Unsupervised Event Coreference Resolution

Cosmin Adrian Bejan*
Vanderbilt University

Sanda Harabagiu**
University of Texas at Dallas

The task of event coreference resolution plays a critical role in many natural language processing applications such as information extraction, question answering, and topic detection and tracking. In this article, we describe a new class of unsupervised, nonparametric Bayesian models with the purpose of probabilistically inferring coreference clusters of event mentions from a collection of unlabeled documents. In order to infer these clusters, we automatically extract various lexical, syntactic, and semantic features for each event mention from the document collection. Extracting a rich set of features for each event mention allows us to cast event coreference resolution as the task of grouping together the mentions that share the same features (they have the same participating entities, share the same location, happen at the same time, etc.).

Some of the most important challenges posed by the resolution of event coreference in an unsupervised way stem from (a) the choice of representing event mentions through a rich set of features and (b) the ability of modeling events described both within the same document and across multiple documents. Our first unsupervised model that addresses these challenges is a generalization of the hierarchical Dirichlet process. This new extension presents the hierarchical Dirichlet process's ability to capture the uncertainty regarding the number of clustering components and, additionally, takes into account any finite number of features associated with each event mention. Furthermore, to overcome some of the limitations of this extension, we devised a new hybrid model, which combines an infinite latent class model with a discrete time series model. The main advantage of this hybrid model stands in its capability to automatically infer the number of features associated with each event mention from data and, at the same time, to perform an automatic selection of the most informative features for the task of event coreference. The evaluation performed for solving both within- and cross-document event coreference shows significant improvements of these models when compared against two baselines for this task.

* Department of Biomedical Informatics, School of Medicine, Vanderbilt University, 400 Eskind Biomedical Library, 2209 Garland Avenue, Nashville, TN 37232, USA. E-mail: adi.bejan@vanderbilt.edu.
** Human Language Technology Research Institute, Department of Computer Science, University of Texas at Dallas, 800 West Campbell Road, Richardson, TX 75080, USA. E-mail: sanda@hlt.utdallas.edu.

Submission received: 6 February 2012; revised submission received: 9 May 2013; accepted for publication: 28 June 2013.

doi:10.1162/COLI_a_00174

1. Introduction

Event coreference resolution consists of grouping together the text expressions that refer to real-world events (also called **event mentions**) into a set of clusters such that all the mentions from the same cluster correspond to a unique event. The problem of event coreference is not new. It was originally studied in philosophy, where researchers tried to determine when two events are identical and when they are different. One relevant theory in this direction was proposed by Davidson (1969), who argued that two events are identical if they have the same causes and effects. Later on, a different theory was proposed by Quine (1985), who considered that each event is associated with a physical object (which is well defined in space and time), and therefore, two events are identical if their corresponding objects have the same spatiotemporal location. According to Malpas (2009), in the same year, Davidson abandoned his suggestion to embrace the Quinean theory on event identity (Davidson 1985).

Resolving event coreference is an essential requirement for many natural language processing (NLP) applications. For instance, in topic detection and tracking, event coreference resolution is required in order to identify new seminal events in broadcast news that have not been mentioned before (Allan et al. 1998). In information extraction, event coreference information was used for filling predefined template structures from text documents (Humphreys, Gaizauskas, and Azzam 1997). In question answering, a novel method of mapping event structures was used in order to provide answer justification (Narayanan and Harabagiu 2004). The same idea of mapping event structures was used in a graph-matching approach for enhancing textual entailment (Haghighi, Ng, and Manning 2005). Event coreference information was also used for detecting contradictions in text (de Marneffe, Rafferty, and Manning 2008).

Previous NLP approaches for solving event coreference relied on supervised learning methods that explore various linguistic properties in order to decide if a pair of event mentions is coreferential or not (Humphreys, Gaizauskas, and Azzam 1997; Bagga and Baldwin 1999; Ahn 2006; Chen and Ji 2009; Chen, Su, and Tan 2010b). In spite of being successful for a particular labeled corpus, in general, these pairwise models are dependent on the domain or language that they are trained on. For instance, in order to adapt a supervised system to run over a collection of documents written in a different language or belonging to a different domain of interest, at least a minimal annotation effort needs to be performed (Daumé III 2007). Furthermore, because these models are dependent on local pairwise decisions, they are unable to capture a global event distribution at the topic- or document-collection level.

To address these limitations, we departed from the idea of using supervised approaches for event coreference resolution and explored how a new class of unsupervised, nonparametric Bayesian models can be used to probabilistically infer coreference clusters of event mentions from a collection of unlabeled documents. In addition, because an event can be mentioned multiple times in a document collection and its mentions may occur both in the same document or across multiple documents, we designed our unsupervised models to solve the two subproblems of **within-document** and **cross-document** event coreference resolution. In order to evaluate the unsupervised models for these two subproblems, we annotated a new data set encoding both within- and cross-document event coreference information.

Besides our contribution of using unsupervised methods to solve within- and cross-document event coreference, in this article we present novel Bayesian models that provide a more flexible framework for representing data than current models. By starting from the generic problem of clustering observable linguistic objects (i.e., event

mentions) encoded into a large collection of text documents where the clusters (i.e., events) can be shared across documents, we devised our unsupervised models such that they provide solutions to the following four desiderata:

1) We prefer the number of clusters (denoted by K) to be probabilistically inferred from data rather than to be assigned to an a priori fixed value. This desideratum of allowing K to be a free parameter in the Bayesian models devised for our problem constitutes a more realistic approach because, in general, document collections encode an unspecified number of latent linguistic structures.

2) We redefine the task of finding clusters of mentions that refer to the same events as the task of identifying those mentions that share the same **event participants** and the same **event properties**. For example, the same entity must participate in all the event mentions that are coreferential; also, all the coreferential mentions must have the same spatiotemporal location. These characteristics extracted for each event mention from text are also called **linguistic features** and, in general, the event mentions corresponding to each of these clusters are characterized by a large set of features. Because of this, we desire that the generative process associated with each Bayesian model to automatically adapt every time a new feature is added in the feature extraction phase.

3) Although each event mention is represented as a feature-rich linguistic object, there is no guarantee that all the features that describe event mentions have a positive impact for the task of event coreference. Some of these features may be redundant or may increase the complexity of the Bayesian models solving this task and, consequently, they may contribute to lowering the overall performance of event coreference. To address these problems, we wish to incorporate into the Bayesian models a feature selection mechanism that is able to automatically build a set of the most salient features from the initial feature set such that only these salient features will participate in the process of clustering event mentions. In this regard, we assume that a feature is salient if it corresponds to a large number of samples in the generative process. We denote the size of the salient feature set by M. Furthermore, in spite of the fact that the initial feature space describing event mentions can have an unbounded number of features, we want the set of salient features to be finite (i.e., M–finite) at any given point in time during the generative process corresponding to each Bayesian model.

4) Finally, we also want our Bayesian models to capture the structural dependencies of the observable objects. In this way, the models can take advantage of the sequential order in which the event mentions are generated inside each document.

We believe that these four desiderata constitute a more natural approach for clustering complex linguistic objects from a large collection of documents and relax many of the constraints imposed in the current clustering tasks.

It is worth pointing out that the generic problem described here can be instantiated by tasks not only from the area of computational linguistics, but also from other research areas as well. For instance, in biomedical informatics, clinical researchers can use the new Bayesian models to perform studies over various cohorts of patients. In this configuration, the observations to be clustered correspond to patients, and the features associated with the patients can be extracted from clinical reports or can be represented by structured clinical information (e.g., white blood cells, temperature,

heart rate, respiratory rate, sputum culture). Another instance of the generic problem described here is from data mining. In this domain, clustering tasks can be performed over structured information stored in large tables (e.g., products, restaurants, hotels). For this type of problem, each object is associated with a row in a table and the features correspond to table columns.

2. Related Work

Unlike entity coreference resolution, event coreference resolution is a relatively less-studied task. One rationale is that events are expressed in many more varied linguistic constructs. For example, event mentions are typically predications that require more complex lexico-semantic processing, and furthermore, the capability of extracting features that characterize them has been available only since semantic parsers based on PropBank (Palmer, Gildea, and Kingsbury 2005) and FrameNet (Baker, Fillmore, and Lowe 1998) corpora have been developed. In contrast, entity coreference resolution has been intensively studied and many successful techniques for identifying mention clusters have been developed (Cardie and Wagstaf 1999; Haghighi and Klein 2009; Stoyanov et al. 2009; Haghighi and Klein 2010; Raghunathan et al. 2010; Rahman and Ng 2011).

Even if entity coreference resolution has received much attention from the computational linguistic researchers, there is only limited work that incorporates event-related information to solve entity coreference, typically by considering the verbs that are present in the context of a referring entity as features. For instance, Haghighi and Klein (2010) include the governor of the head of nominal mentions as features in their model. Rahman and Ng (2011) used event-related information by looking at which semantic role the entity mentions can have and the verb pairs of their predicates. More recently, Lee et al. (2012) proposed an approach to jointly model event and entity coreference by allowing information from event coreference to help entity coreference, and the other way around. Their supervised method uses a high-precision entity resolution method based on a collection of deterministic models (called sieves) to produce both entity and event clusters that are optimally merged using linear regression. A similar technique that treated entity and event coreference resolution jointly was reported in He (2007) using narrative clinical data.

Research that aimed at resolving only event coreference was initiated by the template merging task required in MUC evaluations and was primarily focused on scenario-specific events (Humphreys, Gaizauskas, and Azzam 1997; Bagga and Baldwin 1999). More recently, various supervised approaches using a mention-pair probabilistic framework (Ahn 2006), spectral graph clustering (Chen and Ji 2009), and tree kernel–based methods (Chen, Su, and Tan 2010b) have been used to solve event coreference. Tree kernel–based methods have also been used to solve a special case of event coreference resolution called event pronoun resolution (Chen, Su, and Tan 2010a; Kong and Zhou 2011). To the best of our knowledge, the framework for solving event coreference presented in this article, extending the approach reported in Bejan and colleagues (Bejan et al. 2009; Bejan and Harabagiu 2010), is the only line of research on event coreference resolution that uses fully unsupervised methods and is based on Bayesian models.

Over the past years, Bayesian models have been extensively used for the purpose of solving similar problems or subproblems of the generic problem presented in the previous section. In 2003, Blei, Ng, and Jordan proposed a parametric approach, called **latent Dirichlet allocation** (LDA), for automatically learning probability distributions of words corresponding to a specific number of latent classes (or **topics**) from a large

collection of text documents. In this latent class model, documents are expressed as probabilistic mixtures of topics, while each topic has assigned a multinomial distribution over the words from the entire document collection. This approach also uses an exchangeability assumption by modeling the documents as bags of words. The LDA model and variations of it have been used in many applications such as topic modeling (Blei, Ng, and Jordan 2003; Griffiths and Steyvers 2004), word sense disambiguation (Boyd-Graber, Blei, and Zhu 2007), object categorization from a collection of images (Sivic et al. 2005, 2008), image classification into scene categories (Li and Perona 2005), discovery of event scenarios from text documents (Bejan 2008; Bejan and Harabagiu 2008b), and attachment of attributes to a concept ontology (Reisinger and Paşca 2009). The LDA model, although attractive, has the disadvantage of requiring a priori knowledge regarding the number of latent classes.

A more suitable approach for solving our problem is the **hierarchical Dirichlet process** (HDP) model described in Teh et al. (2006). Like LDA, this model considers problems that involve groups of data, where each observable object is sampled from a mixture model and each mixture component is shared across groups. However, the HDP mixture model is a nonparametric generalization of LDA that is also able to automatically infer the number of clustering components K (the first desideratum for our problem). It consists of a set of **Dirichlet processes** (DPs) (Ferguson 1973), in which each DP is associated with a group of data. In addition, these DPs are coupled through a common random base measure which is itself distributed according to a DP. Due to the fact that a DP provides a nonparametric prior for the number of classes K, the HDP setting allows for this number to be unbounded in each group. More recently, various other applications have been proposed to improve the existing HDP inference algorithms (Wang, Paisley, and Blei 2011; Bryant and Sudderth 2012). HDP has been used in a wide variety of applications such as maneuvering target tracking (Fox, Sudderth, and Willsky 2007), visual scene analysis (Sudderth et al. 2008), information retrieval (Cowans 2004), entity coreference resolution (Haghighi and Klein 2007; Ng 2008), event coreference resolution (Bejan et al. 2009; Bejan and Harabagiu 2010), word segmentation (Goldwater, Griffiths, and Johnson 2006), and construction of stochastic context-free grammars (Finkel, Grenager, and Manning 2007; Liang et al. 2007).

Although infinite latent class models like HDP have the advantage of automatically inferring the number of categorical outcomes K, they are still limited in representing feature-rich objects. Specifically, in their original form, they are not able to model the data such that each observable object can be generated from a combination of multiple features. For example, in HDP, each data point is represented only by its corresponding word. For this reason, we built new Bayesian models on top of already-existing models with the main goal of providing a more flexible framework for representing data. The first model extends the HDP model such that it takes into account additional linguistic features associated with event mentions. This extension is performed by using a conditional independence assumption between the observed random variables corresponding to object features. Thus, instead of considering as features only the words that express the event mentions (which is the way an observable object is represented in the original HDP model), we devised an HDP extension that is also able to represent features such as location, time, and agent for each event mention. This extension was inspired from the fully generative Bayesian model proposed by Haghighi and Klein (2007). However, Haghighi and Klein's model was strictly customized for the task of entity coreference resolution. As also noted in Ng (2008) and Poon and Domingos (2008), whenever new features need to be considered in Haghighi and Klein's model, the extension becomes a challenging task. Also, Daumé III and Marcu (2005) performed

related work in this direction by proposing a generative model for solving supervised clustering problems.

As an alternative to the HDP model, an important extension of latent class models that are able to represent feature-rich objects is the **Indian buffet process** (IBP) model presented in Griffiths and Ghahramani (2005). The IBP model defines a distribution over infinite binary sparse matrices that can be used as a nonparametric prior on the features associated with observable objects. Moreover, extensions of this model were considered in order to provide a more flexible approach for modeling the data. For example, the **Markov Indian buffet process** (mIBP) (Van Gael, Teh, and Ghahramani 2008) was defined as a distribution over an unbounded set of binary Markov chains, where each chain can be associated with a binary latent feature that evolves over time according to Markov dynamics. Also, the **phylogenetic Indian buffet process** (pIBP) (Miller, Griffiths, and Jordan 2008) was created as a non-exchangeable, nonparametric prior for latent feature models, where the dependencies between objects were expressed as tree structures. Examples of applications that utilized these models are: identification of protein complexes (Chu et al. 2006), modeling of dyadic data (Meeds et al. 2006), modeling of choice behavior (Görür, Jäkel, and Rasmussen 2006), and event coreference resolution (Bejan et al. 2009; Bejan and Harabagiu 2010).

