Title
Unsupervised Models of Entity Reference Resolution

Author
Haghighi, Aria Delier

Publication Date
2010

Peer reviewed|Thesis/dissertation

Unsupervised Models of Entity Reference Resolution

by

Aria Delier Haghighi

A dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:

Professor Dan Klein, Chair
Professor Stuart Russell
Professor Marti Hearst

Fall 2010

Unsupervised Models of Entity Reference Resolution

by

Aria Delier Haghighi

Abstract

Unsupervised Models of Entity Reference Resolution

by

Aria Delier Haghighi

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Dan Klein, Chair

A central aspect of natural language understanding consists of linking information across multiple sentences and even combining multiple sources (for example: articles, conversations, blogs and tweets). Understanding this global information structure requires identifying the people, objects, and events as they evolve over a discourse. While natural language processing (NLP) has made great progress on sentence-level tasks such as parsing and machine translation, far less progress has been made on the processing and understanding of large units of language such as a document or a conversation.

The initial step in understanding discourse structure is to recognize the entities (people, artifacts, locations, and organizations) being discussed and track their references throughout. Entities are referred to in many ways: with proper names (`Barack Obama`), nominal descriptions (`the president`), and pronouns (`he` or `him`). Entity reference resolution is the task of deciding to which entity a textual mention refers.

Entity reference resolution is influenced by a variety of constraints, including syntactic, discourse, and semantic constraints. Even some of the earliest work (Hobbs, 1977, 1979), has recognized that while syntactic and discourse constraints can be declaratively specified, semantic constraints are more elusive. While past work has successfully learned many of the syntactic and discourse cues, there has yet to be an entity reference resolution system that exploits semantic cues and operationalizes these observations into a coherent model.

This dissertation presents unified statistical models for entity reference resolution that can be learned in an unsupervised way (without labeled data) and models soft semantic constraints probabilistically along with hard grammatical constraints. While the linguistic insights which underlie this model have been observed in some of the earliest anaphora resolution literature (Hobbs, 1977, 1979), the machine learning techniques which allow these cues to be used collectively and effectively are relatively recent (Blei et al., 2003; Teh et al., 2006; Blei and Frazier, 2009). In particular, our models use recent insights into Bayesian nonparametric modeling (Teh et al., 2006) to effectively learn entity partition structure when the number of entities is not known ahead of time. The primary contribution of this dissertation is combining the linguistic observations of past researchers with modern structured machine

learning techniques. The models presented herein yield state-of-the-art reference resolution results against other systems, supervised or unsupervised.

Professor Dan Klein
Dissertation Committee Chair

To Ramsey: To this day, part of what inspires me is thinking about what you would have done if you were still here.

To my mom: part of what inspires me is the things you have done to get me here.

To Sara: part of what inspires me is thinking about what we will do.

Contents

List of Figures

List of Tables

Acknowledgements

There are many people I have to thank for helping me to get here.

First, my advisor Dan Klein. Without hyperbole, Dan is the most impressive, intelligent, and creative researcher I know. Simply put, he is the best advisor and teacher one could hope for.

Also, to the starting class of the Berkeley NLP group: Slav Petrov, John DeNero, Percy Liang, and Alexandre Bouchard-Cote. As with the advisor, I really lucked out with the initial group. I think we all made each other better researchers and I will miss all of you dearly. John DeNero: we never did make our millions, but you and Jessica definitely enriched my life. To those who joined later: Adam Pauls, David Burkett, John Blitzer, Taylor Berg-Kirkpatrick, David Hall Mohit Bansal, Aditi Muralidharan, and Anna Rafferty. You are all great and I wish I had more time with you. To Blitzer and Renee: rock band?

Special thanks to Chris Manning at Stanford. I would not be in NLP if it were not for how much I admired Chris and the way he thinks about problems. I suspect there are many people whose interest in NLP originate with Chris; I am one of many. Also to Andrew Ng and Daphne Koller: you both are amazing researchers and teachers. I certainly would not be here without the technical background and exemplar you imparted upon me.

Thanks also to Regina Barzilay, Eugene Charniak, Fernando Pereira, Lucy Vanderwende, Andrew McCallum, Marti Hearst, Hal Daume, Jason Eisner, Bob Moore, Chris Quirk, and many others for their feedback, support, and advice on this and related work.

To my fellow NLP grad student colleagues: Jenny Finkel, Micha Elsner, Hoifung Poon, and many others. I admire and can't wait to work with all of you.

To my friends: Shawn Standefer, Alex Gurevich, Jason Wolfe, Michael Felker, Scott Lulovics, Guy Iseley, Sam Henry, Elan Bar, Denis Kuo, and Ryan Ruby. You all are so great.

To my siblings: Maria, Michael, and Ali. Sad to no longer be on the same coast as you guys.

To my mom and Sara: Go read the dedication!

Chapter 1

Introduction

Most natural language processing (NLP) work examines language through a microscope, analyzing structure at or below the level of individual sentences. However, most of the information that we care about exists more globally, linking multiple sentences and even combining multiple sources (for example: articles, conversations, blogs and tweets). Understanding this global information requires identifying the people, objects, and events as they evolve over a discourse. While NLP has made great progress on sentence-level tasks such as parsing and machine translation, much less progress has been made on the processing and understanding of large units of language such as a document or a conversation. Understanding this discourse-level of structure is ultimately essential for building systems capable of full natural language understanding.

The initial step in understanding discourse structure is to recognize the entities (people, artifacts, locations, and organizations) being discussed and track their references throughout. Entities are referred to in many ways: with proper names (`Barack Obama`), nominal descriptions (`the president`), and pronouns (`he` or `him`). These references, which we call *mentions*, are typically realized as noun phrases (NP). Entity reference resolution is the task of deciding to which entity a textual mention refers. For a concrete example of this task, see the excerpts in Figure 1.1. In these examples, textual mention boundaries are denoted by brackets, which we initially treat as given. The entity to which a given mention refers is indicated by color as well as subscript. For instance in Figure 1.1(a), the mentions `Weir Group`, `corporation`, and `whose` are proper, nominal, and pronominal references (respectively) to the *Weir Group* entity.

There has been much research on the task of automatic entity resolution, often called coreference resolution. Much of the earliest work, (Hobbs, 1977; Sidner, 1979), focused on the sub-problem of resolving pronoun reference to antecedent references; this sub-task is often called *anaphora resolution*. Hobbs (1977) presented a relatively simple algorithm which for a given referential pronoun searched for a nearby antecedent reference which was compatible with the pronoun in number and gender. For instance, in Figure 1.1(b), the

The [Weir Group]$_1$, [whose]$_1$ [headquarters]$_2$ is in the [U.S]$_3$, is a specialized [corporation]$_1$. This [power plant]$_4$,[which]$_4$, will be situated in [Jiangsu]$_5$, has a large generation [capacity]$_6$.

(a)

[The castle]$_1$ in [Camelot]$_2$ remained [the residence]$_3$ of [the king]$_4$ until 536 when [he]$_4$ moved [it]$_3$ to [London]$_5$.

(b)

[Apple]$_1$ says [iPad]$_2$ [sales]$_3$ have topped [two million]$_4$ since [it]$_2$ was released.

(c)

Figure 1.1. Examples of entity reference resolution. The heads of textual mentions are denoted by brackets and the entity identity is indicated by color and subscript assignments. Sometimes entity resolution requires semantic information. In (b), we present an example from Hobbs (1977) meant to illustrate how semantic compatibility can benefit pronoun resolution. In example (c), we present another example of this phenomena.

pronoun `he` is singular and male in gender and any potential antecedent must not conflict with those values; this rules out `castle`, `Camelot`, and `residence` since those words, as inanimate objects, are neuter gender in English. Subject to these constraints, the correct antecedent of the pronoun tended to be close to the pronoun. Hobbs (1977) noted that this simple procedure seemed to perform quite well,[1] but noted that many errors were the result of *semantic incompatibility* between pronoun and antecedent. An example sentence that Hobbs (1977) uses to illustrate this is presented in Figure 1.1(b). The pronoun `it` is the direct object of `moved`. This context is incompatible with `castle` which is a large object and unlikely to be moved. The pronoun is incompatible with `London` and `Camelot` since those are specific locations which cannot be moved. Hobbs (1977) provides several other examples where the context along with our semantic knowledge of the world can aid pronoun reference

[1]See Chapter 5 where we present a system which elaborates on this simple approach and confirm its findings.

disambiguation. However, Hobbs (1977) does not have a concrete procedure for obtaining the relevant semantics facts for utilizing this information in a pronoun resolution system.

