, we extracted 7622 event instances corresponding to an event collection of 2549 unique events. The events in TimeBank are annotated following the TimeML specifications (Pustejovsky *et al.* 2003a) and can be expressed as verbs, nominals and adjectives.

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 order to see how well it models the data, we compared our unsupervised method against several event scenarios extracted from FrameNet and against a baseline model consisting of a smoothed unigram model.

Frame Scenario Modeling

Despite its limitations, the method that maps events on semantic frames creates valid scenarios with relevant events in every scenario. Therefore, we can evaluate how well the frame scenarios are covered by the event scenarios extracted with our model and compute a coverage score for each frame scenario. To perform this evaluation, we automatically extracted four frame scenarios from FrameNet based on the relations that exist between frames. These frame-based scenarios are listed in Table 4. In the next step, we selected all the TimeBank events that evoke frames from the frame-based scenarios following the method described in section 3. For instance, Table 5 lists all the events from the TimeBank corpus that evoke frames belonging to the COMMERCE scenario.

COMMERCE scenario	CRIME scenario
Commerce scenario, Businesses, Exchange currency, Commercial transaction, Fining, Commerce pay, Commerce sell, Commerce collect, Commerce buy, Import export	Seeking, Committing crime, Criminal investigation, Want suspect, Arrest, Surrendering, Trial, Sentencing, Appeal, Notification of charges, Entering of plea, Verdict, Questioning, Kidnapping
ATTACK scenario	**EMPLOYMENT scenario**
Attack, Defend, Hostile encounter, Fighting activity	Employment scenario, Firing, Get a job, Being employed, Quitting

Table 4: The frame scenarios used in evaluation.

As can be observed in Table 5, we derived two types of frame-based scenarios: (1) a generic scenario having semantic frames as basic elements (left column in the table),

Frame	Frame-evoking events
Commerce scenario	price pricecutting price-cutting priced prices pricing pricings cost costs
Businesses	business businesses establish established establishing establishment
Exchange currency	change changed changes changing exchange convert converted convertible
Commercial transaction	transaction transactions
Commerce pay	pay payable paying payment payments payout pays
Commerce collect	charge charged charges
Commerce buy	buy buy-back buying buy-out buyouts bought purchase purchased purchases purchasing
Fining	damage damages
Import export	export exports
Commerce sell	sale sales sell seller selling sell-off sold

Table 5: Example of frame scenario mapped on TimeBank.

and (2) a specific scenario containing the TimeBank frame-evoking events (right column in the table). Thus, we can compute a frame-based coverage score and an event-based score for each frame scenario. To compute the coverage score, we selected the best matching LDA scenario for each scenario derived from FrameNet and evaluated the percentage of events from the LDA scenarios that is needed to cover all the events from the frame-based scenarios. In this evaluation procedure, we iterated through events in order of their relevance to an LDA-inferred scenario, and compute the percentage of events from an FrameNet scenario covered at the current iteration step. The sooner it reaches 100% coverage, the better the LDA scenario models its corresponding FN scenario. For the frame-based coverage score it is sufficient to reach only one frame-evoking event to cover its corresponding frame.

Figure 3 illustrates the frame and event coverage curves for each of the four frame-based scenarios. The best coverage results are obtained for COMMERCE and EMPLOYMENT scenarios. This can be explained by the fact that the majority of the documents from TimeBank are from financial publications. Specifically, for the COMMERCE frame-based scenario, the best LDA scenario when computing the event coverage measure is the 13th scenario. This scenario requires to iterate on the first 940 (36.87%) relevant events in order to cover all the events from the frame-based scenario. In a similar way, the best LDA scenario for covering all the frames in the COMMERCE scenario requires only the first 696 (27.3%) relevant events. The frame coverage for the EMPLOYMENT scenario is even better. To cover 60% of the frames, the LDA scenario that best models the frames in the EMPLOYMENT scenario (LDA scenario 44) requires only 1.68% of its most relevant events, whereas to cover all the frames from this FN scenario, the 44th LDA scenario iterates only through 10.86% of its relevant events.

The event scenario model performs slightly worse on covering the ATTACK and CRIME scenarios. One explanation is that these scenarios are not well represented in the document collection. For example, the event *plead* is the only event that evokes the last covered frame (ENTERING OF PLEA) on the coverage curve corresponding to the CRIME scenario and appears only once in the document collection. Another ex-

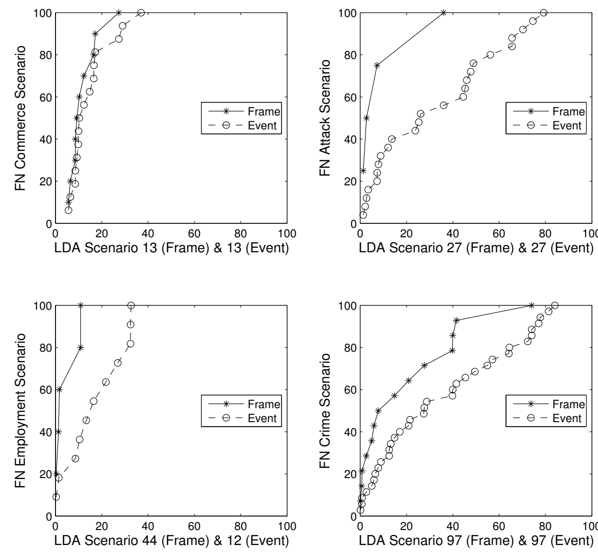

Figure 3: Coverage curves for four FN scenarios.

planation for a weak coverage is caused by the events that can evoke multiple frames. For instance, *order* is an ambiguous event that can evoke SENTENCING, REQUEST and BAIL SETTING semantic frames. Since SENTENCING is a frame in the CRIME scenario, the contribution of the *order* event to the coverage score has to be encountered as well. However, we noticed that this event is disambiguated in TimeBank corpus as evoking the REQUEST frame in most of the cases and therefore the impact of the *order* event for the CRIME scenario is diminished. We leave the study of frame-based coverage using disambiguated events to future work.