Our extension of the HDP model still does not fulfill all the desiderata for the generic problem introduced in Section 1. It still requires a mechanism to automatically select a finite set of salient features that will be used in the clustering process (third desideratum) as well as a mechanism for capturing the structural dependencies between objects (fourth desideratum). To overcome these limitations, we created two additional models. First, we incorporated the mIBP framework into our HDP extension to create the **mIBP–HDP** model. And second, we coupled an infinite latent feature model with an infinite latent class model into a new discrete time series model. For the infinite latent feature model, we chose the **infinite factorial hidden Markov model** (iFHMM) (Van Gael, Teh, and Ghahramani 2008) coupled with the mIBP mechanism in order to represent the latent features as an infinite set of parallel Markov chains; for the infinite latent class model, we chose the **infinite hidden Markov model** (iHMM) (Beal, Ghahramani, and Rasmussen 2002). We call this new hybrid the **iFHMM–iHMM** model.

2.1 Contribution

This article represents an extension of our previous work on unsupervised event coreference resolution (Bejan et al. 2009; Bejan and Harabagiu 2010). In this work, we present more details on the problem of solving both within- and cross-document event coreference as well as describe a generic framework for solving this type of problem in an unsupervised way. As data sets, we consider three different resources, including our own corpus (which is the only corpus available that encodes event coreference annotations across and within documents). In the next section, we provide additional information on how we performed the annotation of this corpus. Another major contribution of this article is an extended description of the unsupervised models for solving event coreference. In particular, we focused on providing further explanations about the implementation of the mIBP framework as well as its integration into the HDP and iHMM models. Finally, in this work, we significantly extended the experimental results section, which also includes a novel set of experiments performed over the OntoNotes English corpus (LDC-ON 2007).

3. Event Coreference Data Sets

Because our nonparametric Bayesian models are also unsupervised, they do not require the data set(s) on which they are trained to be annotated with event coreference information. The only requirement for them to infer coreference clusters of event mentions is to have the observable objects (i.e., the event mentions) identified in the order they occur in the documents as well as to have all the linguistic features associated with these objects extracted. However, in order to see how well these models perform, we need to compare their results with manually annotated clusters of event mentions. For this purpose, we evaluated our models on three different data sets annotated with event coreference information.

The first data set was used for the event coreference evaluations performed in the automatic content extraction (ACE) task (LDC-ACE 2005). This resource contains only a restricted set of event types such as LIFE, BUSINESS, CONFLICT, and JUSTICE. As a second data set, we used the OntoNotes English corpus (release 2.0), a more diverse resource that provides a larger coverage of event (and entity) annotations. The utilization of the ACE and OntoNotes corpora for evaluating our event coreference models is, however, limited because these resources provide only within-document event coreference annotations. For this reason, as a third data set, we created the **EventCorefBank** (ECB) corpus[1] to increase the diversity of event types and to be able to evaluate our models for both within- and cross-document event coreference resolution. Recently, Lee et al. (2012) extended the EventCorefBank corpus with entity coreference information and additional annotations of event coreference.

One important step in the creation process of the ECB corpus consists of finding sets of related documents that describe the same **seminal event**[2] such that the annotation of coreferential event mentions across documents is possible. In this regard, we searched the Google News archive[3] for various topics whose description contains keywords such as *commercial transaction*, *attack*, *death*, *sports*, *announcement*, *terrorist act*, *election*, *arrest*, *natural disaster*, and so on, and manually selected sets of Web documents describing the same seminal event for each of these topics. In a subsequent step, for every Web document, we automatically tokenized and split the textual content into sentences, and saved the preprocessed data in a uniquely identified text file. Next, we manually annotated a limited set of events in each text file in accordance with the TimeML specification (Pustejovsky et al. 2003a). To mark the event mentions and the coreferential relations between them we utilized the Callisto[4] and Tango[5] annotation tools, respectively. Additional details regarding the annotation process for creating the ECB resource are described in Bejan and Harabagiu (2008a).

Several annotation fragments from ECB are shown in Example (1). In this example, event mentions are annotated at the sentence level, sentences are grouped into documents, and the documents describing the same seminal event are organized into topics. The topics shown in Example (1) describe the seminal event of arresting sea pirates by a

1 The ECB corpus is available at `http://www.hlt.utdallas.edu/~ady/data/ECB1.0.tar.gz`.

2 A seminal event in a document is the event that triggers the topic of the document and has interconnections with the majority of events from its surrounding textual context. Furthermore, the set of documents describing the same seminal event defines a **topic**. A more detailed description of seminal events can be found in topic detection and tracking literature (Allan 2002).

3 `http://news.google.com`.

4 `http://callisto.mitre.org`.

5 Tango is a tool designed for annotating relations between the event mentions encoded in the TimeML format and is available at `http://timeml.org/site/tango/tool.html`.

Topic 12

Document 3

s_1: *In another anti-piracy operation, Navy warship on Saturday repulsed an attack on a merchant vessel in the Gulf of Aden and* [***nabbed***]$_{em_1}$ *23 Somali and Yemeni sea brigands.*

Topic 43

Document 3

s_4: *AMD agreed to* [***buy***]$_{em_2}$ *Markham, Ontario-based ATI for around $5.4 billion in cash and stock, the companies announced Monday.*

s_5: *The* [***acquisition***]$_{em_3}$ *would turn AMD into one of the world's largest providers of graphics chips.*

Topic 44

Document 2

s_1: *Hewlett-Packard is negotiating to* [***buy***]$_{em_4}$ *technology services provider Electronic Data Systems.*

. . .

s_8: *With a market value of about $115 billion, HP could easily use its own stock to finance the* [***purchase***]$_{em_5}$.

s_9: *If the* [***deal***]$_{em_6}$ *is completed, it would be HP's biggest* [***acquisition***]$_{em_7}$ *since it* [***bought***]$_{em_8}$ *Compaq Computer Corp. for $19 billion in 2002.*

Document 5

s_2: *Industry sources have confirmed to eWEEK that Hewlett-Packard will* [***acquire***]$_{em_9}$ *Electronic Data Systems for about $13 billion.*

Topic 55

Document 2

s_2: *Despite his* [***arrest***]$_{em_{10}}$ *on suspicion of driving under the influence yesterday, Chargers receiver Vincent Jackson will play in Sunday's AFC divisional playoff game at Pittsburgh.*

Document 3

s_2: *San Diego Chargers receiver Vincent Jackson was* [***arrested***]$_{em_{11}}$ *on suspicion of drunk driving on Tuesday morning, five days before a key NFL playoff game.*

s_3: *Police* [***apprehended***]$_{em_{12}}$ *Jackson in San Diego at 2:30 a.m. and booked him for the misdemeanor before his release.*

Example 1
Examples of event mention annotations.

Navy warship (topic 12), the event of buying ATI by AMD (topic 43), the event of buying EDS by HP (topic 44), and the event of arresting a reputed football player (topic 55). When taken out of context, the event mentions annotated in this example refer only to two **generic events**: *arrest* and *buy*. On the other hand, when these mentions are contextually associated with the event properties expressed in Example (1), five **individuated events** can be distinguished: e_1={em_2, em_3}, e_2={em_{4-7}, em_9}, e_3={em_8}, e_4={em_1}, and e_5={em_{10}, em_{11}, em_{12}}. For example, em_{4-7} are event mentions referring to the same real event (of buying EDS by HP), whereas em_2 (*buy*) and em_4(*buy*) correspond to different

individuated events because they have a different AGENT (i.e., BUYER(em_2)=*AMD* is different from BUYER(em_4)=*HP*). Similarly, the mentions em_1(*nabbed*) and em_{12} (*apprehended*) do not corefer because they correspond to different spatial and temporal locations (e.g., LOCATION(em_1)=*Gulf of Aden* is different from LOCATION(em_{12})=*San Diego*).

This organization of event mentions leads to the idea of creating an event hierarchy as the one illustrated in Figure 1. Specifically, this figure depicts the hierarchy of the events described in Example (1). In this hierarchy, the nodes on the first level correspond to event mentions (e.g., em_{11} corresponds to *arrested*), the nodes on the second level correspond to individuated events (e.g., e_5 subsumes all the event mention nodes that refer to the arrest of Vincent Jackson), and, finally, the nodes on the third level correspond to generic events (e.g., the node *arrest* contains all possible arrest events). In this article, our focus is to discover the nodes on the second level of this hierarchy.

As can be seen from Example (1), solving the event coreference problem poses many interesting challenges. For instance, in order to solve the coreference chain of event mentions that refer to the event e_2, we need to take into account the following issues: (i) a coreference chain can encode both within- and cross-document coreference information; (ii) two mentions from the same chain can have different word classes (e.g., em_4(*buy*)–verb, em_5(*purchase*)–noun); (iii) not all the mentions from the same chain are synonymous (e.g., em_4(*buy*) and em_9(*acquire*)), although a semantic relation might exist between them (e.g., in WordNet [Fellbaum 1998], the genus of *buy* is *acquire*); (iv) not all the properties associated with an event mention are expressed in text (e.g., all the properties of em_5(*purchase*) are omitted). In Section 7, we discuss additional challenges of the event coreference problem that are not observed in Example (1).

4. Linguistic Features for Event Coreference Resolution

The main idea for solving event coreference is to identify the event mentions (from the same or different documents) that share the same characteristics (e.g., all the mentions in a cluster convey the same meaning in text, have the same participants, and happen in the same space and temporal location). Moreover, finding clusters of event mentions that share the same characteristics is identical to finding clusters of mention features that correspond to the same real event. For instance, Figure 2 depicts five clusters of linguistic features that characterize the five individuated events from Example (1). As can be observed, each individuated event corresponds to a subset of features that are usually common to all the mentions referring to it. For this purpose, we extracted various linguistic features associated with each event mention from the ACE, OntoNotes, and ECB corpora.

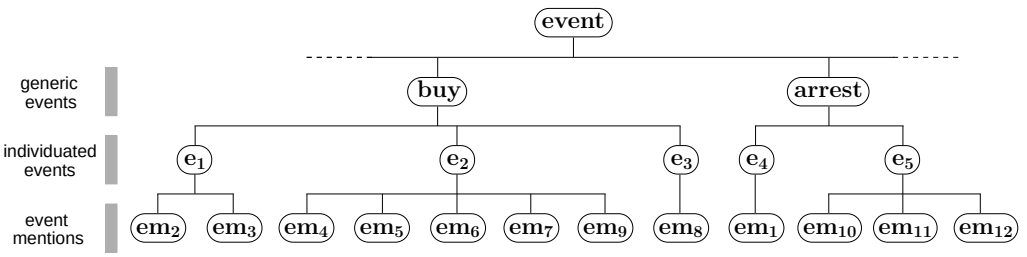

Figure 1
Fragment from the event hierarchy.

Before describing in detail all the categories of linguistic features considered for solving event coreference, we would like to emphasize that we make a clear distinction between the notions of **feature type** and **feature value** throughout this article. A feature type is represented by a characteristic that can be extracted with a specific methodology and is associated with at least two feature values. For instance, the feature values corresponding to the feature type WORD consist of all the distinct words extracted from a given data set. In order to differentiate between the same values of different feature types, we inserted to the notation of each feature value the name of its corresponding feature type (e.g., WORD:*play*).

4.1 Lexical Features (LF)

We capture the lexical context of an event mention by extracting the following features: the head word (HW), the lemmatized head word (HL), the lemmatized left and right words surrounding the mention (LHL, RHL), and the HL features corresponding to the left and right mentions (LHE, RHE). For instance, the lexical features extracted for the event mention em_8(*bought*) from our example are HW:*bought*, HL:*buy*, LHL:*it*, RHL:*Compaq*, LHE:*acquisition*, and RHE:*acquire*.

4.2 Class Features (CF)

This category of features aims to group mentions into several types of classes: the part-of-speech of the HW feature (POS), the word class of the HW feature (HWC), and the event class of the mention (EC). The HWC feature type is associated with the following four feature values: VERB, NOUN, ADJECTIVE, and OTHER. As feature values for the EC feature type, we consider the seven event classes defined in the TimeML specification language (Pustejovsky et al. 2003a): OCCURRENCE, PERCEPTION, REPORTING, ASPECTUAL, STATE, I_ACTION, and I_STATE. To extract all these event classes for all the event mentions, we used an event identifier trained on the TimeBank corpus (Pustejovsky et al. 2003b), a linguistic resource encoding temporal elements such as events, time expressions, and temporal relations. More details about this event identifier are described in Bejan (2007).

4.3 WordNet Features (WF)

In our efforts to create clusters of attributes corresponding to event mentions as close as possible to the true attribute clusters of the individuated events, we built two sets of word clusters using the entire lexical information from the WordNet database. After

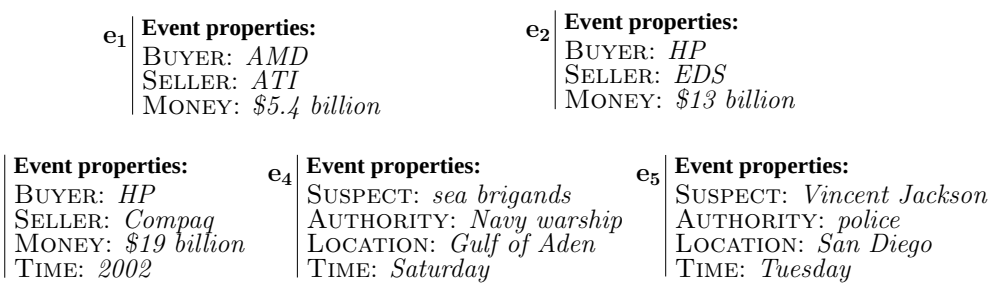

Figure 2
Linguistic features associated with the individuated events encoded in Example (1).

creating these sets of clusters, we associated each event mention with only one cluster from each set. For the first set, we used the transitive closure of the WordNet SYNONYMOUS relation to form clusters with all the words from WordNet (WNS). For instance, the verbs *buy* and *purchase* correspond to the same cluster ID because there exist a chain of SYNONYMOUS relations between them in WordNet. For the second set, we considered as grouping criteria the categorization of words from the WordNet lexicographer's files (WNL). In addition, for each word that is not represented in WordNet, we created a new cluster ID in each set of clusters.

4.4 Semantic Features (SF)

To extract features that characterize participants and properties of event mentions, we used the semantic parser described in Bejan and Hathaway (2007). One category of semantic features that we identified for event mentions is the **predicate argument structures** encoded in the PropBank annotations (Palmer, Gildea, and Kingsbury 2005). The predicate argument structures in PropBank are represented by events (or verbs) and by the semantic roles (or **predicate arguments**) associated with these events. For example, ARG0 annotates a specific type of semantic role which represents the AGENT, DOER, or ACTOR of a specific event. Another argument is ARG1, which plays the role of the PATIENT, THEME, or EXPERIENCER of an event. In Example (1), for instance, the predicate arguments associated with the event mention em_8(*bought*) are ARG0:[*it*], ARG1:[*Compaq Computer Corp.*], ARG3:[*for $19 billion*], and ARG-TMP:[*in 2002*].