As another example of this phenomenon consider correctly resolving the pronoun `it` in Figure 1.1(c). In general, pronouns are coreferent with a 'close' mention. In this example however, there are multiple mentions between the pronoun and the correct antecedent mention `iPad`. Typically, speakers limit potential referring entities according to evidence from usage. In this case, whatever entity to which `it` refers is something which can be released. The entity type of the closest mention `two million` is a numerical quantity which are, in general, unlikely to be released. In contrast, the entity type of the mention `iPad` is an electronic product which is likely to be the object of `released`. However, it is not clear from this single usage of the term `iPad` that is an electronic device. Discovering this information might require synthesizing information across multiple usages of the entity from different discourses.

There has been much linguistic research on the grammatical constraints which either ensure or disallow NP coreference. Lees and Kilma (1963) and Langacker (1969) present constraints on which the relative positions of a pronoun and a potential antecedent prohibit (or require) the use of a reflexive pronoun (e.g., `himself`). While much work in automatic reference resolution has benefitted from this research,[2] there has been less work on characterizing the more heuristic constraints which underlie the semantic compatibility in anaphora resolution. A notable exception is Kehler et al. (2008), who present psycholinguistic experiments which affirm that coherence-relationships, such as in Hobbs (1979), can override traditional grammatical preferences in pronoun interpretation.

Despite this recognition for the need to incorporate soft semantic constraints in reference resolution, there has yet to be an entity reference resolution system which operationalizes these observations into a coherent model. This dissertation presents unified statistical models which perform entity reference resolution; these models can be learned in an unsupervised way (without labeled data) and they are capable of capturing soft semantic constraints probabilistically along with hard grammatical constraints. The fact that it is unsupervised allows it to be trained on large unannotated datasets which facilitate learning broad-coverage semantic information. In particular, we present a model (see Chapter 6) that is able to handle examples like the *iPad* example presented earlier.[3] This model yields the state-of-the-art reference resolution results against other systems, supervised or unsupervised.

While the linguistic insights which underlie this model have been observed in some of the earliest anaphora resolution literature (Hobbs, 1977, 1979), the machine learning techniques which allow these cues to be used collectively and effectively are relatively recent (Blei et al., 2003; Teh et al., 2006; Blei and Frazier, 2009). In particular, our models use recent insights into Bayesian non-parametric modeling (Teh et al., 2006) to effectively learn entity partition structure when the number of entities is not known ahead of time. The primary contribution of this dissertation is combining the linguistic observations of past researchers with modern structured machine learning techniques.

[2]See Chapter 5, where we exploit many of this constraints.

[3]While in principle, this model can handle the Hobbs (1977) example, the semantic distinction is in practice probably too subtle to learn.

The dissertation as outlined as follows: In Chapter 2, we present a more concrete description of the task as well as evaluation metrics. In Chapter 3, we present a brief overview of other automatic reference resolution approaches as well as related problems. In Chapter 4, we present a Bayesian non-parametric generative model of entities and mentions, which can be learned in an unsupervised fashion. The primary purpose of this chapter is to present a modeling framework capable of capturing entity reference patterns. In Chapter 5, we present a simple deterministic system whose goal is to identify the key syntactic and semantic linguistic factors relevant to entity resolution. In Chapter 6, we present a model which combines the statistical modeling framework presented in Chapter 4 with the empirical insights made in Chapter 5. Finally, in Chapter 7, we present an application of the modeling framework described in Chapter 6 to the task of template extraction for information extraction and demonstrate the efficacy of our entity resolution on an external task.

Chapter 2

Entity Reference Resolution Task Overview

In this chapter we present a brief overview of the entity reference resolution task, also known as coreference resolution. We introduce the task and basic terms we use throughout (Section 2.1) as well as fully describe the evaluation metrics (Section 2.2). Finally, we discuss potential applications of entity reference resolution to other natural language processing problems in Section 2.3.

2.1 Task Definition

We assume we are given an input document D, for which we are given a set of *mentions* $m_1, \ldots, m_n$; we use $\mathbf{m}$ to denote the sequence of mentions. Each mention m_i specifies a span of tokens within the document text. A further constraint is that that the text of the mention refers to an entity (person, place, vehicle, organization, etc.) as opposed to an event.[1] Typically these entity mentions are realized as noun phrases (NP). The source of these given mentions will either come from annotated data, a setting we call the **gold mention** setting, or will be automatically detected, the **system setting**. In this work, we explore the gold mention setting in Chapters 4 and 5 and the system setting in Chapters 6 and 7. Anytime we explore the system setting, we use simple deterministic techniques to detect mentions from syntactic parse trees (see Section 6.5.2 for a discussion of how this is done).

[1]There is not a principled reason to restrict mentions or entities to such objects. In fact, the distinction between entity and non-entity can sometimes be difficult to discern. The ambiguity of this restriction can and does lead inconsistent choices when annotators label entity mentions.

The goal of entity resolution is to cluster a subset of document mentions such that all mentions within a cluster refer to the same underlying entity. Concretely, an entity resolution system for a given document proposes a mention clustering $C_1, \ldots, C_k$, where each $C_i \subseteq \mathbf{m}$ and no C_i's contain common elements. This proposed clustering $\mathbf{C} = (C_1, \ldots, C_k)$ is evaluated against the true annotated clustering, which we denote by $\mathbf{C}^* = (C_1^*, \ldots, C_{k^*}^*)$. Importantly, the number of clusters that are proposed (k) and those annotated (k^*) may not be identical. In the case of gold mention evaluation, the set of mentions in the proposed and true clustering are identical. However in the system setting, the set of mentions can, and often do, differ. We defer discussion of evaluating the system setting until Section 2.2.2.

There are many kinds of information a system may associate with a mention in order to facilitate entity resolution, but are not strictly part of the task, but nonetheless crucial to most systems. Typically, systems associate a **mention type** with each mention. There are three basic mention types: proper, nominal, and pronominal. Proper mentions are canonically names of an entity, and in most contexts are used to unambiguously identify an entity. For instance, the mention `Barack Obama`, is a proper mention reference to the entity *Barack Obama*. Nominal mentions are descriptions of an entity. For instance, `the 44th president of the United States of America`, is a definite description of the `Barack Obama` entity. Some nominal descriptions such as the `the 43rd president of the United States of America` uniquely identify an entity, but others such as the `the company` can describe many entities. In general, a given nominal mention restricts the set of entities to which it can refer. Typically, nominal mentions both invoke an entity as well as provide information about that entity, such as their occupation, or nationality. Pronominal mentions consist of the pronouns of a language and their primary purpose is to refer to a discourse entity (see Section 2.1.2 for a class of exceptions); for instance the pronoun `he` to refer to *Barack Obama*. Depending on the language, a pronoun carries different features (often called *phi-features* in the linguistic literature), which constrain potential coreferring mentions. In English, pronouns carry a gender feature (*male*, *female*, and *neuter*), which can limit the scope of compatible antecedents. In other languages, such as Farsi, pronouns do not carry any gender information and this information cannot be used to limit reference resolution hypothesis. Assigning a mention to a mention type can be done deterministically assuming we have access to mention tokens. See Section 5.3 for a description of how this is done.

2.1.1 Why Not Call it Coreference Resolution?

While many refer to this task as coreference resolution, we prefer the name entity reference resolution because the referents we consider are entities, rather than events or abstract things. This restriction strongly influences several modeling and pre-processing choices and is important to reinforce in the task name.

2.1.2 Non-referring Pronouns

Not all pronouns are referential. Consider the pronoun in: `It is raining`. It is not clear that the pronoun of this sentence refers to any entity. Such pronouns are often called *pleonastic* or *dummy* pronouns. Previous work have discussed this issue and proposed successful techniques for detecting non-referential pronouns (Hobbs, 1977; Lappin and Leass, 1994; Muller, 2006; Denis and Baldridge, 2007; Bergsma et al., 2008). These non-referential pronouns are distinct from pronouns which do refer but not to entities, but instead to events: `We argued for hours. It was awful.` The `it` here does refer, but not to an entity which we will consider in this work. In the gold mention setting, this of course is not an issue. However, in the system setting experiment (see Chapter 6), we allow pronouns to be non-referential but do not distinguish between these two cases.