Document Modeling

In addition to frame-based scenario evaluation, we also measured the predictive power of the event scenario model by comparing it with a smoothed unigram baseline model using the perplexity score. Perplexity is a commonly used measure in language modeling and captures how surprised a model is when exposing to unseen data. Lower perplexity scores are better and imply that the model is less surprised for the unseen data. Formally, the perplexity score of a held-out test data from the event annotated corpus, $\mathcal{C}_{test}$, is defined as:

$$Perplexity(C_{test}) = exp\left(-\frac{\sum_{d=1}^{M}\log p(e_d)}{\sum_{d=1}^{M} N_d}\right)$$

where M is the number of documents in $\mathcal{C}_{test}$, $p(e_d)$ is the probability of events corresponding to a document d, and N_d is the total number of events from d. In our experimental setup, we used the 5-fold cross validation scheme and computed the perplexity of every held-out test set given the model learned on the remaining part of the data. Figure 4 illustrates the average perplexity of our LDA-based model when compared to a smoothed unigram model. In addition,

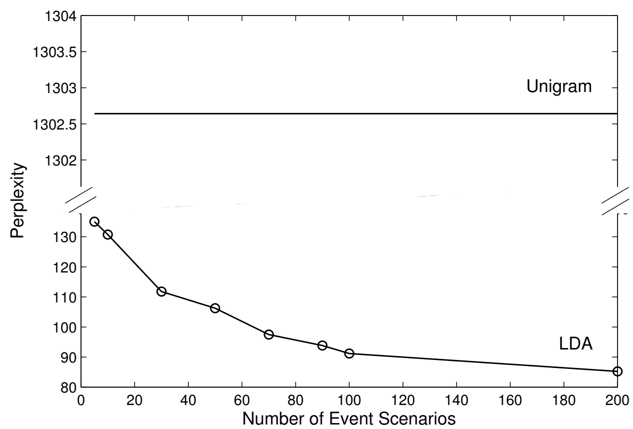

Figure 4: Perplexity results for unigram and LDA-based event models.

this plot shows the impact of the predefined event scenario numbers on the LDA-based model. As can be observed, the more scenarios extracted, the lower the average perplexity value is. When extracting 10 scenarios the LDA-based model achieves a perplexity reduction of 89.96% against the unigram model, whereas for 200 scenarios the perplexity reduction increases to 93.46%. Therefore, these results support the LDA model assumption that a document encodes multiple event scenarios to different degrees.

Conclusions

Discovering event scenarios from texts constitutes a stepping stone in the process of understanding and reasoning about texts. In this paper, we introduced a novel method for automatically extracting event scenarios from texts. The method is based on the LDA model that uses the observed events from texts to learn latent scenarios. Additionally, we introduced a method for evaluating event scenarios by comparing them with hand-annotated frame scenarios.

In our evaluations, we proved that an LDA-based approach is a suitable model for learning event scenarios even when it is trained on a small collection of event annotated documents. When comparing the event scenarios extracted by this method with several predefined scenarios derived from FrameNet, the results show a good coverage for those scenarios that are representative for the document collection. Our experiments also show a better generalization score for the LDA-based event model when it is compared to a smoothed unigram event model.

References

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

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

Boyd-Graber, J., and Blei, D. 2007. PUTOP: Turning Predominant Senses into a Topic Model for Word Sense Disambiguation. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, 277–281.

Boyd-Graber, J.; Blei, D.; and Zhu, X. 2007. A Topic Model for Word Sense Disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 1024–1033.

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

Griffiths, T., and Steyvers, M. 2002. A probabilistic approach to semantic representation. In *Proceedings of the 24th Annual Conference of the Cognitive Science Society*.

Griffiths, T., and Steyvers, M. 2004. Finding Scientific Topics. In *Proceedings of the National Academy of Sciences*.

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

Li, F.-F., and Perona, P. 2005. A Bayesian Hierarchical Model for Learning Natural Scene Categories. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, 524–531.

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

Minka, T., and Lafferty, J. 2002. Expectation-propagation for the Generative Aspect Model. In *Proceedings of the Uncertainty in Artificial Intelligence*.

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

Pustejovsky, J.; Hanks, P.; Sauri, R.; See, A.; Day, D.; Ferro, L.; Gaizauskas, R.; Lazo, M.; Setzer, A.; and Sundheim, B. 2003b. The TimeBank Corpus. In *Corpus Linguistics*, 647–656.

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

Sivic, J.; Russell, B.; Efros, A.; Zisserman, A.; and Freeman, W. 2005. Discovering Object Categories in Image Collections. In *Proceedings of the ICCV-05*.

Teh, Y. W.; Kurihara, K.; and Welling, M. 2008. Collapsed Variational Inference for HDP. In *Advances in Neural Information Processing Systems*.