Event mentions are not only expressed as verbs in text, but they also can occur as nouns and adjectives. Therefore, for a better coverage of semantic features, we also used the semantic annotations encoded in the FrameNet corpus (Baker, Fillmore, and Lowe 1998). FrameNet annotates word expressions capable of evoking conceptual structures, or **semantic frames**, which describe specific situations, objects, or events. The semantic roles associated with a word in FrameNet, or **frame elements**, are locally defined for the semantic frame evoked by the word. In general, the words annotated in FrameNet are expressed as verbs, nouns, and adjectives.

To preserve the consistency of the semantic role features, we aligned the frame elements to the predicate arguments by running the PropBank semantic parser on the manual annotations from FrameNet as well as running the FrameNet parser on the PropBank annotations. Moreover, to obtain a better alignment for each semantic role, we ran both parsers on a large amount of unlabeled text. The result of this process is a map with all frame elements statistically aligned to all predicate arguments. For instance, in 99.7% of the cases the frame element BUYER of the semantic frame COMMERCE BUY is mapped to ARG0, and in the remaining 0.3% of the cases to ARG1. Additionally, we used this map to create a more general semantic feature that assigns a frame element label to each predicate argument. Examples of semantic features for the em_8 mention are ARG0:BUYER, ARG1:GOODS, ARG3:MONEY, and ARG-TMP:TIME.

Another two semantic features used in our experiments are: (1) the semantic frame (FR) evoked by every mention in the data set, since in general, frames are able to capture properties of generic events (Lowe, Baker, and Fillmore 1997); and (2) the WNS feature applied to the head word of every semantic role (e.g., WSARG0, WSARG1).

4.5 Feature Combinations (FC)

We also explored various combinations of the given features. For instance, the feature resulting from the combination of the HW and HWC feature types for em_8(*bought*) in

Example (1) is HW+HWC:*bought*+VERB. Examples of additional feature combinations we experimented with are HL+FR, HW+POS, FR+POS+EC, FE+ARG1, and so forth.

5. Finite Feature Models

In this section, we first present HDP, a nonparametric Bayesian model that is capable of clustering objects based on one feature type (i.e., WORD); then, we introduce a novel extension of this model that describes an algorithm for clustering objects characterized by multiple feature types.

The HDP models take as input a collection of I documents, where each document i has J_i event mentions. Each event mention is characterized by L feature types (FT), and each feature type is represented by a finite vocabulary of feature values (*fv*). For example, the feature values extracted from an event coreference data set and associated with the feature type HW constitute all possible head words of the event mentions annotated in the data set. Therefore, we can represent the observable properties of an event mention as a vector of pairs $\langle(\text{FT}_1:fv_{1i}),\ldots,(\text{FT}_L:fv_{Li})\rangle$, where each feature value index i ranges in the feature value space of its corresponding feature type. In the description of these models, we also consider $\mathbf{Z}$: the set of indicator random variables for indices of events (i.e., an array of size equal with the number of event mentions in the document collection where $Z_{i,j}$ represents the event index of the event mention j from the document i); ϕ_z: the set of parameters associated with an event z; ϕ: a notation for all model parameters; and $\mathbf{X}$: a notation for all random variables that represent observable features. As already introduced in Section 1, we denote by K the total number of latent events.

Given a document collection annotated with event mentions, the goal is to find the best assignment of event indices $\mathbf{Z}^*$, which maximize the posterior probability $P(\mathbf{Z}|\mathbf{X})$. In a Bayesian approach, this probability is computed by integrating out all model parameters:

$$P(\mathbf{Z}|\mathbf{X}) = \int P(\mathbf{Z},\phi|\mathbf{X})d\phi = \int P(\mathbf{Z}|\mathbf{X},\phi)P(\phi|\mathbf{X})d\phi \quad (2)$$

5.1 The HDP$_{1f}$ Model

The one feature model, denoted here as HDP$_{1f}$, constitutes the simplest representation of an HDP model. In this model, depicted graphically in Figure 3(a), the observable components are characterized by only one feature type (e.g., the head lemma corresponding to each event mention). The distribution over events associated with each document, β, is generated by a Dirichlet process with a concentration parameter $\alpha>0$. Because this setting enables a clustering of event mentions at the document level, it is desirable that events be shared across documents and the number of events, K, be inferred from data. To ensure this flexibility, a global nonparametric DP prior with a hyperparameter γ and a global base measure H can be considered for β (Teh et al. 2006). The global distribution drawn from this DP prior, denoted as β_0 in Figure 3(a), encodes the event mixing weights. Thus, the same global events are used for each document, but each event has a document specific distribution β_i that is drawn from a DP prior centered on β_0.

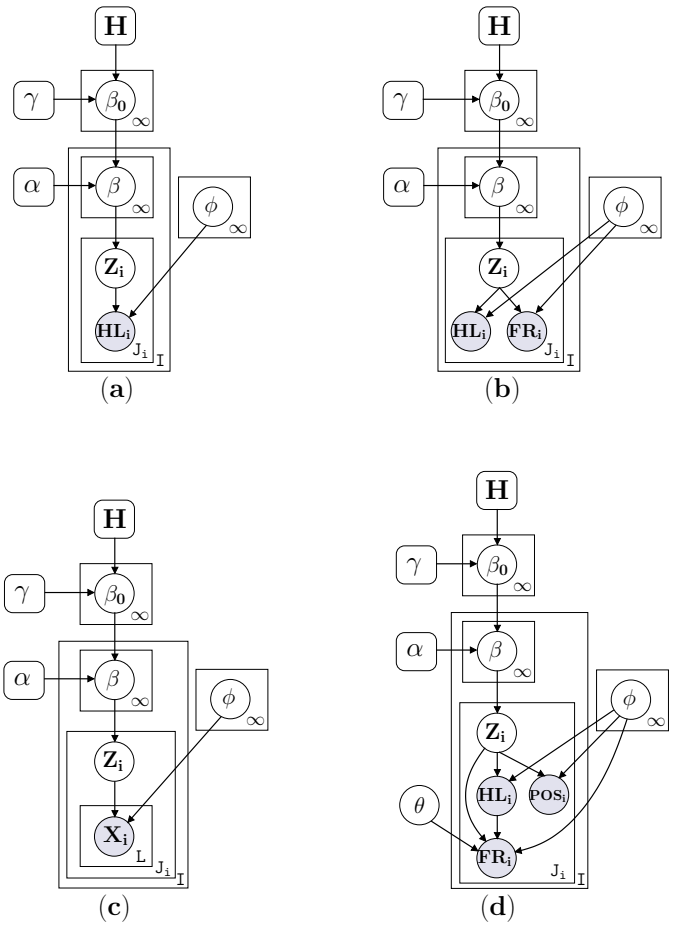

Figure 3
Graphical representation of four HDP models. Each node corresponds to a random variable. In particular, shaded nodes denote observable variables. Each rectangle captures the replication of the structure it contains. The number of replications is indicated in the bottom-right corner of the rectangle. The model depicted in (a) is an HDP model using one feature type; the model in (b) employs the HL and FR feature types; (c) illustrates a flat representation of a limited number of feature types in a generalized framework (henceforth, HDP_{flat}); and (d) captures a simple example of structured network topology of three feature types (henceforth, HDP_{struct}).

To infer the true posterior probability of $P(\mathbf{Z}|\mathbf{X})$, we followed Teh et al. (2006) and used a **Gibbs sampling algorithm** (Geman and Geman 1984) based on the direct assignment sampling scheme. In this sampling scheme, the β and ϕ parameters are integrated out analytically. The formula for sampling an event index for mention j from document i, $Z_{i,j}$, is given by:[6]

$$P(Z_{i,j} \mid \mathbf{Z}^{-i,j}, \mathbf{HL}) \propto P(Z_{i,j} \mid \mathbf{Z}^{-i,j}) P(HL_{i,j} \mid \mathbf{Z}, \mathbf{HL}^{-i,j}) \tag{3}$$

where $HL_{i,j}$ is the head lemma of event mention j from document i.

6 $\mathbf{Z}^{-i,j}$ represents a notation for $\mathbf{Z} - \{Z_{i,j}\}$.

First, in the generative process of an event mention, an event index z is sampled by using a mechanism that facilitates sampling from a prior for infinite mixture models called the **Chinese restaurant franchise** (CRF) representation, as reported in (Teh et al. 2006):

$$P(Z_{i,j} = z \mid \mathbf{Z}^{-i,j}, \beta_0) \propto \begin{cases} \alpha\beta_0^u, & \text{if } z = z_{new} \\ n_z + \alpha\beta_0^z, & \text{otherwise} \end{cases} \tag{4}$$

In this formula, n_z is the number of event mentions with event index z, z_{new} is a new event index not used already in $\mathbf{Z}^{-i,j}$, β_0^z are the global mixing proportions associated with the K events, and β_0^u is the weight for the unknown mixture component.

Next, to generate the mention head lemma (in this model, $\mathbf{X} = \langle \mathbf{HL} \rangle$), the event z is associated with a multinomial emission distribution over the HL feature values having the parameters $\phi = \langle \phi_Z^{hl} \rangle$. We assume that this emission distribution is drawn from a symmetric Dirichlet distribution with concentration λ_{HL}:

$$P(HL_{i,j} = hl \mid \mathbf{Z}, \mathbf{HL}^{-i,j}) \propto n_{hl,z} + \lambda_{HL} \tag{5}$$

where $HL_{i,j}$ is the head lemma of mention j from document i, and $n_{hl,z}$ is the number of times the feature value hl has been associated with the event index z in $(\mathbf{Z}, \mathbf{HL}^{-i,j})$.

5.2 The HDP$_{flat}$ Model

A model in which observable components are represented only by one feature type has the tendency to cluster these components based on their corresponding feature values. This model may produce good results for tasks such as topic discovery where the linguistic objects rely only on lexical information. Because event coreference involves clustering complex objects characterized by a large number of features, it is desirable to extend the HDP$_{1f}$ model with a generalized model where additional feature types can be easily incorporated. Moreover, this extension should allow multiple feature types to be added simultaneously.

To facilitate this extension, we assume that the feature variables are conditionally independent given $\mathbf{Z}$. This assumption considerably reduces the complexity of computing $P(\mathbf{Z} \mid \mathbf{X})$. For example, if we want to incorporate into the previous model the feature type associated with the semantic frame evoked by every event mention (i.e., *FR*), the formula becomes:

$$\begin{aligned} P(Z_{i,j} \mid \mathbf{HL}, \mathbf{FR}) &\propto P(Z_{i,j}) P(HL_{i,j}, FR_{i,j} \mid \mathbf{Z}) \\ &\propto P(Z_{i,j}) P(HL_{i,j} \mid \mathbf{Z}) P(FR_{i,j} \mid \mathbf{Z}) \end{aligned} \tag{6}$$

In this formula, we omit the conditioning components of $\mathbf{Z}$, **HL**, and **FR** for the sake of clarity. The graphical representation corresponding to this model is illustrated in Figure 3(b). In general, if $\mathbf{X}$ consists of L feature variables, the inference formula for the Gibbs sampler is defined as:

$$P(Z_{i,j} \mid \mathbf{X}) \propto P(Z_{i,j}) \prod_{FT \in \mathbf{X}} P(FT_{i,j} \mid \mathbf{Z}) \tag{7}$$

The graphical model for this general setting is depicted in Figure 3(c). Drawing an analogy, the graphical representation involving **Z** and feature variables resembles the graphical representation of a naive Bayes classifier.

5.3 The HDP$_{struct}$ Model

When dependencies between feature type variables exist (e.g., in our case, frame elements are dependent on the semantic frames that define them, and frames are dependent on the words that evoke them), various global distributions are involved for computing $P(\mathbf{Z}|\mathbf{X})$. For the model depicted in Figure 3(d), for instance, the posterior probability is given by:

$$P(Z_{i,j}|\mathbf{X}) \propto P(Z_{i,j})P(FR_{i,j}|HL_{i,j},\theta)\prod_{FT\in\mathbf{X}} P(FT_{i,j}|\mathbf{Z}) \quad (8)$$

In this model, $P(FR_{i,j}|HL_{i,j},\theta)$ is a global distribution parameterized by θ, and FT is a feature type variable from the set $\mathbf{X}=\langle\mathbf{HL},\mathbf{POS},\mathbf{FR}\rangle$. However, one limitation of this particular model is that it requires domain knowledge in order to establish the dependencies between the feature type variables.

For all the HDP extended models, we computed the prior and likelihood factors as described in the HDP$_{1f}$ model. In the inference mechanism, we assigned soft counts to those likelihood factors whose corresponding feature values cannot be extracted for a given event mention (e.g., unspecified predicate arguments). It is worth noting that there exist event mentions for which not all the features can be extracted. For instance, the feature types corresponding to the left and right lemmatized head words (denoted in Section 4 as LHE and RHE, respectively) are missing for the first and last event mentions in a document. Also, many semantic roles can be absent for an event mention in a given context.

6. Infinite Feature Models

One of the main limitations of the HDP extensions presented in the previous section is that these models have limited capabilities in representing the observable objects characterized by a large number of feature types. This is because, in order to sample the event indices into the set of indicator random variables **Z**, the HDP models need to store in memory large matrices that encode the significant statistics for the observable components associated with each cluster. More specifically, in order to compute the likelihood factors in Equation (5), for each feature type FT$_i$, $i = 1 \ldots L$, we assigned a counting matrix having the number of rows equal with the number of distinct feature values corresponding to FT$_i$ and $K+1$ columns, where K represents the number of inferred events. For instance, the counting matrix corresponding to the head lemma feature type (HL) stores the number of times each feature value of the HL feature type has been associated with each event index during the HDP generative process. The number $n_{hl,z}$ in Equation (5), for example, is stored in a cell of this matrix.

Just to have an idea of how much memory the HDP models require to infer the events from OntoNotes, we made the following calculation. In OntoNotes, we automatically identified a total number of 81,938 event mentions for which we extracted 454,170 distinct feature values. For all data sets, we considered L = 132 feature types, which means that, on average, each feature type is associated with approximately 3,440 feature

values. Because K is bounded by the total number of event mentions considered (i.e., the case when each event mention is associated with a different event), the maximum value that it can reach when inferring the event indices from OntoNotes is 81,938. If we consider that each cell from the counting matrices associated with each feature type is represented into the memory by one byte, the total space required to store only one such matrix is, on average, 81,938×3,440 bytes. By a simple computation, the total amount of memory to store all 132 matrices is $\sim$ 34.6 gigabytes (GB). Furthermore, by adding more data, the amount of memory needed by the HDP models increases considerably. For instance, if we consider all three data sets (with a total number of 148,402 event mentions and 832,611 distinct feature values), the memory space required increases to 115 GB. Because in our implementation we used the int type (4 bytes) to represent the counting matrices, the total amount of memory required by the HDP extensions to infer the event indices from OntoNotes and all three data sets when considering all 132 feature types is in fact $4 \times 34.6 = 138.4$ GB and $4 \times 115 = 460.3$ GB, respectively.