2.2 Evaluation Metrics

Abstractly, evaluating entity resolution involves comparing a proposed mention clustering $\mathbf{C}$ with the true mention clustering $\mathbf{C}^*$. A further issue is that in the system setting, the mentions in the proposed clustering $\mathbf{C}$ may not be an identical set of mentions. Therefore when evaluating the system setting, we must also have a way to match a proposed mention to a gold mention as well as know how to deal with mentions missing from the proposal as well as spurious mentions which cannot be matched to the true set of mentions (see Section 2.2.2 for more details). All of these metrics are *intrinsic* evaluation which measures the quality of the induced clusterings. See Chapter 7, where we evaluate the model described in Chapter 6 on the *external* task of template information extraction.

There are several metrics used in the entity resolution literature and no single metric is recognized as universally best. We describe the most common metrics which we utilize. We initially describe these evaluation metrics (Section 2.2.1) in the context of the gold mention setting where we assume that the set of mentions in the true and proposed clusterings are identical. In Section 2.2.2, we discuss adopting these evaluation metrics to the system mention setting.

2.2.1 Gold Mention Evaluation

We will use $\mathbf{C}^* = (C_1^*, \ldots, C_{k^*}^*)$ to denote the true mention clustering and $\mathbf{C} = (C_1, \ldots, C_k)$ the proposed system clustering. In this section, we assume the gold mention setting, so that the set of true and proposed mentions are identical; thus $\mathbf{C}$ and $\mathbf{C}^*$ are clusterings over the same set of elements. We will use $\mathbf{C}(m)$ to denote the unique proposed cluster which contains mention m; similarly $\mathbf{C}^*(m)$ denotes the unique true cluster in the true clustering which contains mention m.

All the metrics we propose are composed of a precision (p) and recall (r) for either each

mention or each document. In general, a precision measures how much of what we predict is correct. Recall measures how much of the true annotation did we recover. The specific definitions of precision and recall will vary according to evaluation metric. We compute precision and recall for an entire corpus by averaging precision and recall across all mentions and documents in the corpus. We then compute an F_1 measure in the standard way by taking the harmonic mean of the precision and recall: $\frac{2pr}{p+r}$.

MUC F-Measure

The MUC F-Measure of a document, introduced in Vilain et al. (1995), is the harmonic mean of the MUC precision and MUC recall. The MUC recall measures the number of system cluster merging operations needed to cover each true cluster. Concretely, the MUC recall is given by,

$$\texttt{MucRecall}(\mathbf{C}^*, \mathbf{C}) = \frac{\sum_{C^* \in \mathbf{C}^*} |C^*| - \texttt{CoverSize}(C^*, \mathbf{C})}{\sum_{C^* \in \mathbf{C}^*} |C^*| - 1} \tag{2.1}$$

where $\texttt{CoverSize}(C^*, \mathbf{C})$ is the number of elements of $\mathbf{C}$ needed to cover C^* and is given by $|\cup_{m \in C^*} \mathbf{C}^{-1}(m)|$; note that $\texttt{CoverSize}(C^*, \mathbf{C}) \geq 1$ and $\leq |\mathbf{C}$, so the recall lies in the $[0, 1]$ range. The MUC precision is obtained by swapping the roles of true and system clusters in Equation 2.1:

$$\texttt{MucPrecision}(\mathbf{C}^*, \mathbf{C}) = \frac{\sum_{C \in \mathbf{C}} |C| - \texttt{CoverSize}(C, \mathbf{C}^*)}{\sum_{C \in \mathbf{C}} |C| - 1} \tag{2.2}$$

The MUC F-measure has a major well-known deficiency: it tends to favor systems which err on the side of merging true clusters rather than splitting them. In the extreme case, if the system proposes a single cluster for all mentions, it gets a perfect MUC precision and recall. Nonetheless, if the number of proposed system clusters is not dramatically less than the number of true clusters, the metric is generally informative.

B^3 F-Measure

The B^3 F-Measure, introduced in Bagga and Baldwin (1998), assigns a precision and recall to each true mention; averaging these precision and recalls across true mentions yields precision and recall statistics for the document. The B^3 precision for a true mention m is given by,

$$B^3 - \texttt{precision}(m) = \frac{|\mathbf{C}(m) \cap \mathbf{C}^*(m)|}{|\mathbf{C}(m)|} \tag{2.3}$$

The mention precision measures the fraction of mentions we propose are coreferent with m over the number we propose are coreferent. The recall is similarly defined,

$$B^3 - \texttt{recall}(m) = \frac{|\mathbf{C}(m) \cap \mathbf{C}^*(m)|}{|\mathbf{C}^*(m)|} \tag{2.4}$$

The document-level B^3 precision and recall are the average of mention precisions and recalls. Note that mention-level precision and recall figures are reported for true mentions. The B^3 metric does not naturally incorporate proposed spurious mentions. However, there are two variants of B^3 due to Stoyanov et al. (2009) which we discuss in Section 2.2.2.

Pairwise F-measure

The pairwise F-measure represents the true and system clusterings as a bag of coreferent mention pairs; precision and recall are measured in the standard way. Concretely,

$$\texttt{Pairs}(\mathbf{C}) = \{(m, m') | m \neq m', \text{ and } m, m' \in C, \text{ for some } C \in \mathbf{C}\} \tag{2.5}$$

Then pair precision is given by,

$$\texttt{Pair-Precision}(\mathbf{C}, \mathbf{C}^*) = \frac{|\texttt{Pairs}(\mathbf{C}) \cap \texttt{Pairs}(\mathbf{C}^*)|}{|\texttt{Pairs}(\mathbf{C})|} \tag{2.6}$$

Similarly, the pair recall is given by,

$$\texttt{Pair-Recall}(\mathbf{C}, \mathbf{C}^*) = \frac{|\texttt{Pairs}(\mathbf{C}) \cap \texttt{Pairs}(\mathbf{C}^*)|}{|\texttt{Pairs}(\mathbf{C}^*)|} \tag{2.7}$$

A shortcoming of the pairwise F-measure is that it can over-penalize a system for failing to merge or split a proposed cluster. Suppose a system splits a true cluster into two proposed clusters each of size n. This counts as $O(n^2)$ pairwise recall errors against the system. Splitting the true cluster may simple result from failing to recognize a nominal description of an entity and associating subsequent compatible pronouns with this cluster.

CEAF F-measure

The CEAF F-measure, introduced by Luo (2005), assumes a similarity function $s(C, C')$ between two clusters; the CEAF precision and recall depends on the max-matching between true and proposed clusterings:

$$a^* = \max_{a \in \mathcal{A}} \sum_{C^* \in \mathbf{C}^*} s(C^*, a(C^*)) \tag{2.8}$$

where $\mathcal{A}$ is the set of matchings from true to proposed clusterings and $a(C^*)$ is the element of $\mathbf{C}$ that C^* is matched to. Given this mapping, the CEAF precision for a document is given by,

$$\texttt{CEAF-Precision}(\mathbf{C}^*, \mathbf{C}) = \frac{\sum_{C^* \in \mathbf{C}^*} s(C^*, a^*(C^*))}{\sum_{C \in \mathbf{C}} s(C, C)} \tag{2.9}$$

The recall is given by,

$$\texttt{CEAF-Recall}(\mathbf{C}^*, \mathbf{C}) = \frac{\sum_{C^* \in \mathbf{C}^*} s(C^*, a^*(C^*))}{\sum_{C^* \in \mathbf{C}^*} s(C^*, C^*)} \tag{2.10}$$

The similarity function between clusters we use are the size of the cluster intersection, i.e $s(C, C') = |C \cap C'|$. This choice of similarity function corresponds to ϕ_3 in Luo et al. (2004); Luo et al. (2004) proposed other similarity functions, but this is the variant that has been reported in the literature (Rahman and Ng, 2009).

2.2.2 System Mention Evaluation

In the system mention setting, the set of mentions in the true and proposed clusterings may not be the same. We first must match proposed mention to true mentions based upon the token spans associated with each (the details on how to match mentions are described Section 6.5). Once we have matched some proposed mentions to true mentions, there are two other kinds of mentions that remain: **missing mentions** are true mentions which have not been matched by a proposed mention and **spurious mentions** are proposed mentions which cannot be matched to a true mentions. Each of the metrics discussed so far must be adapted to handle both spurious and missing mentions:

MUC F-Measure

The MUC measure can be used if we supplement the proposed clustering with singleton clusters for missing mentions. Similarly, we add singleton clusters in the true clustering for each spurious proposed mention. Thus if we fail to propose a mention which matches a given true mention, the $\texttt{CoverSize}(\cdot, \cdot)$ in Equation 2.1 is incremented by one to cover the singleton cluster containing the missed mention. The MUC precision term (Equation 2.2) is similarly penalized for each spurious mention.