Due to this limitation, the HDP extensions will be able to run only using a restricted, manually selected set of feature types.[7] Therefore, the existence of a novel methodology that is able to consider a much smaller subset of representative feature values from the entire feature space is necessary. For this purpose, we devised two novel approaches that provide a more flexible representation of the data by modeling event mentions with an infinite number of features and by using a mechanism to automatically select a finite set of the most salient features for each mention in the inference process. The first approach uses the **Markov Indian buffet process** (mIBP) to represent each object as a sparse subset of a potentially unbounded set of latent features (Griffiths and Ghahramani 2006; Ghahramani, Griffiths, and Sollich 2007; Van Gael et al. 2008), and combines it with the HDP extension presented in the previous section. We call this hybrid the **mIBP–HDP model**. The second approach uses the **infinite factorial hidden Markov model** (iFHMM), which is an extension of mIBP, and combines it with the **infinite hidden Markov model** (iHMM) to form the **iFHMM–iHMM model**.

6.1 The mIBP–HDP Model

In this section, we describe a model that is able to represent event mentions characterized by an unbounded set of feature values into the HDP framework. Although the feature space describing event mentions is unbounded, this approach is able to model the uncertainty in the number of feature values M that will be used for clustering event mentions and, at the same time, is able to guarantee that this number is finite at any point in time during the generative process. First, we use mIBP to describe a mechanism for assigning to each event mention a sparse subset of feature values from the set of M observed feature values used in the clustering process. We will use the set of notations introduced in this description when presenting both mIBP–HDP and iFHMM–iHMM models. Then, we will show how this mechanism is coupled into the HDP framework.

6.1.1 The Markov Indian Buffet Process. The Markov Indian buffet process (Van Gael, Teh, and Ghahramani 2008) defines a distribution over an unbounded set of independent hidden Markov chains, where each chain is associated with a binary latent feature value

7 Because, in general, most of the counts corresponding to each feature value are assigned to a single cluster, a partial solution for this problem would be an efficient way of managing the sparsity in the counting matrices. However, the main issue of representing the entire set of features into the HDP models remains unaddressed.

that evolves over time according to Markov dynamics. Specifically, if we denote by M the total number of Markov chains associated with the latent feature values and by T the number of observations, mIBP defines a probability distribution over a binary matrix $\mathbf{F}$ with an unbounded number of rows M ($M \to \infty$) and T columns.

In our framework, we use mIBP to incrementally build the set of M observed feature values that will be used for clustering event mentions (denoted as $\{f^1, f^2, \ldots, f^M\}$), as well as to determine which of these feature values will be selected to explain each event mention. The sequence of observations is associated with the sequence of event mentions, $y_1, y_2, \ldots, y_T$, and each latent feature value in the mIBP framework is associated with one observed feature value from the unbounded set of features that characterize our event mentions. It is worth mentioning that, at any given time point during the mIBP generative process, from the unbounded set of observed features, we index only these M observed feature values that correspond to the set of hidden feature values.

The selection of the observed feature values which will represent each event mention in the clustering process is determined by the indicator random variables of the binary matrix $\mathbf{F}$. For instance, the selection of the observed feature value f^i for the event mention y_t is indicated by an assignment of the binary random variable F_t^i to 1 in the mIBP generative process. More specifically, the set of observed feature values that will represent the event mention y_t is indicated in the matrix by the column vector of binary random variables $\mathbf{F_t} = \langle F_t^1, F_t^2, \ldots, F_t^M \rangle$. Therefore, $\mathbf{F}$ decomposes the event mentions and represents them as feature value factors, which can then be associated with hidden variables in an iFHMM model as described in Van Gael, Teh, and Ghahramani (2008).

The transition probabilities of the binary Markov chain associated with a latent feature value, $\mathbf{F^m} = \langle F_1^m, F_2^m, \ldots, F_T^m \rangle$, are given by the following transition matrix:

$$\mathbf{W}^{(m)} = \begin{pmatrix} 1 - a_m & a_m \\ 1 - b_m & b_m \end{pmatrix} \tag{9}$$

where $\mathbf{W}_{ij}^{(m)} = P(F_{t+1}^m = j \mid F_t^m = i)$, the parameters $a_m \sim \mathsf{Beta}(\alpha'/M, 1)$ and $b_m \sim \mathsf{Beta}(\gamma', \delta')$, and the initial state $F_0^m = 0$. In the mIBP process, the hidden variable associated with an observed feature value f^m and an event mention y_t is generated from the following Bernoulli distribution:

$$F_t^m \sim \mathsf{Bernoulli}(a_m^{1-F_{t-1}^m} b_m^{F_{t-1}^m}) \tag{10}$$

Based on these definitions, we computed the probability of the feature matrix $\mathbf{F}$[8] (in which the parameters $\mathbf{a}$ and $\mathbf{b}$ are integrated out analytically) by recording the number of $0 \to 0$, $0 \to 1$, $1 \to 0$, and $1 \to 1$ transitions for each binary chain m into the counting variables c_m^{00}, c_m^{01}, c_m^{10}, and c_m^{11}, respectively. For example, the c_m^{11} associated with the feature value representing the VERB class (f^m = HWC:VERB) counts how many times this feature value was assigned to the event mention y_t when it was also assigned to the previous event mention y_{t-1} during the generative process.

The stochastic process that derives the probability distribution in terms of these variables is defined as follows. In the first step, the process assigns a value of 1 to a number of $\mathsf{Poisson}(\alpha')$ latent features for the first component. In our implementation, this statement is equivalent with the process of randomly selecting for the first event

8 Technical details for computing this probability are described in Van Gael, Teh, and Ghahramani (2008).

mention a number of $\mathsf{Poisson}(\alpha')$ observed feature values. In the general case, the sampling of the binary variable from the m^{th} Markov chain and associated with the t^{th} event mention depends on the value assigned to the hidden variable in the previous $t-1$ step:

$$\begin{aligned} P(F_t^m = 1 \,|\, F_{t-1}^m = 1) &= \frac{c_m^{11} + \delta'}{\gamma' + \delta' + c_m^{10} + c_m^{11}} \\ P(F_t^m = 1 \,|\, F_{t-1}^m = 0) &= \frac{c_m^{01}}{c_m^{00} + c_m^{01}} \end{aligned} \tag{11}$$

As a result, in our implementation, the observed feature value f^m is selected for the t^{th} event mention according to the probabilities presented in Equation (11). For example, in order to select the feature value which indicates that the t^{th} event mention has the OCCURRENCE event class (i.e., f^m = EC:OCCURRENCE), we need to determine whether or not the event mention $t-1$ from the document collection selected this feature value. In the cases when EC:OCCURRENCE was previously selected for the event mention $t-1$ ($F_{t-1}^m = 1$), we select this feature value according to $P(F_t^m = 1 \,|\, F_{t-1}^m = 1)$. Otherwise, the selection is determined according to $P(F_t^m = 1 \,|\, F_{t-1}^m = 0)$. Furthermore, in the t^{th} step of the generative process, the same sampling mechanism is repeated until all M latent feature values are generated. After sampling all these feature values for the t^{th} event mention, an additional number of $\mathsf{Poisson}(\alpha'/t)$ new feature values are assigned to this mention, and M gets incremented accordingly.

As an observation regarding the mIBP generative process, it has been shown that M grows logarithmically with the number of observed components (in our case, event mentions) (Ghahramani, Griffiths, and Sollich 2007; Doshi-Velez 2009). This type of growth is desirable because it provides a scalable solution for our models to work in an efficient way on fairly large data sets.

6.1.2 Integration of mIBP into HDP. One direct application of the mIBP model is to integrate it into the framework of the HDP extension model described in the previous section. In this way, the new nonparametric extension will have the benefits of capturing the uncertainty regarding the number of mixture components that are characterized by a potentially infinite number of feature values. However, to make this hybrid work, we have to devise a mechanism in which only a finite set of relevant feature values will be selected to explain each observation (i.e., event mention) in the HDP inference process.

Our idea of selecting a finite set of representative feature values for each event mention is based on a heuristic approach that is used after running the mIBP generative process. Specifically, by considering the event mention y_t, f^m one of the feature values that characterizes y_t, q_m the number of times f^m was selected for all mentions during mIBP, and v_t a threshold variable for y_t such that $v_t \sim \mathsf{Uniform}(1, \max\{q_m \mid F_t^m = 1\})$, we define the finite set of feature values B_t corresponding to the observation y_t as:

$$B_t = \{f^m \mid F_t^m = 1 \wedge q_m \geq v_t\} \tag{12}$$

A pictorial representation of this idea is illustrated in Figure 4, where only the feature values f^m with the corresponding counts q_m above the threshold indicated by v_t are selected in B_t. The finiteness of this feature set is based on the observation that, at any time point during the generative process of the mIBP model, only a finite set of latent features have assigned a value of 1 for an event mention. Furthermore, based on

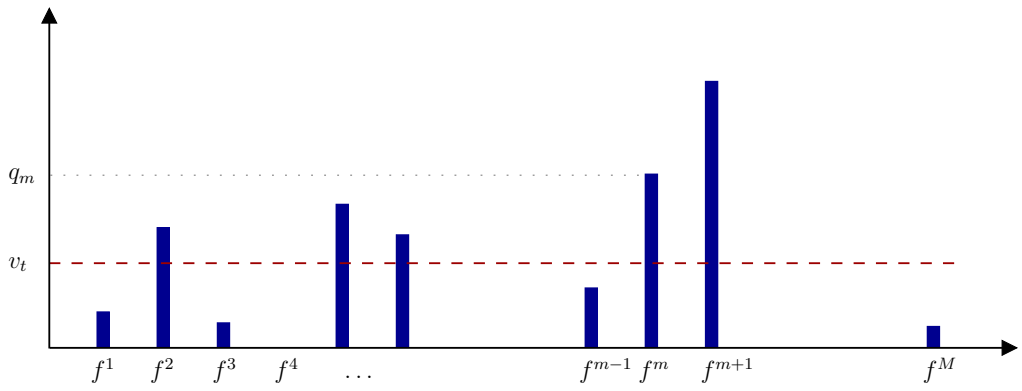

Figure 4
Graphical representation of the mechanism for filtering the feature values associated with the event mention y_t. After this mechanism is applied, y_t will be represented only by the feature values f^m for which their corresponding counts q_m are above the threshold variable v_t.

the assumption that the more a feature value is selected during the mIBP generative process the more relevant it is for the event coreference task, each set B_t contains the most informative feature values that are able to explain its corresponding event mention y_t. This last property is ensured by the second constraint imposed when building each set B_t (i.e., $q_m \geq v_t$). Due to the fact that the threshold variables are sampled using a uniform distribution, we denote this model as mIBP–HDP$_{uniform}$.

The feature values selected by this mechanism are used to represent the event mentions in the clustering process of the HDP. The main difference from the original implementation of the HDP extensions is that, in this new model, instead of representing the event mentions by the entire set of feature values from the initial feature space (which can be as large as possible), only a restricted subset of these feature values is considered. Furthermore, due to the random process of selecting the feature values, the number of feature values associated with each event mention can vary significantly. We adapted the implementation of the HDP framework to this modification by truncating all counting matrices such that they will represent only the feature values selected in mIBP. More specifically, we removed from each counting matrix the rows corresponding to all the feature values that were not selected during the mIBP generative process. Because M grows as $O(\log T)$, it now becomes feasible for the HDP extension models to represent event mentions using the entire set of feature types. It is important to mention that this modification does not affect the implementation of the Gibbs sampler in the HDP framework because we always normalize the probabilities corresponding to the likelihood factors in Equation (5) when computing the posterior distribution over event indices.

Moreover, using the assumption that the relevance of a feature value is proportional with the number of times it was selected during the mIBP generative process, we explored additional heuristics for building the sets of feature values B_t for each event mention. In general, we chose these new heuristics to be biased towards selecting more relevant feature values f^m for each event mention y_t (i.e., their counts q_m to be closer to $\max\{q_m \mid F_t^m = 1\}$). One such heuristic is based on the method that considers for each event mention y_t all feature values f^m with the counts $q_m \geq 1$ (i.e., $v_t = 1$). In this case, each set B_t contains all the observed feature values selected for each event mention y_t during the mIBP process, and therefore it represents a subset of the set of observed feature values $\{f^1, f^2, \ldots, f^M\}$. It is worth mentioning that all the subsets of $\{f^1, f^2, \ldots, f^M\}$

are finite due to the fact that M is finite at any given point in time during mIBP. In consequence, all the B_t sets derived using this heuristic are finite. Because no feature value is filtered out after it was assigned to an event mention during mIBP, we denote the model implementing this heuristic as mIBP–HDP$_{unfiltered}$. Starting from the distribution of the counting variables q_m corresponding to those feature values f^m selected during the mIBP generative process for an event mention y_t, another heuristic considers for building each set B_t only the feature values with the counts above the median of this distribution (mIBP–HDP$_{median}$). Finally, the last heuristic we experimented with is based on the idea of sampling the threshold variables v_t directly from the distribution of the counting variables associated with each event mention y_t (mIBP–HDP$_{discrete}$). The implementation of these three heuristics is possible due to the observation that in the mIBP–HDP framework the size of each set B_t is not required to be known in advance.