B^3 F-Measure

There are two variants, proposed by Stoyanov et al. (2009), of the B^3 algorithm for the system mention setting. The first variant B^3All computes the mention precision as follows. If the proposed mention can be matched to an annotated mention, the precision is identical to the normal B^3 formula (Equation 2.3). If the mention m is spurious, the precision is given by $|\mathbf{C}(m)|^{-1}$, the size of the proposed cluster containing the spurious mention. So the larger the cluster to which the spurious mention is included, the lower the precision. The precision for the document is obtained, as before, by averaging over all proposed mentions.[2] Similarly for the B^3 recall, the recall for a missing mention m is given by $|\mathbf{C}^*(m)|^{-1}$, the size of the true cluster containing the missing mention. Note that this implies no penalty for failing to annotate a singleton true mention.

[2]Note that if a spurious mention is in a singleton cluster, the precision associated with it is 1, which will not harm the precision, but of course can inflate it. Stoyanov et al. (2009) does not discuss this possibility, but in our implementation we removed singleton spurious mentions from evaluation to avoid inflation. In practice, we did not find this affected results significantly.

Stoyanov et al. (2009) also proposed the B^3None variant, which simply discards spurious mentions from the proposed clustering, but penalizes missing mentions by setting the B^3 − `Recall` to 0.0 for missing mentions; thus one does incur a penalty for missing a mention in a singleton cluster.

Pairwise F-measure

The only modification to the pairwise F-measure is that when computing the precision we only consider proposed mention pairs where at least one has been matched to an annotated mention. Concretely, if we denote the set of true mentions by $\mathcal{M}$, we substitute the mention pair definition in Equation 2.5, with,

$$\texttt{Pairs}(\mathbf{C}) = \{(m, m') | m \neq m', \text{ and } m, m' \in C, \text{ for some } C \in \mathbf{C}, \text{ and } m \text{ or } m' \in \mathcal{M}\}$$

We choose not to penalize a system for declaring that two non-annotated mentions are coreferent. If we mark an annotated mention as coreferent with an unannotated mention, we do incur a precision penalty. The definition of $\texttt{Pairs}(\mathbf{C}^*)$ is unaltered from the original.

2.3 Potential Applications

There are several potential applications of entity reference resolutions across NLP. In Chapter 7, we present an application of our main model, see Chapter 6, to template information extraction. We provide a brief overview here over other potential applications:

Question Answering

Although most approaches to question answering (QA) involves search for a declarative variant of a question (Ravichandran and Hovy, 2002), much of the information we glean requires entity reference resolution to acquire. Consider,

> President Barack Obama received the Serve America Act after congress' vote. He signed the bill last Thursday. The president said it would greatly increase service opportunities for the American people.

The answer to the question: *Who signed the Serve America Act?* is technically contained within passage, but unless one resolved `he` and `the bill` to the appropriate entities, it cannot be extracted from this text. Depending on the domain and the popularity of the answer being looked for, the pattern matching approach (Ravichandran and Hovy, 2002) may not be applicable. See Vicedo and Ferrandez (2000) for a further discussion.

Machine Translation

Since the pronouns of different languages mark for different properties of the underlying entity, translating between some languages correctly requires tracking an entity and its properties. Consider translating the following sentence of Farsi (and gloss below) into English:[3]

او	قانونرا	امظاءکرد
oo	ganoon-ra	emza-kard
person-pro	*law-obj*	*signed-past*

The first (leftmost) word of the excerpt is a Farsi animate pronoun which does not mark for the gender of the person. This sentence cannot be accurately translated into English in isolation, since a faithful translation must render the pronoun as a `he` or `she` and this information isn't discernible from just this sentence. Of course, sentences such as this are not uttered in isolation, a translator would, using knowledge of the underlying entity reference, translate the pronoun appropriately. Automatically doing so requires entity resolution. Despite problems such as this, the vast majority of statistical machine translation system translate sentences independently.

Summarization

Entity reference resolution stands to benefit textual summarization in many ways. One way explored by Nenkova (2008) is to use rewrites of entities in summaries to remove some of the redundancy of automatic summaries. Chambers and Jurafsky (2009) utilizes reference resolution in order to build a model of narrative schemas and their participants which has clear applications to summarization.

[3]Although Farsi is written right-to-left, the words here are ordered left-to-right for the benefit of the reader.

Chapter 3

Related Work and Tasks

In this chapter we provide a brief overview of existing approaches to the entity resolution task (Section 3.1) as well related tasks (Section 3.2). We only describe the related work and tasks which are most directly related to the work in this dissertation.

3.1 Related Work

There has been much prior work on statistical approaches to reference resolution (McCarthy, 1996; Soon et al., 2001; Cardie and Wagstaff, 1999; Ng and Cardie, 2002; Ng, 2005; Bengston and Roth, 2008). Most of this work has focused on detecting the pairwise relationship that holds between a mention and its nearest *antecedent*, a previous mention which shares the same underlying entity reference. An overview of this line of research is given in Section 3.1.1. Far less explored are *entity-centered* models which rather than modeling compatibility between pairs of mentions, instead models the entity which underlies all referring mentions (McCallum and Wellner, 2005; Luo et al., 2004; Culotta et al., 2007; Haghighi and Klein, 2007, 2009; Wick et al., 2009; Haghighi and Klein, 2010; Rahman and Ng, 2009). This line of research, to which the work in this dissertation belongs, is overviewed in Section 3.1.3. Much less explored are unsupervised or lightly-supervised approaches (Cardie and Wagstaff, 1999; Bhattacharya and Getoor, 2006; Haghighi and Klein, 2007; Poon and Domingos, 2008; Ng, 2008; Haghighi and Klein, 2009, 2010). An overview of this work can be found in Section 3.1.3.

Feature Type	Feature	Description
Lexical	SOON_STR	Do the strings for m_i and m_j (discarding determiners)?
Syntactic	PRONOUN_1	Is m_j a pronoun?
	PRONOUN_2	Is m_i a pronoun?
	DEFINITE_2	Does m_i start with a demonstrative determiner (`this`,`that`,`these`,`those`)?
	NUMBER	Do m_i and m_j agree in syntactic number?
	GENDER	Do m_i and m_j agree in syntactic gender?
	BOTH_PROPER	Are m_i and m_j both proper mentions?
	APPOSITIVE	Are m_i and m_j in an appositive relationship.
Semantic	WNCLASS	Do m_i and m_j share the same *WordNet* semantic class.
	ALIAS	Is m_i an *alias* for m_j?
Positional	SENTNUM	Number of sentences between m_i and m_j

Table 3.1. All features used by the Soon et al. (2001) systems. All features are between a mention m_i and a potential antecedent m_j. Except for SENTNUM and WNCLASS, all features are binary.

3.1.1 Pairwise Approaches

The earliest statistical work in entity resolution was dominated by *pairwise* approaches. In this approach, a classifier is learned in a supervised manner to determine whether a given mention is coreferent with a particular antecedent. Concretely, suppose a document consists of mentions $m_1, \ldots, m_n$. We use $a_i \in \{1, \ldots, i\}$ to represent the anaphor decision for mention m_i. When $a_i < i$, this means that m_{a_i} is the selected anaphor, and $a_i = i$ denotes the event that the mention has no anaphor (i.e., it is the start of a new entity). Typically, pairwise systems (see below for exceptions) make independent anaphor decision and then assign a cluster id, c_i, to a mention m_i by following anaphor decisions backwards:

$$c_i = \begin{cases} i, & \text{if } a_i = i \\ c_{a_i}, & \text{otherwise} \end{cases} \tag{3.1}$$

We associate a cluster with the set of mentions assigned the same cluster id. This yields a proposed mention clustering.

An early important pairwise system, RESOLVE, described in McCarthy (1996), uses decision trees to determine anaphoricity and a set of features that are still commonly used by approaches to the present. RESOLVE was designed to be used within an information extraction system; because of this, many of its features are specific to the particular information extraction domain that was McCarthy (1996)'s final task. In addition to features to ensure the compatibility of linguistic phi-features (such as number, gender, and animacy), McCarthy (1996) also discovered that simple token-, sentence- and paragraph-distance features are useful for good performance. Soon et al. (2001) proposed a similar domain-independent system which utilized a richer set of features including semantic class features, which yielded

significant benefits (see Table 3.1 for a full listing). This system has been reproduced and used as a baseline in many other works (Ng and Cardie, 2002; Luo et al., 2004; Ng, 2005; Finkel and Manning, 2008). Another important contribution made by Soon et al. (2001) is how the anaphor decision is cast as a machine learning problem. A positive training example consists of each mention and the nearest coreferent antecedent. However, it is less clear as to what should constitute a negative training example. McCarthy (1996) took all pairs of non-coreferent mentions as negative training examples. A disadvantage of this is that the number of negative pairs far outweigh the positive and this can negatively affect learning. Soon et al. (2001) instead only extract negative training pairs for mentions which are sufficiently close to the target mention; specifically, all mentions closer to the target mention than to the nearest true antecedent.

Ng and Cardie (2002) presented a pairwise system which gave strong improvements over Soon et al. (2001)'s system. Most of these improvements came from improved NP detection (using shallow parsing), features for more linguistic constraints (such as binding theory), as well as the syntactic positions of the mention and the candidate antecedent. This latter feature encoded an important tendency described in *centering-theory* (Grosz et al., 1995), where in a coherent discourse, the subject of a sentence tends to be coreferent with the subject of the previous sentence.

More recently, Bengston and Roth (2008), presented a pairwise system which added several features; specifically, *WordNet* (Fellbaum, 1998), named entity resolution, and modifier alignment features.[1] Perhaps Bengston and Roth (2008)'s most important contribution was in breaking down the anaphor decision, a_i, by first deciding whether a mention has an antecedent or not (i.e, whether $a_i < i$). First a binary classifier decides whether a mention has an anaphor. Then a separate antecedent classifier selects amongst the available antecedents.

Many pairwise approaches have noted (Soon et al., 2001; Ng and Cardie, 2002) that errors have resulted from the lack of global coherency in the entities. As a particular example of this: both a male pronoun (`he`) and a female pronoun (`she`) may select the same proper mention antecedent.

There have been several approaches to maintain the basic pairwise approach but attempt to ensure more coherent global structure. McCallum and Wellner (2005) define a graph on document mentions where edge weights represent pairwise scores; then McCallum and Wellner (2005) use graph partitioning in order to reconcile pairwise scores into a final coherent clustering. Ng (2005) instead use a pairwise classifier to generate candidate partitions, but then re-rank those partitions according to a cluster-level model. Denis and Baldridge (2007) and Finkel and Manning (2008) use linear programming to ensure at inference time that basic compatibility holds globally amongst entity mentions. Crucially, these constraints are only applied at inference and do not affect learning. All these systems crucially rely on pairwise models to learn mention compatibility.

[1] The versions of *WordNet* used by Soon et al. (2001) and Bengston and Roth (2008) differ significantly.

How this work relates to pairwise approaches

The system proposed in Chapter 5 and the discourse component of the Chapter 6 model (see Section 6.3.2) are pairwise models, largely similar to those discussed here. The Chapter 5 system differs from the approaches described here in that it is unlearned and deterministic. The head-matching and agreement features (Section 5.3.1) used in this system are also used by the earliest pairwise systems (specifically Soon et al. (2001) and Ng and Cardie (2002)). One key difference here is that we achieve strong performance with these features without any annotated data to tune feature weights. The experiments which show that syntactic tree distance outperforms raw token distance as a proxy for discourse salience (Section 5.3.1) are novel. Syntactic tree distance is implicitly used by systems such as the one described in Hobbs (1977). The novelty in the Chapter 5 system is the mining of semantic information specifically for the task of reference resolution. The actual technique used for mining is broadly similar to other approaches (see Hearst (1992) for instance), but differ in the amount of data used and the targeted end application.

The discourse module presented in Section 6.3.2 is an unsupervised pairwise model which uses many of the same features as other pairwise approaches discussed here. Crucially, since this module is meant to capture only the discourse configurational structure of reference resolution, no lexical or semantic compatibility features are used since those are handled by the semantic entity-centered module (see Section 6.3.1).

3.1.2 Entity-Based Approaches

While far less explored than pairwise approaches, entity-based models have enjoyed increasing recent success. We consider a system entity-based if the primary decision made by the system is whether a mention is compatible with all entity mentions as opposed to a particular antecedent mention.

Luo et al. (2004) proposed a model where entity clustering is cast as a search problem in a bell-tree representation of mention partitions.[2] In this approach, a classifier goes from left-to-right across mentions and assigns each mention to an existing entity or creates a new entity. The binary probability of mention m joining entity e is modeled as a binary decision, where event $L = 0$ denotes it is not a member and $L = 1$ denotes that it is. At any given hypothesis, there are existing entities $e_1, \ldots, e_k$. Luo et al. (2004) consider the entities in order of which is most recently used. For each entity in this list, the model decides whether or not to link the current mention to that entity, with probability $P(L = 1|m, e)$. If a link is made, we move on to incorporate the next mentions. Otherwise, the model considers the next entity. If the mention isn't linked to any entity, it starts its own cluster.The form of the probability $P(L = 1|m, e)$ is log-linear and uses arbitrary features between a mention and an entity. Most of the features used by Luo et al. (2004) are variants of the mention-pairwise

[2]A bell number, $B(n)$, represents the number of ways to cluster n objects; it is given by $\frac{1}{e}\sum_{k=0}^{\infty}\frac{k^n}{k!}$. A bell-tree is a representation of all possible clusterings over n items, where each leaf of the tree corresponds to a possible clustering hypothesis.

features used in Ng and Cardie (2002) except that the for compatibility features, the feature is active only if the mention is compatible with all current entity mentions. For instance, the variant of the NUMBER feature (from Table 3.1) used here ensures that all mentions in the entity share the same number feature (or at least there are no conflicting values). Inference in this model is difficult because of the complexity of the hypothesis space (all bell-trees over n mentions).

Haghighi and Klein (2007) proposed the first generative clustering model for the full NP entity reference resolution problem.[3] This model is described fully in Chapter 4. Ng (2008) proposed several extensions to the approach of Haghighi and Klein (2007) which yielded significant performance gains. In particular, they utilized the salience components of the model with only pronominal mentions. Also, in oder to counter-act the bias of popular entities, Ng (2008) sampled entities for a given mention which which explicitly agreed in number, gender, and type with the current mention.

Poon and Domingos (2008) propose an unsupervised Markov Logic (Richardson and Domingos, 2006) approach to entity-based reference resolution which shares many similarities with Haghighi and Klein (2007). Most of the features used by the model are pairwise features (such as head-matching as well as appositive features). Importantly, Poon and Domingos (2008) utilize smarter head-detection and use of syntactic constraints relative to Haghighi and Klein (2007). These declarative constraints can be directly incorporated into the Markov Logic framework.

Rahman and Ng (2009) propose an approach similar to Ng (2005), where a base pairwise model generates a list of entity partition candidates. In contrast to Ng (2005), there are a richer set of entity-based features used to rank the entity partitions. These features are largely variants of common pairwise compatibility feature, but only hold when all the mentions of an entity are compatible. See Chapter 6 where we more directly compare to this system.

How this work relates to entity-based approaches

The work in this dissertation is distinct from other entity-based work in that it is unsupervised and generative. This combination facilitates the use of large amounts of unlabeled data which yields performance gains since semantic information can be learned in tandem with reference resolution structure (see Chapter 6). In particular, it is the only model capable of performing within- and cross-document reference resolution. Much of the recent unsupervised entity reference resolution work (Ng, 2008; Poon and Domingos, 2008) has been influenced by the work presented in Chapter 4.

[3]Milch et al. (2005),Daume and Marcu (2005) and Bhattacharya and Getoor (2006) proposed generative clustering models for the related, but not identical, tasks of proper NP resolution and citation resolution. See Section 3.2.2.

3.1.3 Unsupervised Approaches

While less common than supervised approaches, there has been a growth of recent interest in unsupervised entity reference resolution. While much of the earliest work (Hobbs, 1977, 1979) are technically unsupervised, in fact they are not learned, they do not present quantitative evaluation. Cardie and Wagstaff (1999) present the earliest evaluated unsupervised system. Specifically, they present a clustering model in which distances between mentions are given by parametrized incompatibility functions; the feature of these distances are similar to those used by pairwise approaches (see Section 3.1.1). These pairwise distances are used as input to a pairwise clustering algorithms. The Cardie and Wagstaff (1999) approach does several parameters to maximize labeled data performance. It achieved impressive performance relative to existing fully supervised approaches, including McCarthy (1996), which is fully supervised.