6.2 The iFHMM–iHMM Model

Over the years, the **hidden Markov model** (HMM) (Rabiner 1989) has proven to be one of the most commonly used statistical tools for modeling time series data. Due to the efficiency in estimating its parameters, various HMM generalizations were proposed for a better representation of the latent structure encoded in this type of data. Figure 5 illustrates a hierarchy of HMM extensions whose main criteria of expansion is based on relaxing the constraints on the parameters M (the number of state chains) and K (the number of clustering components). In the **factorial hidden Markov model** (FHMM), Ghahramani and Jordan (1997) introduced the idea of factoring the hidden state space into a finite number of state variables, in which each of these variables has its own Markovian dynamics. Later on, Van Gael, Teh, and Ghahramani (2008) introduced the

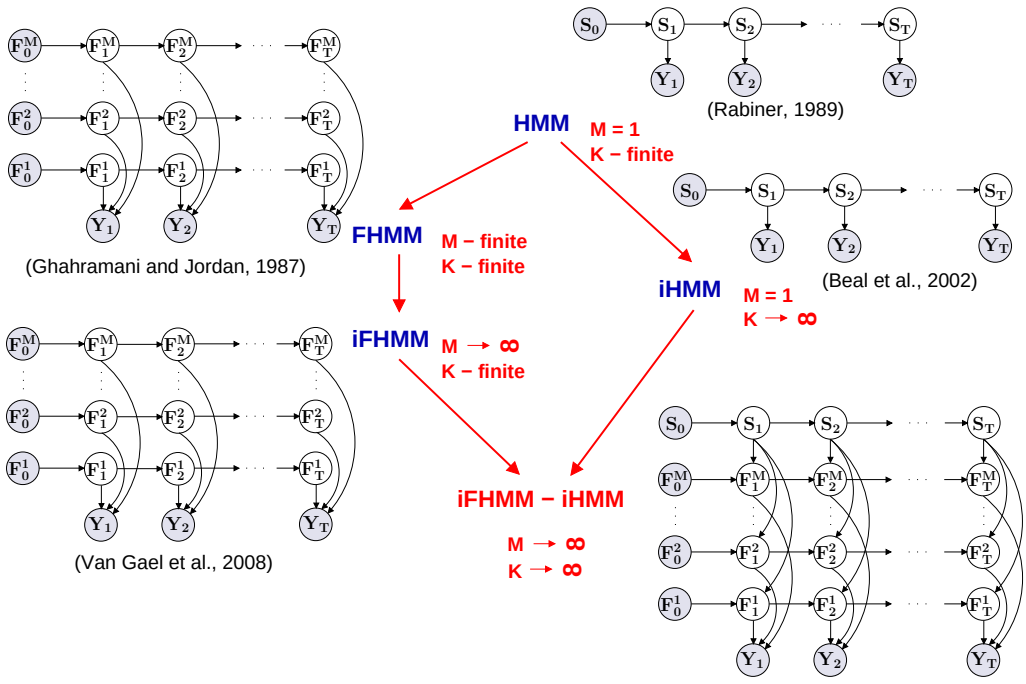

Figure 5
Extensions of the hidden Markov model.

infinite factorial hidden Markov model (iFHMM) with the purpose of allowing the number of parallel Markov chains M to be learned from data. Although the iFHMM provides a more flexible representation of the latent structure, it cannot be used as a framework where the number of clustering components K is infinite. In this direction, Beal, Ghahramani, and Rasmussen (2002) proposed the **infinite hidden Markov model** (iHMM) in order to perform inferences with an infinite number of states K. To further increase the representational power for modeling discrete time series data, we introduce a novel nonparametric extension that combines the best of the iFHMM and iHMM models (denoted as iFHMM–iHMM) and lets both parameters M and K to be learned from data.

As shown in Figure 5, the graphical representation of this new model consists of a sequence of hidden state variables, $(s_1, \ldots, s_T)$, that corresponds to the sequence of event mentions $(y_1, \ldots, y_T)$. Each hidden state s_t can be assigned to one of the K latent events, $s_t \in \{1, \ldots, K\}$, and each mention y_t is represented by a column vector of binary random variables $\langle F_t^1, F_t^2, \ldots, F_t^M \rangle$. One element of the transition probability π is defined as $\pi_{ij} = P(s_t = j \,|\, s_{t-1} = i)$, and a mention y_t is generated according to a likelihood model $\mathcal{F}$ that is parameterized by a state-dependent parameter ϕ_{s_t} $(y_t \,|\, s_t \sim \mathcal{F}(\phi_{s_t}))$. The observation parameters ϕ are independent and identically distributed drawn from a prior base distribution H.

6.2.1 Inference. The main idea of the inference mechanism corresponding to this new model is illustrated in Figure 6. As depicted in this figure, each step in the generative process of the new hybrid model is performed in two consecutive phases. In the first phase, the binary random variables associated with each feature value from the iFHMM framework are sampled using the mIBP mechanism, and consequently, the most salient feature values are selected for each event mention (Figure 6: Phase I). Of note, the B_t sets of feature values associated with each event mention y_t are determined using the same set of heuristics as described in Section 6.1. In the second phase, the feature values sampled so far, which become observable during this phase, are used in an adapted **beam sampling algorithm** (Van Gael et al. 2008) to infer the clustering components or latent events (Figure 6: Phase II).

Because we utilized the same mechanism for determining the sets of relevant feature values for each event mention (as described in Section 6.1), in this section we focus on describing our implementation of the beam sampling algorithm. The beam sampling algorithm (Van Gael et al. 2008) combines the ideas of slice sampling (Neal 2003) and dynamic programming for an efficient sampling of state trajectories. Because in time series models the transition probabilities have independent priors (Beal, Ghahramani, and Rasmussen 2002), Van Gael et al. (2008) also used the HDP mechanism to allow couplings across transitions. For sampling the whole hidden state trajectory $\mathbf{s}$, this algorithm uses a **forward filtering-backward sampling technique**.

As described in Van Gael et al. (2008), in the forward step, an auxiliary variable u_t is sampled for each mention y_t, $u_t \sim \mathsf{Uniform}(0, \pi_{s_{t-1}s_t})$. The auxiliary variables $\mathbf{u}$ are used to filter only those trajectories $\mathbf{s}$ for which $\pi_{s_{t-1}s_t} \geq u_t$, for all t. Also, in this step, for all t, the probabilities $P(s_t \,|\, y_{1:t}, u_{1:t})$ are computed as follows:

$$P(s_t \mid y_{1:t}, u_{1:t}) \propto P(y_t \mid s_t) \sum_{s_{t-1}: u_t < \pi_{s_{t-1}s_t}} P(s_{t-1} \mid y_{1:t-1}, u_{1:t-1}) \tag{13}$$

In this formula, the dependencies involving parameters π and ϕ are omitted for clarity.

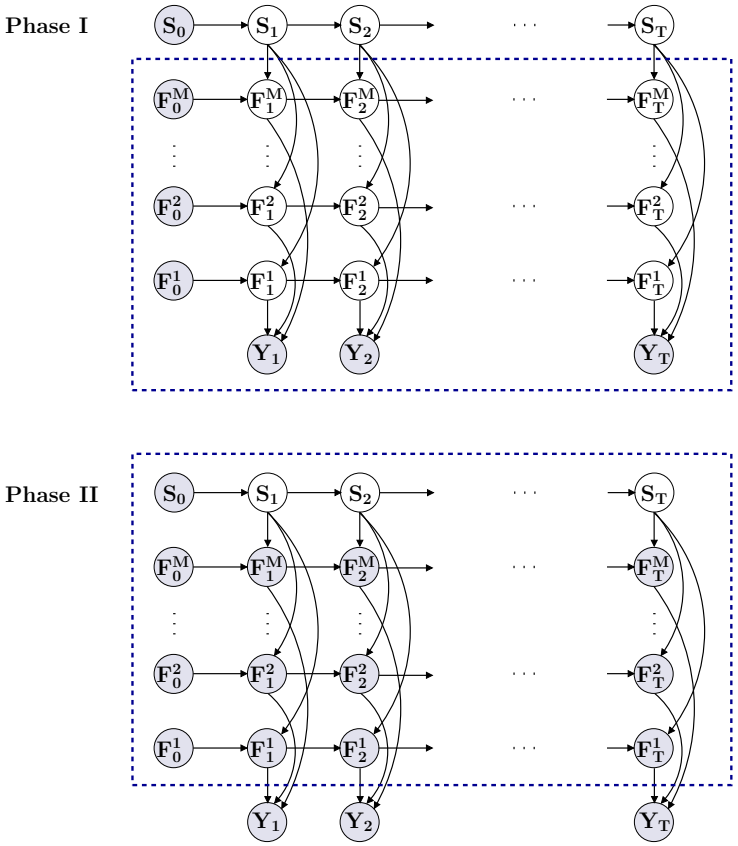

Figure 6
A step in the generative process of the iFHMM–iHMM model is performed in two phases: (Phase I) sample the feature values for each event mention, and (Phase II) sample the latent events.

In the backward step, first, the event for the last state s_T is directly sampled from $P(s_T | y_{1:T}, u_{1:T})$ and then, for all $t : T-1, 1$, each state s_t given s_{t+1} is sampled using the following formula:

$$P(s_t \mid s_{t+1}, y_{1:T}, u_{1:T}) \propto P(s_t \mid y_{1:t}, u_{1:t}) P(s_{t+1} \mid s_t, u_{t+1}) \tag{14}$$

To sample the emission distribution ϕ efficiently and to ensure that each mention is characterized by a finite set of representative features, in our implementation of the beam sampling algorithm, we set the base distribution H to be conjugate with the data distribution $\mathcal{F}$ in a Dirichlet-multinomial model with the multinomial parameters $(o_1, \ldots, o_K)$ defined as:

$$o_k = \sum_{t=1}^{T} \sum_{f^m \in B_t} n_{mk} \tag{15}$$

where n_{mk} counts how many times the feature value f^m was assigned in the generative process to event k, and B_t stores a finite set of feature values for y_t as defined in

Section 6.1. As can be noticed, the multinomial parameters defined here are finite due to the fact that each set of feature values B_t is finite and the number of event mentions T is fixed. This allows us to define a proper emission distribution for the new hybrid model. In a similar manner to the notations of the mIBP–HDP model, we make notations of the iFHMM–iHMM model according to the heuristic used for selecting the feature values.

7. Evaluation

In this section, we present the evaluation framework of the Bayesian models for both within-document (WD) and cross-document (CD) coreference resolution. We start by briefly describing the experimental set-up and coreference evaluation measures, and then continue by showing the experimental results on the ACE, OntoNotes, and EventCorefBank data sets. Finally, we conclude with an analysis of the most common errors made by the Bayesian models.

7.1 The Experimental Set-up

In the data processing phase, we extracted the linguistic features described in Section 4 for each event mention annotated in the three data sets. As a result of this phase, in the ACE corpus, we identified 6,553 event mentions grouped into 4,946 events, and in the OntoNotes corpus, we identified 11,433 event mentions grouped into 3,393 events. Likewise, in the new ECB corpus, we distinguished 1,744 event mentions, 1,302 within-document events, 339 cross-document events, and 43 seminal events (or topics). Table 1 lists additional statistics extracted from these three data sets after performing this phase.

It is also worth mentioning that for processing OntoNotes we devoted additional efforts. This is because, in spite of the fact that OntoNotes provides coreference annotations for both entity and event mentions, the annotations from this data set do not specify which of the mentions refer to entities and which of them refer to events. Therefore, in order to identify only the event mentions from OntoNotes, we first ran our event identifier (Bejan 2007) and then marked as event mentions only those mentions annotated in this data set that overlap with the mentions extracted by the event identifier.

Table 1
Statistics of the ACE, OntoNotes, and ECB corpora.

	ACE	OntoNotes	ECB
Number of true mentions	6,553	11,433	1,744
Number of system mentions	45,289	81,938	21,175
Number of within-document events	4,946	3,393	1,302
Number of cross-document events	–	–	339
Number of documents	745	1,540	482
Number of seminal events	–	–	43
Average number of true mentions/within-document event	1.32	3.37	1.34
Average number of true mentions/document	8.79	7.42	3.62
Average number of true mentions/seminal event	–	–	40.55
Average number of system mentions/document	60.79	53.2	43.93
Average number of within-document events/document	6.63	2.20	2.70
Average number of within-document events/seminal event	–	–	30.27
Average number of cross-document events/seminal event	–	–	7.88
Average number of documents/seminal event	–	–	11.20
Number of distinct feature values for system mentions	391,798	454,170	237,197

Using this procedure, we marked a number of 4,940 mentions as event mentions from the total number of 67,500 mentions annotated in OntoNotes. In a second step of processing OntoNotes, we extended the number of event mentions to 11,433 by marking all the mentions that share the same cluster with at least one event mention from the set of 4,940 previously identified event mentions. From the 6,493 event mentions marked in this step, the majority of them correspond to nouns (4,707) and to the *it* pronoun (767).

Although only a small subset of event mentions was manually annotated with event coreference information in the three data sets (also called the set of **true** or **gold event mentions**), during the generative process, we considered all possible event mentions that are expressed in the data sets for every specific event. We believe this is a more realistic approach, in spite of the fact that we evaluated only the manually annotated events. For this purpose, we ran the event identifier described in Bejan (2007) on the ACE, OntoNotes, and ECB corpora, and extracted 45,289, 81,938, and 21,175 event mentions, respectively. It is also worth mentioning that the set of event mentions obtained from running the event identifier (also called the set of **system event mentions**) on ACE and ECB includes more than 98% from the set of true event mentions. In terms of feature space dimensionality over the two data sets, we performed experiments with a set of 132 feature types, where each feature type consists, on average, of 6,300 distinct feature values.

In the evaluation phase, we considered only the true mentions from the ACE test data set and from the test sets of a five-fold cross validation scheme on the OntoNotes and ECB data sets. For evaluating the cross-document coreference annotations from EventCorefBank, we adopted the same approach as described in Bagga and Baldwin (1999) by merging all the documents from the same topic into a meta-document and then scoring this document as performed for within-document evaluation. To compute the final results of our experiments, we averaged the results over five runs of the generative models.

7.2 Coreference Resolution Metrics

Because there is no agreement on the best coreference resolution metric, we used four metrics for our evaluation: the **link**-based MUC metric (Vilain et al. 1995), the **mention**-based B^3 metric (Bagga and Baldwin 1998), the **entity**-based CEAF metric (Luo 2005), and the pairwise (PW) metric. These metrics report results in terms of recall (R), precision (P), and F-score (F) by comparing the true set of coreference chains $\mathcal{T}$ (i.e., the manually annotated coreference chains) against the set of chains predicted by a coreference resolution system $\mathcal{S}$. Here, a **coreference link** represents a pair of coreferential mentions whereas a **coreference chain** represents all the event mentions from the same cluster with coreference links between consecutive mentions.

The MUC recall computes the number of common coreference links in $\mathcal{T}$ and $\mathcal{S}$ divided by the number of links in $\mathcal{T}$, and the MUC precision computes the number of common links in $\mathcal{T}$ and $\mathcal{S}$ divided by the number of links in $\mathcal{S}$. As was previously noted (Luo et al. 2004; Denis and Baldridge 2008; Finkel and Manning 2008), this metric favors the systems that group mentions into smaller number of clusters (or, in other words, systems that predict large coreference chains) and does not take into account single mention clusters. For instance, a system that groups all entity mentions into the same cluster achieves a MUC score that surpasses any published results of known systems developed for the task of entity coreference resolution.

The B^3 metric was designed to overcome some of the MUC metric's shortcomings. This metric computes the recall and precision for each mention and then estimates the

overall score by averaging over all mention scores. For a given mention m, the scorer compares the true coreference chain that contains the mention m (T_m) against the system chain that contains the same mention m (S_m). Thus, the recall for m is the ratio of the number of common elements in S_m and T_m over the number of elements in T_m. Similarly, the precision corresponding to the mention m is the ratio of the number of common elements in S_m and T_m over the number of elements in S_m. Because this metric computes the precision and recall for each mention, it will penalize in precision the systems that predict a small number of clusters. Because of the same reason, this metric includes single mention clusters in the evaluation.

The Constrained Entity-Alignment F-Measure (CEAF) scorer finds the best alignment between the set of true coreference chains $\mathcal{T}$ and the set of predicted coreference chains $\mathcal{S}$. This is equivalent to finding the best mapping in a weighted bipartite graph. We computed the weight of a pair of coreference chains (T_i, S_j), with $T_i \in \mathcal{T}$ and $S_j \in \mathcal{S}$, by using the ϕ_4 similarity measure described in Luo et al. (2004). Therefore, the CEAF recall and precision measures are computed as the overall similarity score of the best alignment divided by the self-similarity score of the coreference links in $\mathcal{T}$ and $\mathcal{S}$, respectively.