Haghighi and Klein (2007) (see Chapter 4 for a full model description) present an unsupervised Bayesian non-parametric model in which mentions are generated from underlying latent entity representations. This model also allows for cross-document entity detection since the underlying model structure allows for entity sharing amongst documents. Ng (2008) present both extensions to Haghighi and Klein (2007) (see Section 3.1.2) as well propose a novel unsupervised pairwise model, which utilizes many of the features in Ng and Cardie (2002). This pairwise model, while unsupervised, is not strictly generative and does not ensure global consistency amongst entity mentions. Poon and Domingos (2008)'s Markov Logic model is also unsupervised and incorporates a rich set of linguistic constraints.

How this work relates to other unsupervised approaches

Chapter 4 presents the first unsupervised probabilistic generative model of entity reference resolution. Importantly, it is the only model which can seamlessly perform within- and cross-document coreference resolution. In Chapter 6, we present a model which also is unique in learning and using soft semantic constraints.

3.2 Related Tasks

The abstract problem of deciding when two references reference the same object has many instantiations ranging from natural language processing to databases. There are many variants of the entity reference resolution tasks as well as related task which utilize similar or identical terminology. We describe the related tasks most relevant to this work.

3.2.1 Pronoun Anaphora Resolution

As mentioned in Section 5.1, anaphora resolution is the task of identifying an antecedent mention to which a referential pronoun refers. The decisions made by a pronoun anaphora system are identical to the anaphor decision described in Section 3.1.1, except the decisions are limited to pronoun mentions and we do not produce or evaluate clusters.

Some of the earliest work in reference resolution focused on pronoun anaphora resolution (Hobbs, 1977; Lappin and Leass, 1994; Ge et al., 1998). Strictly speaking, a sub-task, of the full entity reference resolution task, since antecedents are determined by full reference resolution. Nonetheless, much of the difficulty of full reference resolution reduces to identifying a coreferring antecedents for each pronoun. One benefit of this task is that it admits simpler machine learning models; each pronoun decides upon one of a fixed number of potential antecedents. Although see Denis (2007) for a ranking, rather than classification, approach. Charniak and Elsner (2009) presents a simple, but highly-effective, EM algorithm for anaphora identification. In contrast, full reference resolution must contend with combinatorial structure of all possible mention clusterings. However, many of the insights made on this task, particularly Charniak and Elsner (2009), are incorporated into the models described here.

3.2.2 Citation Deduplication and Record Linkage

Another thread of entity reference resolution work has focused on the problem of identifying proper name matches between documents (Milch et al., 2005; Bhattacharya and Getoor, 2006; Daume and Marcu, 2005). Much of this work focuses on domains such as the problem of citation deduplication from papers (Pasula et al., 2003). In this task, the goal is to resolve citations from academic papers to an underlying database of citations. The complexity in this task arises from possible variations of author and paper names (e.g., dropping a middle initial from an author name or determiner from a paper title) as well as possible spelling errors. Similar problems arise in the management of large databases, where a given record for an individual may be duplicated with minor surface name variations (Fellegi and Sunter, 1969). Since the mentions considered in this thread of research are almost exclusively proper mentions, most of the effort in this task is devoted to having rich surface string similarity features.

This family of tasks shares a lot with the task considered in this dissertation: They are both clustering problems, where the number of clusters is unknown. In fact, similar machine learning frameworks as ours have been used to tackle these problems. Milch et al. (2005) proposed BLOG, a specification for generic generative models in which inference can be automatically carried out. This framework is also applied to citation de-duplication with success. Daume and Marcu (2005) proposes a supervised clustering model based on the Dirichlet Process prior. These models, as ours, are generative ones, since the focus is on cluster discovery and the data is generally unlabeled.

One key difference between this task and the task considered in this dissertation is the

context in which mentions appear. In entity reference resolution as considered in this dissertation, mentions are situated within a linguistic discourse (e.g., a document); in this context, nominal and pronominal references are used in addition to full proper names. One significant source of difficulty in our entity reference resolution problem comes from the simultaneous resolution of pronouns and nominals as well as proper mentions. The models developed in subsequent chapters are aimed at modeling the linear discourse structure of a document in order to aid reference resolution. Much of what makes our task difficult is not present in these deduplication problems.

On the other hand, much of the difficulty of the deduplication tasks arises from the lack of consistent surface canonical names; this is not typically present in our task, where names are, for instance, typically spelled consistently in a given document. So while the machine learning frameworks between these two areas can be shared, much of the phenomena that is being modeled differ significantly.

Chapter 4

Generative Models of Entity Resolution

4.1 Introduction

Broadly speaking, the process of evoking an entity in natural language can be decomposed into two processes. First, speakers directly introduce new entities into discourse, entities which may be shared across discourses. This initial reference is typically accomplished with proper or nominal expressions. Second, speakers refer back to entities already introduced. This anaphoric reference is canonically, though of course not always, accomplished with pronouns, and is governed by linguistic and cognitive processes. In this chapter, we present a bayesian nonparametric generative model of a document corpus which naturally connects these two processes, sharing a global set of entities across documents, and then modeling reference chains inside documents. This chapter is broadly concerned with the machine learning framework that will facilitate learning entity reference structure, rather than incorporating known linguistic constraints (this is done in Chapter 5).

Most recent entity reference resolution work has focused on the task of deciding which *mentions* (noun phrases) in a document are coreferent. The dominant approach, described more fully in Section 3.1.1, is to decompose the task into a collection of pairwise coreference decisions. One then applies discriminative learning methods to pairs of mentions, using features which encode properties such as distance, syntactic environment, and so on (Soon et al., 2001; Ng and Cardie, 2002). Although such approaches have been successful, they have several liabilities. First, rich features require bountiful labeled data, which we do not have for entity resolution tasks in most domains and languages. Second, entity reference resolution is inherently a clustering or partitioning task where each cluster contains the

mention references to a single entity. Naive pairwise methods can and do fail to produce coherent partitions, for example by making non-transitive decisions. One classic method of addressing this issue is to take the transitive closures which can result in globally incoherent entities. In Section 3.1.1, we described recent work which have attempted more sophisticated techniques to reconcile a pairwise model with ensuring a reasonable global hypothesis. While these attempts have enjoyed success, it might be more beneficial to explore models which directly model entity partition structure.

The model presented in this chapter naturally captures both within- and cross-document entity resolution. At the top, a hierarchical Dirichlet process (Teh et al., 2006) captures cross-document entity (and parameter) sharing, while, at the bottom, a sequential model of salience captures within-document sequential structure.[1] As a joint model of several kinds of discourse variables, it can be used to make predictions about either kind of coreference, though we focus experimentally on the within-document measures. The model presented in this chapter, at the time of the publication of Haghighi and Klein (2007), achieved the best entity reference resolution performance for an unsupervised system on the standard MUC-6 test set.

4.2 Experimental Setup

The experimental setup used for the systems in each chapter will differ in a small number of ways. We briefly describe the experimental methodology used in this chapter. The experiments in this chapter use the conventions of the the Automatic Context Extraction (ACE) task (NIST, 2004).

Recall that we assume document in a corpus consists of a set of *mentions*, typically noun phrases (NPs). Each mention is a *reference* to some entity in the domain of discourse. The entity reference resolution task is to partition the mentions according to referent. Mentions can be divided into three basic types, *proper* mentions (names), *nominal* mentions (descriptions), and *pronominal* mentions (pronouns).

In section 4.3, we present a sequence of increasingly enriched models, motivating each from shortcomings of the previous. As we go, we will mention performance of each model on data from ACE 2004 (NIST, 2004). In particular, we will use as our development corpus the English translations of the Arabic and Chinese treebanks, comprising 95 documents and about 3905 mentions. This data was used heavily in development for model design and hyper-parameter selection. In section 4.5, we present results for new test data from MUC-6 on which no tuning or development was performed. This test data will form our basis for comparison to previous work.

For the experiments in this chapter, we only consider the *gold mention setting*, described fully in Section 2.1, where annotated mention spans are provided to the system at training and test time. Furthermore, we assume that the head word of each mention and the mention

[1] See Appendix A for a review of the Dirichlet process and its extensions.

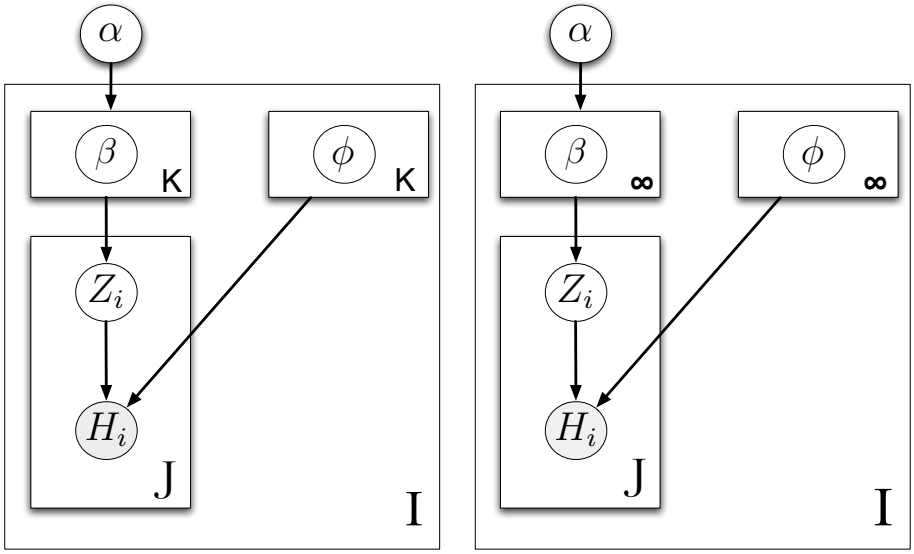


Figure 4.1. Graphical model depiction of document level entity models described in sections 4.3.1 and 4.3.2 respectively. The shaded nodes indicate observed variables. The ∞ in the plate diagram denotes there is no finite bound on the possible number of components.

type (proper, nominal, or pronominal) are also provided to the system. This experimental condition allows us to focus on building and evaluating a model of entity reference resolution rather than mention detection or NP parsing.

For the ACE data sets, the head and mention type are given as part of the mention annotation. For the MUC data, the head was crudely chosen to be the rightmost mention token and the mention type was automatically detected. As Poon and Domingos (2008) demonstrate, this crude heuristic can certainly be improved upon. We will not assume any other information to be present in the data beyond the text itself. In particular, unlike much related work, we do not assume gold named entity recognition (NER) labels, lexical resources such as *WordNet* (Fellbaum, 1998), or geographical or geopolitical information found in gazetteers. Indeed we do not assume observed NER labels or POS tags at all. The focus of this chapter is to develop the machine learning framework for modeling entity partition structure rather than focusing on the available resources which are known to benefit entity reference resolution.

Our primary performance metric will be the MUC F-measure (Vilain et al., 1995), commonly used to evaluate coreference systems on a within-document basis; see Section 2.2.1 for a fuller description. Since our system relies on sampling, all results are averaged over five random runs.

(a) [The Weir Group$_1$, whose$_2$] headquarters$_3$ is in the US$_4$, is a large, specialized corporation$_5$ investing in the area of electricity generation. This [power plant$_6$, which$_7$] will be situated in Rudong$_8$, Jiangsu$_9$, has an annual generation capacity of 2.4 million kilowatts.

(b) [The Weir Group$_1$, whose$_1$] headquarters$_2$ is in the US$_3$, is a large, specialized corporation$_4$ investing in the area of electricity generation. This [power plant$_5$, which$_1$] will be situated in Rudong$_6$, Jiangsu$_7$, has an annual generation capacity of 2.4 million kilowatts.

(c) [The Weir Group$_1$, whose$_1$] headquarters$_2$ is in the US$_3$, is a large, specialized corporation$_4$ investing in the area of electricity generation. This [power plant$_5$, which$_5$] will be situated in Rudong$_6$, Jiangsu$_7$, has an annual generation capacity of 2.4 million kilowatts.

Figure 4.2. Example output from various models. The output from (a) is from the infinite mixture model of section 4.3.2. It incorrectly labels both boxed cases of anaphora. The output from (b) uses the pronoun head model of section 4.3.3. It correctly labels the first case of anaphora but incorrectly labels the second pronominal as being coreferent with the dominant document entity *The Weir Group*. This error is fixed by adding the salience feature component from section 4.3.4 as can be seen in (c).

4.3 Coreference Resolution Models

In this section, we present a sequence of generative coreference resolution models for document corpora. Each model will be motivated from errors made from the last model. All are statistical mixture models, where the mixture components correspond to entities. As far as notation, we assume a collection of I documents. We use the random variable Z to refer to the index of an entity. For a document with n mentions, we use $\mathbf{Z} = (Z_1, \ldots, Z_n)$ to refer to the entity index assignments to the mentions of a document (i.e., Z_i is the entity index of the ith mention).

We will use ϕ_z to denote the parameters for an entity z, and $\boldsymbol{\phi}$ to refer to the concatenation of all such ϕ_z. M will refer somewhat loosely to the collection of variables associated with a mention in our model (such as the head or gender). We will be explicit about the representation of M and ϕ_z shortly.

Our goal will be to find the setting of $\mathbf{Z}$ entity indices which maximize the posterior probability:

$$\begin{aligned} \mathbf{Z}^* &= \arg\max_{\mathbf{Z}} P(\mathbf{Z}|\mathbf{M}) = \arg\max_{\mathbf{Z}} P(\mathbf{Z}, \mathbf{M}) \\ &= \arg\max_{\mathbf{Z}} \int P(\mathbf{Z}, \mathbf{M}, \boldsymbol{\phi})\, dP(\boldsymbol{\phi}) \end{aligned}$$

where $\mathbf{Z}, \mathbf{M}$, and $\boldsymbol{\phi}$ denote all the entity indices, observed values, and parameters of the model. Note that we take a Bayesian approach in which all parameters are integrated out

(or sampled). The inference task is thus primarily a search problem over the entity index assignments $\mathbf{Z}$.

4.3.1 A Finite Mixture Model

Our first, overly simplistic, corpus model is the standard finite mixture of multinomials shown in Figure 4.1(a). In this model, each document is independent save for some global hyperparameters. Inside each document, there is a finite mixture model with a fixed number K of components. The distribution β is drawn from a symmetric Dirichlet distribution with concentration α. For each mention in the document, we choose a component (an entity index) z from β. Entity z is then associated with a multinomial emission distribution over head words with parameters ϕ^h_Z, which are drawn from a symmetric Dirichlet over possible mention heads with concentration λ_H.[2] We use V to denote the size of the mention head vocabulary. This process can be summarized as:

For each document D,

  Draw $\beta \sim \textsc{Dirichlet}(\alpha, K)$

  For each entity $k = 1, \ldots, K$

    $\phi^h_k \sim \textsc{Dirichlet}(\lambda_H, V)$

  For each mention head H_i, $i = 1, \ldots, n$

    Draw $Z_i \sim \textsc{Multinomial}(\beta)$

    Draw $H_i \sim \textsc{Multinomial}(\phi^h_{Z_i})$

As we describe our models, we simultaneously develop the accompanying Gibbs sampling procedure to obtain samples from $P(\mathbf{Z}|\mathbf{M})$. In principle, one could use the EM algorithm to perform clustering in this model, but EM will not extend effectively to subsequent models. For now, all heads H are observed and all parameters (β and $\boldsymbol{\phi}$) can be integrated out analytically: for details see Teh et al. (2006). The only sampling is for the values of $Z_{i,j}$, the entity index of mention j in document i. Using standard notation from the Markov chain Monte Carlo (MCMC) literature (), we use $\mathbf{Z}^{-i,j}$ to denote $\mathbf{Z} - \{Z_{i,j}\}$, all the entity indicator variables except for the one being sampled. The relevant conditional distribution is:

$$P(Z_{i,j}|\mathbf{Z}^{-i,j}, \mathbf{H}) \propto P(Z_{i,j}|\mathbf{Z}^{-i,j})P(H_{i,j}|\mathbf{Z}, \mathbf{H}^{-i,j}) \tag{4.1}$$

where $H_{i,j}$ is the head of mention j in document i. Expanding each term, we have the contribution of the prior:

$$P(Z_{i,j} = z|\mathbf{Z}^{-i,j}) \propto n_z + \alpha \tag{4.2}$$

[2] In general, we will use a subscripted λ to indicate concentration for finite Dirichlet distributions. Unless otherwise specified, λ concentration parameters will be set to e^{-4} and omitted from diagrams.

where n_z is the number of elements of $\mathbf{Z}^{-i,j}$ with entity index z. Similarly we have for the contribution of the emissions:

$$P(H_{i,j} = h|\mathbf{Z}, \mathbf{H}^{-i,j}) \propto n_{h,z} + \lambda_H \tag{4.3}$$

where $n_{h,z}$ is the number of times we have seen head h associated with entity index z in $(\mathbf{Z}, \mathbf{H}^{-i,j})$.