The last coreference metric that we considered, the PW metric, finds correspondences between all mentions pairs (m_i, m_j) from the true and system chains with the coreference chains linking the mentions m_i and m_j in the system and true chains, respectively. As can be noticed, this metric overpenalizes those systems that predict too many or too few clusters when compared with the number of true clusters.

7.3 Experimental Results

Tables 2, 3, 4, and 5 list the results performed by our proposed baselines (rows 1–2), by the HDP models (rows 3–8), by the mIBP–HDP model (row 9), and by the iFHMM–iHMM model (rows 10–13). We discuss the performance achieved by these models in the remaining part of this section.

Table 2
Results for WD coreference resolution on the ACE data set.

	Model configuration	MUC			B^3			CEAF			PW		
		R	P	F	R	P	F	R	P	F	R	P	F
1	BL_{eclass}	94.3	33.1	49.0	97.9	25.0	39.9	14.7	64.4	24.0	93.5	8.2	15.2
2	BL_{syn}	71.9	29.6	41.5	89.3	36.7	52.1	25.1	64.8	36.2	63.8	10.5	18.1
3	HDP_{1f} (HL)	62.2	43.1	50.9	86.0	70.6	77.5	62.3	76.4	68.6	50.5	27.7	35.8
4	HDP_{flat} (LF)	52.5	51.1	51.8	82.9	82.6	82.7	74.9	75.8	75.3	42.4	41.9	42.1
5	(LF+CF)	48.9	53.1	51.0	82.0	84.9	83.4	77.8	75.3	76.6	37.9	45.1	41.2
6	(LF+CF+WF)	53.8	53.9	53.9	83.3	83.6	83.4	76.3	76.2	76.3	42.2	43.9	43.0
7	(LF+CF+WF+SF)	53.5	54.2	53.9	83.4	84.2	**83.8**	76.9	76.5	**76.7**	43.3	47.1	**45.1**
8	HDP_{struct} (HL→FR→FEA)	61.9	49.0	**54.7**	86.2	76.9	81.3	69.0	77.5	73.0	53.2	38.1	44.4
9	mIBP-$HDP_{unfiltered}$	48.7	41.9	45.1	81.7	76.4	79.0	68.8	73.8	71.2	37.4	28.9	32.6
10	iFHMM-$iHMM_{unfiltered}$	50.7	52.0	51.4	82.8	83.6	83.2	75.8	75.0	75.4	41.4	42.6	42.0
11	iFHMM-$iHMM_{discrete}$	52.5	50.2	51.3	83.1	81.5	82.3	73.7	75.1	74.4	41.9	40.1	41.0
12	iFHMM-$iHMM_{median}$	52.8	49.6	51.1	83.0	81.3	82.1	73.2	75.2	74.2	40.7	39.0	39.8
13	iFHMM-$iHMM_{uniform}$	48.7	48.8	48.7	81.9	82.2	82.1	74.6	74.5	74.5	37.2	39.0	38.1

Table 3
Results for WD coreference resolution on the OntoNotes data set.

	Model configuration	MUC			B^3			CEAF			PW		
		R	P	F	R	P	F	R	P	F	R	P	F
1	BL_{eclass}	77.6	68.3	72.7	71.2	50.3	58.9	38.1	56.1	45.4	57.1	42.1	47.9
2	BL_{syn}	54.3	61.8	57.8	52.9	63.2	57.6	50.7	39.5	44.4	30.6	42.5	34.7
3	HDP_{1f} (HL)	78.3	72.2	75.1	77.1	54.1	63.6	40.4	50.4	44.9	67.4	44.0	52.9
4	HDP_{flat} (LF)	70.3	77.6	73.8	81.6	58.3	67.9	40.0	42.6	41.2	81.2	41.5	54.5
5	(LF+CF)	72.1	76.4	74.2	74.8	62.7	68.2	43.4	38.3	40.7	72.8	43.7	54.5
6	(LF+CF+WF)	79.6	77.1	78.2	81.7	57.3	67.1	39.6	43.1	41.1	81.0	41.9	54.6
7	(LF+CF+WF+SF)	72.1	77.6	74.7	74.9	64.0	**68.8**	39.4	50.7	44.3	74.5	44.1	**55.0**
8	HDP_{struct} (HL→FR→FEA)	84.7	78.1	**81.2**	85.6	55.2	67.1	67.4	37.3	**48.0**	79.1	40.1	51.4

7.3.1 Baseline Results. A simple baseline for event coreference, which was proposed by Ahn (2006), consists of grouping event mentions by their event classes (BL_{eclass}). To compute this baseline, we grouped mentions into clusters according to their corresponding EC feature value. In consequence, this baseline categorizes events into a small number of clusters, since the event identifier for extracting the EC features is trained to predict the seven event classes annotated in TimeBank. A second baseline that we implemented groups two event mentions if there is a (transitive) SYNONYMOUS relation between their corresponding head lemmas (BL_{syn}). To implement this baseline, we used the clusters built over the WordNet SYNONYMOUS relations as described in Section 4. Similarly to the MUC results reported for entity coreference resolution, the baselines that group event mentions into very few clusters are overestimated by the MUC metric (e.g., the MUC F-scores of BL_{eclass} in Table 5).

7.3.2 HDP Results. Due to memory limitations, we evaluated the HDP models on a restricted set of manually selected feature types. For the HDP_{1f} model, which plays

Table 4
Results for WD coreference resolution on the ECB data set.

	Model configuration	MUC			B^3			CEAF			PW		
		R	P	F	R	P	F	R	P	F	R	P	F
1	BL_{eclass}	92.2	39.8	55.6	97.7	55.8	71.0	44.5	80.1	57.2	93.7	25.4	39.8
2	BL_{syn}	75.0	34.3	47.0	91.5	57.4	70.5	45.7	75.9	57.0	65.3	21.9	32.6
3	HDP_{1f} (HL)	46.9	54.8	50.4	84.3	89.0	86.5	83.4	79.6	81.4	36.6	53.4	42.6
4	HDP_{flat} (LF)	34.7	85.7	49.4	81.4	98.2	89.0	92.7	77.2	84.2	24.7	82.8	37.7
5	(LF+CF)	36.1	83.4	50.1	81.5	98.0	89.0	92.8	77.9	84.7	24.6	80.7	37.4
6	(LF+CF+WF)	38.0	90.2	53.2	82.0	98.9	89.6	93.7	78.4	85.3	26.8	89.9	41.0
7	(LF+CF+WF+SF)	37.8	92.9	53.4	82.1	99.2	89.8	93.9	78.2	85.3	27.0	92.4	41.3
8	HDP_{struct} (HL→FR→FEA)	47.4	82.7	**60.1**	84.3	97.1	**90.2**	92.7	81.1	**86.5**	34.4	83.0	**48.6**
9	mIBP-$HDP_{unfiltered}$	38.2	68.8	48.9	82.1	95.3	88.2	90.3	78.5	84.0	26.5	67.9	37.7
10	iFHMM-$iHMM_{unfiltered}$	38.9	84.4	52.9	82.6	97.7	89.5	92.7	78.5	85.0	28.5	82.4	41.8
11	iFHMM-$iHMM_{discrete}$	40.2	85.2	54.6	82.6	98.1	89.7	93.2	79.0	85.5	29.7	85.4	44.0
12	iFHMM-$iHMM_{median}$	39.5	84.3	53.6	82.6	97.8	89.5	92.9	78.8	85.3	29.3	83.7	43.0
13	iFHMM-$iHMM_{uniform}$	39.5	85.2	53.9	82.5	98.1	89.6	93.1	78.8	85.3	29.4	86.6	43.7

Table 5
Results for CD coreference resolution on the ECB data set.

	Model configuration	MUC			B^3			CEAF			PW		
		R	P	F	R	P	F	R	P	F	R	P	F
1	BL_{eclass}	90.5	61.1	**72.9**	93.8	49.6	64.9	36.6	72.7	48.7	90.7	28.6	43.3
2	BL_{syn}	80.9	55.1	65.5	84.6	48.1	61.3	32.8	63.6	43.3	66.2	26.0	37.3
3	HDP_{1f} (HL)	47.7	70.5	56.8	67.0	86.2	75.3	76.2	57.1	65.2	34.9	58.9	43.5
4	HDP_{flat} (LF)	41.1	90.5	56.5	63.8	97.3	77.0	84.9	54.3	66.1	27.2	88.5	41.5
5	(LF+CF)	43.8	90.7	59.0	64.6	97.3	77.6	85.3	55.6	67.2	27.6	88.7	42.0
6	(LF+CF+WF)	46.2	93.0	61.6	65.8	98.0	78.7	86.7	57.1	68.8	29.6	93.0	44.8
7	(LF+CF+WF+SF)	44.4	95.3	60.5	65.0	98.7	78.3	86.9	56.0	68.0	29.2	95.1	44.4
8	HDP_{struct} (HL→FR→FEA)	51.9	89.5	65.7	69.3	95.8	**80.4**	86.2	60.1	**70.8**	37.5	85.6	**52.1**
9	mIBP-$HDP_{unfiltered}$	40.0	79.8	53.2	63.1	94.1	75.5	82.7	54.6	65.7	26.1	77.0	38.9
10	iFHMM-$iHMM_{unfiltered}$	48.2	89.8	62.7	67.2	96.4	79.1	85.6	58.0	69.1	32.5	87.7	47.2
11	iFHMM-$iHMM_{discrete}$	47.0	88.4	61.3	66.2	96.2	78.4	84.8	57.2	68.3	32.2	88.1	47.1
12	iFHMM-$iHMM_{median}$	48.3	89.6	62.7	67.0	96.5	79.0	86.1	58.3	69.5	33.1	88.1	47.9
13	iFHMM-$iHMM_{uniform}$	48.4	89.0	62.7	67.0	96.4	79.0	85.5	58.0	69.1	33.3	88.3	48.2

the role of baseline for the HDP_{flat} and HDP_{struct} models, we considered HL as the most representative feature type for performing the clustering of event mentions. In this configuration, the HDP_{1f} model outperforms the BL_{eclass} and BL_{syn} baselines. For the HDP_{flat} models (rows 4–7 in Tables 2–5), we classified the experiments according to the set of manually selected feature types. We found that the best configuration of features for this model consists of a combination of feature types from all the categories of features described in Section 4 (row 7 in Tables 2–5). For the experiments of the HDP_{struct} model, we considered the set of features of the best HDP_{flat} experiment as well as the conditional dependencies between the HL, FR, and FEA feature types.

In general, the HDP_{flat} model achieved the best performance results on the ACE test data set (the results in Table 2), whereas the HDP_{struct} model, which also encounters dependencies between feature types, proved to be more effective on the ECB data set for both within- and cross-document event coreference evaluation (as shown in Tables 4 and 5). On the OntoNotes data set, as listed in Table 3, HDP_{flat} shows better results than HDP_{struct} when considering the B^3 and PW metrics, whereas HDP_{struct} outperforms HDP_{flat} when considering the MUC and CEAF metrics. Moreover, the results of the HDP_{flat} and HDP_{struct} models show an F-score increase by 4–10 percentage points over the HDP_{1f} model, and therefore prove that the HDP extensions provide a more flexible representation for clustering objects characterized by rich properties than the original HDP model.

We also plot the evolution of the generative process associated with an HDP model. For instance, Figure 7 shows that the HDP_{flat} model corresponding to the experiment from row 7 in Table 2 converges in 350 iteration steps to a posterior distribution over event mentions from the ACE corpus with around 2,000 latent events.

7.3.3 mIBP–HDP Results. In spite of its advantage of working with a potentially infinite number of features in an HDP framework, the mIBP–HDP model (row 9 in Tables 2, 4, and 5) did not achieve a satisfactory performance in comparison with the other proposed models. However, the results were obtained by automatically selecting only 2% of

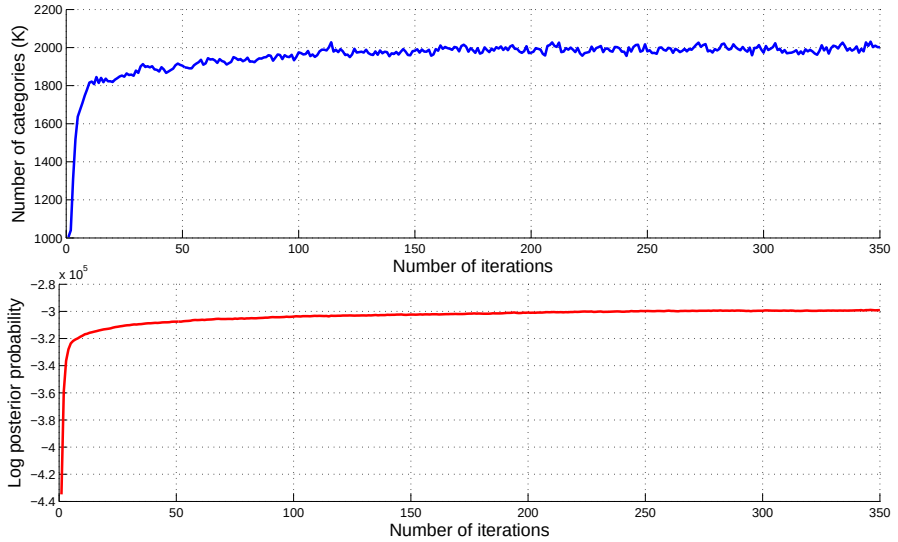

Figure 7
Evolution of the number of categories and the log of the posterior probability for the HDP_{flat} model.

distinct feature values from the entire set of values extracted from both corpora. When compared with the restricted set of features considered by the HDP_{flat} and HDP_{struct} models, the percentage of values selected by the mIBP–HDP model is only 6%.

7.3.4 iFHMM–iHMM Results. The results achieved by the iFHMM–iHMM model using automatic selection of feature values remain competitive against the results of the HDP models, where the feature types were manually tuned. When comparing the strategies for filtering feature values in the iFHMM–iHMM framework, we could not find a distinct separation between the results obtained by the $\text{iFHMM–iHMM}_{unfiltered}$, $\text{iFHMM–iHMM}_{discrete}$, $\text{iFHMM–iHMM}_{median}$, and $\text{iFHMM–iHMM}_{uniform}$ models. As observed from Tables 2, 4, and 5, most of the iFHMM–iHMM results fall in between the HDP_{flat} and HDP_{struct} results. Moreover, the results listed in these tables indicate that the iFHMM–iHMM model is a better framework than the HDP framework for capturing the event mention dependencies simulated by the mIBP feature sampling scheme.

A study of the impact on the performance results of the parameter α' that controls the number of feature values selected in the iFHMM–iHMM framework is presented in Figure 8. The results plotted in this figure show a small variation in performance for different values of α' indicating that the iFHMM–iHMM model is able to successfully handle the feature values that introduce additional noise in the data. Figure 8 also shows that the iFHMM–iHMM model achieves the best results on the ACE data set for a relative small value of α' ($\alpha' = 10$), which corresponds to 0.05% feature values sampled from the total number of feature values considered. However, because the number of event mentions in the ECB corpus is smaller than the number of mentions in the ACE corpus, the iFHMM–iHMM model utilizes a larger number of features values (0.91% of feature values selected for $\alpha' = 150$) extracted from the new corpus in order to obtain most of its best results.

The experiments depicted in Figure 8 were performed by using the *unfiltered* heuristic for selecting feature values in the iFHMM–iHMM model. Similar results were

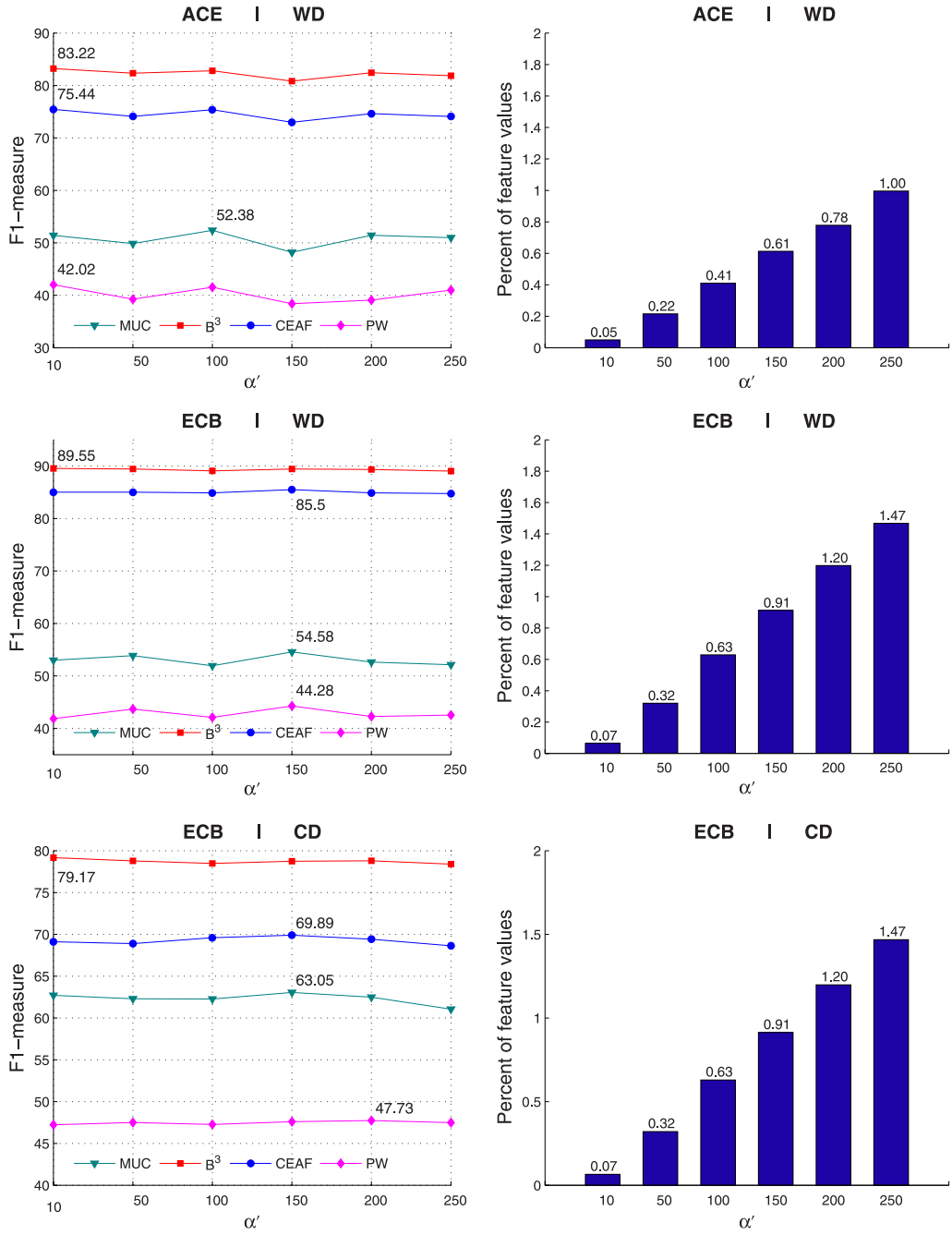

Figure 8
Performance results and feature reduction bars of the iFHMM–iHMM models for various α'.

obtained when considering the rest of the heuristics integrated in the iFHMM–iHMM framework. Also, the iFHMM–iHMM experiments using the *unfiltered*, *discrete*, *median*, and *uniform* heuristics from Tables 2–5 were performed by setting α' to 10, 100, 100, and 50, respectively. For the parameters γ' and δ', we considered a default value of 0.5.

To gain a deeper insight into the behavior of the iFHMM–iHMM model, we show in Figure 9 the performance results obtained by this model for different sets of feature types. For this purpose, we ran the iFHMM–iHMM$_{uniform}$ model with a fixed value of the α' parameter ($\alpha' = 50$) on increasing fractions of feature types.[9] The results confirm the fact that the sampling scheme of the feature values used in the iFHMM–iHMM framework does not guarantee the selection of the most salient features. However, the constant trend in the performance values shown in Figure 9 proves that iFHMM–iHMM is a robust generative model for handling noisy and redundant features. For instance, noisy features for our problem can be generated from errors in semantic parsing, event class extraction, POS tagging, and disambiguation of polysemous semantic frames. To strengthen this statement, we also compare in Table 6 the results obtained by an iFHMM–iHMM model that considers all the feature values associated with an observable object (iFHMM–iHMM$_{all}$) against the iFHMM–iHMM models that use the mIBP sampling scheme and the *unfiltered*, *discrete*, *median*, and *uniform* heuristics. Because of the memory limitation constraints, we performed the experiments listed in Table 6 by selecting only a subset of feature types from the ones that proved to be salient in the HDP experiments. As listed in Table 6, all the iFHMM–iHMM models that used a heuristic approach for selecting feature values significantly outperform the iFHMM–iHMM$_{all}$ model; therefore, this proves that all the feature selection approaches considered in the iFHMM–iHMM framework are able to successfully filter out a significant number of noisy and redundant feature values.

7.4 Error Analysis

We performed an error analysis by manually inspecting both system and gold-annotated data in order to track the most common errors made by our models. One frequent error occurs when a more complex form of semantic inference is needed to find a correspondence between two event mentions of the same individuated event. For instance, because all properties and participants of em_3(*acquisition*) are omitted in Example (1), and no common features exist between em_2(*buy*) and em_3(*acquisition*) to indicate a similarity between these mentions, they will most probably be assigned to different clusters. This example also suggests the need for a better modeling of the discourse salience for event mentions.

Another common error is made when matching the semantic roles corresponding to coreferential event mentions. Although we simulated entity coreference by using various semantic features, the task of matching participants and properties associated with coreferential event mentions is not completely solved. This is because, in many coreferential cases, partonomic relations between semantic roles need to be inferred.[10] Examples of such relations extracted from ECB are *Israeli forces* $\xrightarrow{\text{PART OF}}$ *Israel*, *an Indian warship* $\xrightarrow{\text{PART OF}}$ *the Indian navy*, *his cell* $\xrightarrow{\text{PART OF}}$ *Sicilian jail*. Similarly for event properties, many coreferential examples do not specify a clear location and time interval

9 The selection of features into the increasing fractions of feature types was randomly performed. The fraction corresponding to the 100% experiment in Figure 9 contains all 132 feature types.

10 This observation was also reported in Hasler and Orasan (2009).

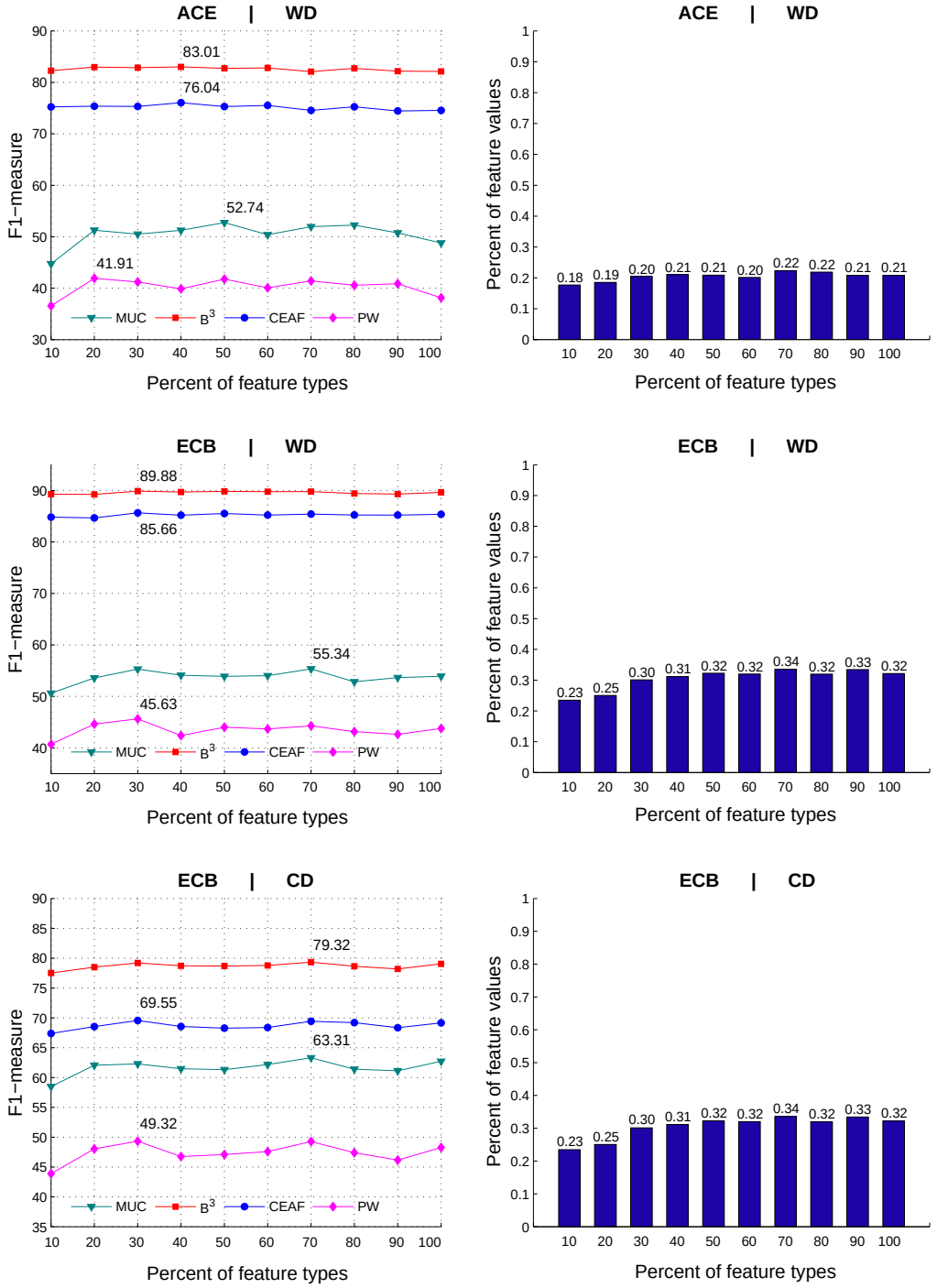

Figure 9
Performance results and feature reduction bars of the iFHMM–iHMM models for various sets of features.

Table 6
Feature non-sampling vs. feature sampling in iFHMM–iHMM models.

	Model configuration	MUC			B^3			CEAF			PW		
		R	P	F	R	P	F	R	P	F	R	P	F
	ACE (within-document event coreference)												
1	iFHMM-iHMM$_{all}$	72.4	30.9	43.3	89.3	39.8	55.0	30.2	68.8	42.0	62.7	9.1	15.9
2	iFHMM-iHMM$_{unfiltered}$	53.5	45.8	49.4	83.3	77.7	80.4	70.6	75.9	73.2	42.1	34.6	38.0
3	iFHMM-iHMM$_{discrete}$	54.3	50.0	52.0	83.8	80.7	82.2	73.0	75.8	74.4	43.9	39.1	41.4
4	iFHMM-iHMM$_{median}$	53.8	48.9	51.2	83.5	80.2	81.8	72.2	75.3	73.7	42.7	38.2	40.3
5	iFHMM-iHMM$_{uniform}$	51.5	47.8	49.6	82.8	80.7	81.7	72.8	75.2	73.9	41.4	39.3	40.3
	ECB (within-document event coreference)												
6	iFHMM-iHMM$_{all}$	64.6	34.0	44.5	89.5	62.5	73.6	53.3	76.5	62.8	60.7	22.9	33.2
7	iFHMM-iHMM$_{unfiltered}$	40.0	76.9	52.4	82.6	96.6	89.0	92.0	79.1	85.1	28.4	75.6	41.0
8	iFHMM-iHMM$_{discrete}$	41.7	77.2	54.0	83.1	96.7	89.4	91.6	79.2	84.9	30.5	79.0	43.9
9	iFHMM-iHMM$_{median}$	39.0	80.0	52.5	82.5	97.3	89.3	92.8	78.9	85.3	29.2	78.8	42.0
10	iFHMM-iHMM$_{uniform}$	40.4	73.4	51.8	82.7	96.0	88.9	91.1	79.0	84.6	29.3	74.9	41.6
	ECB (cross-document event coreference)												
11	iFHMM-iHMM$_{all}$	70.4	54.7	61.5	79.3	54.4	64.5	43.3	61.3	50.7	59.6	26.2	36.4
12	iFHMM-iHMM$_{unfiltered}$	49.3	84.3	62.1	67.2	94.5	78.5	84.7	59.2	69.6	32.8	82.5	46.8
13	iFHMM-iHMM$_{discrete}$	48.8	84.8	61.8	67.6	94.8	78.9	83.8	58.3	68.8	34.3	85.3	48.9
14	iFHMM-iHMM$_{median}$	47.6	86.2	61.4	66.7	95.2	78.4	84.5	57.7	68.5	32.2	83.7	46.3
15	iFHMM-iHMM$_{uniform}$	49.9	82.9	62.2	67.7	93.6	78.4	83.6	59.2	69.2	33.6	79.5	46.9

(e.g., *Jabaliya refugee camp* $\xrightarrow{\text{PART OF}}$ *Gaza*, *Tuesday* $\xrightarrow{\text{PART OF}}$ *this week*). In future work, we plan to build relevant clusters using partonomies and taxonomies such as the WordNet hierarchies built from MERONYMY/HOLONYMY and HYPERNYMY/HYPONYMY relations, respectively.[11]

8. Conclusion

We have described a new class of unsupervised, nonparametric Bayesian models designed for the purpose of solving the problem of event coreference resolution. Specifically, we have shown how already existing models can be extended in order to relax some of their limitations and how to better represent the event mentions from a particular document collection. In this regard, we have focused on devising models for which the number of clusters and the number of feature values corresponding to event mentions can be automatically inferred from data.

Our experimental results for solving the problem of event coreference proved that these models are able to successfully handle such types of requirements on a real data application. Based on these results, we also demonstrated that the new HDP extension, which is able to model observable objects characterized by multiple properties, is a better fit for this type of problem than the original HDP model. Moreover, we believe that the HDP extension can be used for solving clustering problems that involve a small number of feature types and a priori known facts about the salience of these feature

11 This task is not trivial, because if applying the transitive closure on these relations, all words will end up being part of the same cluster with *entity* for instance.

types. On the other hand, when no such prior information is known with respect to the number of feature types, or the total number of features is relatively large, we believe that the iFHMM–iHMM model is a more suitable choice. The main reason is because the new hybrid model is able to perform an automatic selection of feature values. As shown in our experiments, this model was capable of achieving competitive results even when only 2% of feature values were selected from the entire set of features encoded in the ACE, OntoNotes, and ECB data sets.

Acknowledgments
The authors would like to thank the anonymous reviewers, whose insightful comments and suggestions considerably improved the quality of this article.

References

Ahn, David. 2006. The stages of event extraction. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pages 1–8, Sydney.

Allan, James, editor. 2002. *Topic Detection and Tracking: Event-Based Information Organization*. Kluwer Academic Publishers.

Allan, James, Jaime Carbonell, George Doddington, Jonathan Yamron, and Yiming Yang. 1998. Topic detection and tracking pilot study: Final report. In *Proceedings of the Broadcast News Understanding and Transcription Workshop*, pages 194–218, Lansdowne, VA.

Bagga, Amit and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *Proceedings of the 1st International Conference on Language Resources and Evaluation (LREC-1998)*, pages 563–566, Granada.

Bagga, Amit and Breck Baldwin. 1999. Cross-document event coreference: Annotations, experiments, and observations. In *Proceedings of the ACL Workshop on Coreference and its Applications*, pages 1–8, College Park, MD.

Baker, Collin F., Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING-ACL)*, pages 86–90, Montreal.

Beal, Matthew J., Zoubin Ghahramani, and Carl Edward Rasmussen. 2002. The infinite hidden Markov model. In *Advances in Neural Information Processing Systems 14 (NIPS)*, pages 577–584, Vancouver.

Bejan, Cosmin Adrian. 2007. Deriving chronological information from texts through a graph-based algorithm. In *Proceedings of the 20th Florida Artificial Intelligence Research Society International Conference (FLAIRS), Applied Natural Language Processing Track*, pages 259–260, Key West, FL.

Bejan, Cosmin Adrian. 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*, pages 124–129, Coconut Grove, FL.

Bejan, Cosmin Adrian and Sanda Harabagiu. 2008a. A linguistic resource for discovering event structures and resolving event coreference. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC)*, pages 2,881–2,887, Marrakech.

Bejan, Cosmin Adrian and Sanda Harabagiu. 2008b. Using clustering methods for discovering event structures. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*, pages 1,776–1,777, Chicago, IL.

Bejan, Cosmin Adrian and Sanda Harabagiu. 2010. Unsupervised event coreference resolution with rich linguistic features. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1,412–1,422, Uppsala.

Bejan, Cosmin Adrian 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)*, pages 460–463, Prague.

Bejan, Cosmin Adrian, Matthew Titsworth, Andrew Hickl, and Sanda Harabagiu. 2009. Nonparametric Bayesian models for unsupervised event coreference resolution. In *Advances in Neural Information Processing Systems 23 (NIPS)*, pages 73–81, Vancouver.

Blei, David, Andrew Ng, and Michael Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, pages 993–1,022.

Boyd-Graber, Jordan, David Blei, and Xiaojin Zhu. 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)*, pages 1,024–1,033, Prague.

Bryant, Michael and Erik B. Sudderth. 2012. Truly nonparametric online variational inference for hierarchical Dirichlet processes. In *Advances in Neural Information Processing Systems 25 (NIPS)*, pages 2,708–2,716, Lake Tahoe, NV.

Cardie, Claire and Kiri Wagstaf. 1999. Noun phrase coreference as clustering. In *Proceedings of the 1999 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 82–89, College Park, MD.

Chen, Bin, Jian Su, and Chew Lim Tan. 2010a. A twin-candidate based approach for event pronoun resolution using composite kernel. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, pages 188–196, Beijing.

Chen, Bin, Jian Su, and Chew Lim Tan. 2010b. Resolving event noun phrases to their verbal mentions. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 872–881, Cambridge, MA.

Chen, Zheng and Heng Ji. 2009. Graph-based event coreference resolution. In *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing (TextGraphs-4)*, pages 54–57, Singapore.

Chu, Wei, Zoubin Ghahramani, Roland Krause, and David Wild. 2006. Identifying protein complexes in high-throughput protein interaction screens using an infinite latent feature model. In *Pacific Symposium on Biocomputing (PSB-11)*, pages 231–242, Maui, HI.

Cowans, Philip. 2004. Information retrieval using hierarchical Dirichlet processes. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 564–565, Sheffield.

Daumé III, Hal. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 256–263, Prague.

Daumé III, Hal and Daniel Marcu. 2005. A Bayesian model for supervised clustering with the Dirichlet process prior. *Journal of Machine Learning Research (JMLR)*, 6:1551–1577.

Davidson, Donald, 1969. The individuation of events, pages 216–234. In N. Rescher et al., editors, *Essays in Honor of Carl G. Hempel*. Dordrecht: Reidel. Reprinted in D. Davidson, editor, *Essays on Actions and Events*. 2001, Oxford: Clarendon Press.

Davidson, Donald, 1985. *Reply to Quine on Events*. In E. LePore and B. McLaughlin, eds., *Actions and Events: Perspectives on the Philosophy of Donald Davidson*. Blackwell, Oxford, pages 172–176.

de Marneffe, Marie-Catherine, Anna N. Rafferty, and Christopher D. Manning. 2008. Finding contradictions in text. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, pages 1,039–1,047, Columbus, OH.

Denis, Pascal and Jason Baldridge. 2008. Specialized models and ranking for coreference resolution. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP'08)*, pages 660–669, Honolulu, HI.

Doshi-Velez, Finale. 2009. *The Indian Buffet Process: Scalable Inference and Extensions*. Ph.D. thesis, Department of Engineering, University of Cambridge.

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

Ferguson, Thomas S. 1973. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1(2):209–230.

Finkel, Jenny Rose, Trond Grenager, and Christopher Manning. 2007. The infinite tree. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 272–279, Prague.

Finkel, Jenny Rose and Christopher Manning. 2008. Enforcing transitivity in coreference resolution. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT), Short Papers*, pages 45–48, Columbus, OH.

Fox, E. B., E. B. Sudderth, and A. S. Willsky. 2007. Hierarchical Dirichlet processes for tracking maneuvering targets. In *Proceedings of International Conference on Information Fusion*, pages 1,415–1,422, Quebec.

Geman, Stuart and Donald Geman. 1984. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741.

Ghahramani, Zoubin, T. L. Griffiths, and Peter Sollich, 2007. Bayesian Nonparametric Latent Feature Models. In *Bayesian Statistics 8*, edited by J. M. Bernardo et al., pages 201–225. Oxford University Press.

Ghahramani, Zoubin and Michael Jordan. 1997. Factorial hidden Markov models. *Machine Learning*, 29:245–273.

Goldwater, Sharon, Thomas L. Griffiths, and Mark Johnson. 2006. Contextual dependencies in unsupervised word segmentation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 673–680, Sydney.

Görür, Dilan, Frank Jäkel, and Carl Edward Rasmussen. 2006. A choice model with infinitely many latent features. In *Proceedings of the 23rd Annual International Conference on Machine Learning (ICML)*, pages 361–368, Pittsburgh, PA.

Griffiths, Thomas and Mark Steyvers. 2004. Finding scientific topics. In *Proceedings of the National Academy of Sciences*, pages 5,228–5,235.

Griffiths, Tom and Zoubin Ghahramani. 2005. Infinite latent feature models and the Indian buffet process. Technical Report 2005-10, Gatsby Computational Neuroscience Unit, University College London.

Griffiths, Tom and Zoubin Ghahramani. 2006. Infinite latent feature models and the indian buffet process. In *Advances in Neural Information Processing Systems 18 (NIPS)*, pages 475–482, Vancouver.

Haghighi, Aria and Dan Klein. 2007. Unsupervised coreference resolution in a nonparametric Bayesian model. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 848–855, Prague.

Haghighi, Aria and Dan Klein. 2009. Simple coreference resolution with rich syntactic and semantic features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1,152–1,161, Singapore.

Haghighi, Aria and Dan Klein. 2010. Coreference resolution in a modular, entity-centered model. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 385–393, Los Angeles, CA.

Haghighi, Aria, Andrew Ng, and Christopher Manning. 2005. Robust textual inference via graph matching. In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT-EMNLP)*, pages 387–394, Vancouver.

Hasler, Laura and Constantin Orasan. 2009. Do coreferential arguments make event mentions coreferential? In *Proceedings of the 7th Discourse Anaphora and Anaphor Resolution Colloquium (DAARC 2009)*, pages 151–163, Goa.

He, Tian. 2007. Coreference resolution on entities and events for hospital discharge summaries. Master's thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.

Humphreys, Kevin, Robert Gaizauskas, and Saliha Azzam. 1997. Event coreference for information extraction. In *Proceedings of the Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts, 35th Meeting of ACL*, pages 75–81, Madrid.

Kong, Fang and Guodong Zhou. 2011. Improve tree kernel-based event pronoun resolution with competitive information. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1,814–1,819, Barcelona.

LDC-ACE. 2005. ACE (Automatic Content Extraction) English Annotation Guidelines for Events, version 5.4.3 2005.07.01. LDC Catalog Number: LDC2006T06.

LDC-ON. 2007. OntoNotes Release 2.0. LDC Catalog Number: LDC2008T04.

Lee, Heeyoung, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky. 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*, pages 489–500, Jeju Island.

Li, Fei-Fei and Pietro Perona. 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) - Volume 2*, pages 524–531, San Diego, CA.

Liang, Percy, Slav Petrov, Michael Jordan, and Dan Klein. 2007. The infinite PCFG using hierarchical Dirichlet processes. In *Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP/CoNLL)*, pages 688–697, Prague.

Lowe, John B., Collin F. Baker, and Charles J. Fillmore. 1997. A frame-semantic approach to semantic annotation. In *Proceedings of the SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*, pages 18–24, Washington, DC.

Luo, Xiaoqiang. 2005. On coreference resolution performance metrics. In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (EMNLP-2005)*, pages 25–32, Vancouver.

Luo, Xiaoqiang, Abe Ittycheriah, Hongyan Jing, Nanda Kambhatla, and Salim Roukos. 2004. A mention-synchronous coreference resolution algorithm based on the bell tree. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 135–142, Barcelona.

Malpas, Jeff. 2009. Donald Davidson. In *the Stanford Encyclopedia of Philosophy (Fall 2009 Edition)*, edited by Edward N. Zalta. Available at `http://plato.stanford.edu/archives/fall2009/entries/davidson/`.

Meeds, Edward, Zoubin Ghahramani, Radford Neal, and Sam Roweis. 2006. Modeling dyadic data with binary latent factors. In *Advances in Neural Information Processing Systems 19 (NIPS)*, pages 977–984, Vancouver.

Miller, Kurt, Thomas Griffiths, and Michael Jordan. 2008. The phylogenetic Indian buffet process: A non-exchangeable nonparametric prior for latent features. In *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 403–410, Helsinki.

Narayanan, Srini and Sanda Harabagiu. 2004. Question answering based on semantic structures. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, pages 693–701, Geneva.

Neal, Radford M. 2003. Slice Sampling. *The Annals of Statistics*, 31:705–741.

Ng, Vincent. 2008. Unsupervised models for coreference resolution. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 640–649, Honolulu, HI.

Palmer, Martha, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–105.

Poon, Hoifung and Pedro Domingos. 2008. Joint unsupervised coreference resolution with Markov logic. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 650–659, Honolulu, HI.

Pustejovsky, James, 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 the Fifth International Workshop on Computational Semantics (IWCS)*, pages 337–353, Tilburg.

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

Quine, W. V. O., 1985. Events and Reification. In E. LePore and B. P. McLaughlin, editors, *Actions and Events: Perspectives on the Philosophy of Donald Davidson*. Blackwell, Oxford, pages 162–171. Reprinted in R. Casati and A. C. Varzi, editors, *Events*. 1996, Aldershot, Dartmouth, pages 107–116.

Rabiner, Lawrence R. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, pages 257–286.

Raghunathan, Karthik, Heeyoung Lee, Sudarshan Rangarajan, Nate Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. 2010. A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 492–501, Cambridge, MA.

Rahman, Altaf and Vincent Ng. 2011. Coreference resolution with world knowledge. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 814–824, Portland, OR.

Reisinger, Joseph and Marius Paşca. 2009. Latent variable models of concept-attribute attachment. 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*, pages 620–628, Singapore.

Sivic, Josef, Bryan Russell, Alexei Efros, Andrew Zisserman, and William Freeman. 2005. Discovering object categories in image collections. In *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV)*, pages 370–377, Beijing.

Sivic, Josef, Bryan Russell, Andrew Zisserman, William Freeman, and Alexei Efros. 2008. Unsupervised discovery of visual object class hierarchies. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, Anchorage, AK.

Stoyanov, Veselin, Nathan Gilbert, Claire Cardie, and Ellen Riloff. 2009. Conundrums in noun phrase coreference resolution: Making sense of the state of the art. 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*, pages 656–664, Singapore.

Sudderth, Erik B., Antonio Torralba, William T. Freeman, and Alan S. Willsky. 2008. Describing visual scenes using transformed objects and parts. *International Journal of Computer Vision*, 77:291–330.

Teh, Yee Whye, Michael Jordan, Matthew Beal, and David Blei. 2006. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581.

Van Gael, Jurgen, Y. Saatci, Yee Whye Teh, and Zoubin Ghahramani. 2008. Beam sampling for the infinite hidden Markov model. In *Proceedings of the 25th Annual International Conference on Machine Learning (ICML)*, pages 1,088–1,095, Helsinki.

Van Gael, Jurgen, Yee Whye Teh, and Zoubin Ghahramani. 2008. The infinite factorial hidden Markov model. In *Advances in Neural Information Processing Systems 21 (NIPS)*, pages 1697–1704, Vancouver.

Vilain, Marc, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of MUC-6*, pages 45–52, Columbia, MD.

Wang, Chong, John Paisley, and David Blei. 2011. Online variational inference for the hierarchical Dirichlet process. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 752–760, Ft. Lauderdale, FL.