It is worth noting that these sampling equations match those for the latent dirichlet allocation model (LDA) when using collapsed Gibbs sampling described in Griffiths and Steyvers (2004).

4.3.2 Infinite Mixture Model

A clear drawback of the finite mixture model is the requirement that we specify a priori a number of entities K for a document. In general, the number of entities in a document K is not a known constant, but a random variable. We would like our model to select K in an effective, principled way. A mechanism for doing so is to replace the finite Dirichlet prior on β with the non-parametric Dirichlet process (DP) prior (Ferguson, 1973). The Dirichlet process is a nonparametric Bayesian device which can be used for exactly this purpose; a review of some of the basic concepts behind the DP and associated techniques can be found in Appendix A. In particular, in Section A.1.2, we present more of the background for the infinite mixture model.

Drawing our entity distribution β from a DP with concentration parameter α and using the stick-breaking representation, we obtain the model in Figure 4.1(b). Doing so gives the model in Figure 4.1(b). Note that we now list an infinite number of mixture components in this model since there can be an unbounded number of entities. Rather than a finite β with a symmetric Dirichlet distribution, in which draws tend to have balanced clusters, we now have an infinite β. However, most draws will have weights which decay exponentially quickly in the prior (though not necessarily in the posterior). Therefore, there is a natural penalty for each cluster which is actually used.

With $\mathbf{Z}$ observed during sampling, we can integrate out β and calculate $P(Z_{i,j}|\mathbf{Z}^{-i,j})$ analytically, using the Chinese restaurant process representation (Aldous, 1985):

$$P(Z_{i,j} = z|\mathbf{Z}^{-i,j}) \propto \begin{cases} \alpha, & \text{if } z = z_{new} \\ n_z, & \text{otherwise} \end{cases} \tag{4.4}$$

where z_{new} is a new entity index not used in $\mathbf{Z}^{-i,j}$ and n_z is the number of mentions that have entity index z. Aside from this change, sampling is identical to the finite mixture case, though with the number of clusters actually occupied in each sample drifting upwards or downwards. The head emission model is unchanged: heads are emitted using a per-cluster head distribution.

A particular advantage of the DP prior is that it prefers a few large clusters and several smaller "one-off" ones. This roughly reflects the property of newswire text that most

mentions reference a small set of entities which are the main subject of the article. Another advantage (discussed in section 4.3.5) is that we can share entities hierarchically among documents.

This model yielded a 54.5 MUC-F_1 on our development data.[3] This model is, however, hopelessly crude, capturing nothing of the structure of entity resolution. Its largest empirical problem is that, unsurprisingly, pronoun mentions such as `he` are given their own cluster, not labeled as coreferent with any non-pronominal mention (see Figure 4.2(a)).

We can isolate the performance of our system on resolving pronouns by measuring the Pairwise F_1 (see Section 2.2.1) on only pronoun to non-pronoun coreference events. On this evaluation, our system yields 26.4 , confirming the observation that the model is doing a poor job of recovering patterns of pronoun resolution.

4.3.3 Pronoun Head Model

While an entity-specific multinomial distribution over heads makes sense for proper and some nominal mention heads, it does not make sense to generate pronominal mentions this same way. I.e., all entities can be referred to by generic pronouns, the choice of which depends on properties such as gender, not the specific entity. We use this observation to construct an enriched emission model in which pronoun emissions depend upon entity properties.

We enrich an entity's parameters ϕ to contain not only a distribution over lexical heads ϕ^h, but also distributions (ϕ^t, ϕ^g, ϕ^n) over properties, where ϕ^t parametrizes a distribution over entity types (PER, LOC, ORG, MISC), and ϕ^g for gender (MALE, FEMALE, NEUTER), and ϕ^n for number (SG, PL).[4] We assume each of these property distributions is drawn from a symmetric Dirichlet distribution with small concentration parameter in order to encourage a peaked posterior distribution.

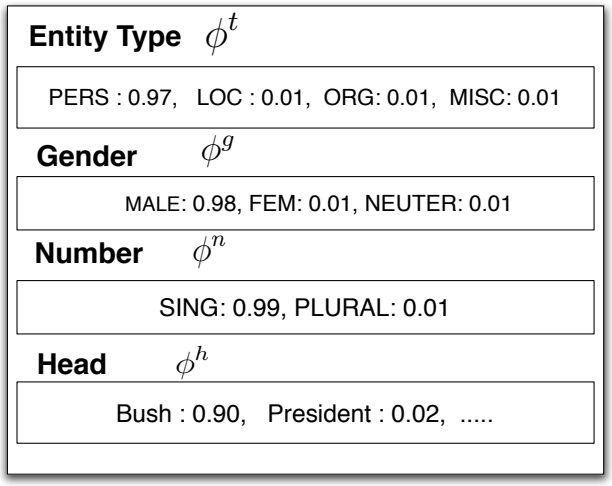
Previously, when an entity z generated a mention, it used to draw a head word from ϕ^h_z. It now undergoes a more complex and structured process. It first uses its property distributions to draw an entity type T, a gender G, a number N from the distributions ϕ^t, ϕ^g, and ϕ^n, respectively.

Once the properties are fetched, a mention type P is chosen (*proper*, *nominal*, *pronoun*), according to a global multinomial (again with a symmetric Dirichlet prior and parameter λ_P). This corresponds to the (temporary) assumption that the speaker makes a random i.i.d. choice at each mention as to the mention type.

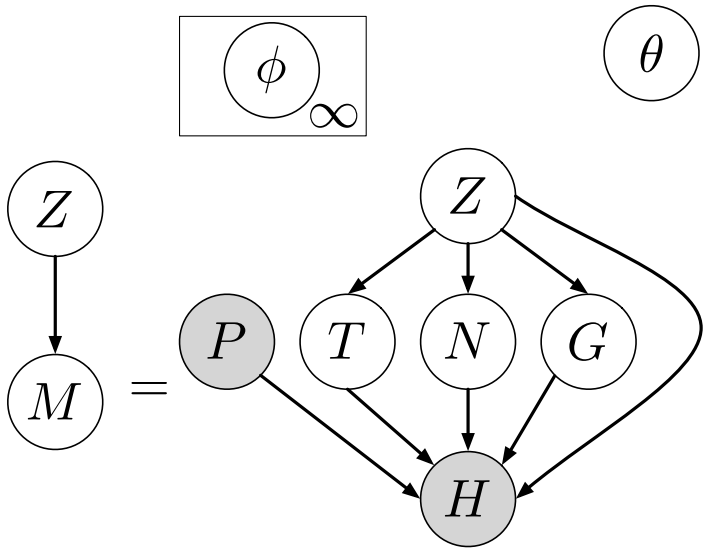
Our head model will then generate a head conditioning on the entity, its properties, and the mention type, as shown in Figure 4.3(b). If P is not a pronoun, the head is drawn directly

[3] See section 7.3 for inference details.

[4] It might seem that entities should simply have, for example, a gender g rather than a distribution over genders ϕ^g. There are two reasons to adopt the softer approach. First, one can rationalize it in principle, for entities like cars or ships whose grammatical gender is not deterministic. However, the real reason is that inference is simplified. In any event, we found these property distributions to be highly determinized in the posterior.



(a)



(b)

Figure 4.3. In (a), we have a depiction of an entity and its parameters. In (b), we have a graphical model depiction of the head model described in section 4.3.3. For a pronoun mention, the head depends only on the entity type (T), number (N), and gender (G) draws. For non-pronominal head, our head is drawn from an entity specific head multinomial in ϕ_z The shaded nodes indicate observed variables. The mention type determines which set of parents are used. The dependence of mention variable on entity parameters $\boldsymbol{\phi}$ and pronoun head model $\boldsymbol{\theta}$ are omitted.

from the entity head multinomial with parameters ϕ_z^h. Otherwise, it is drawn based on a global pronoun head distribution conditioning on the entity properties and parametrized by $\boldsymbol{\theta}$. Formally, it is given by:

$$P(H|Z,T,G,N,P,\boldsymbol{\phi},\boldsymbol{\theta}) = \begin{cases} P(H|T,G,N,\boldsymbol{\theta}), & \text{if } P = \text{PRO} \\ P(H|\phi_Z^h), & \text{otherwise} \end{cases}$$

If all T, G, and N variables were observed, it would be straightforward to obtain the posterior probability of $P(H|\mathbf{Z},\mathbf{T},\mathbf{G},\mathbf{H},\mathbf{N},)$. Although there are many pronoun for which we know the value of the entity type, gender, and/or number, there are many for which this value is unobserved (i.e. `who` or `its`).

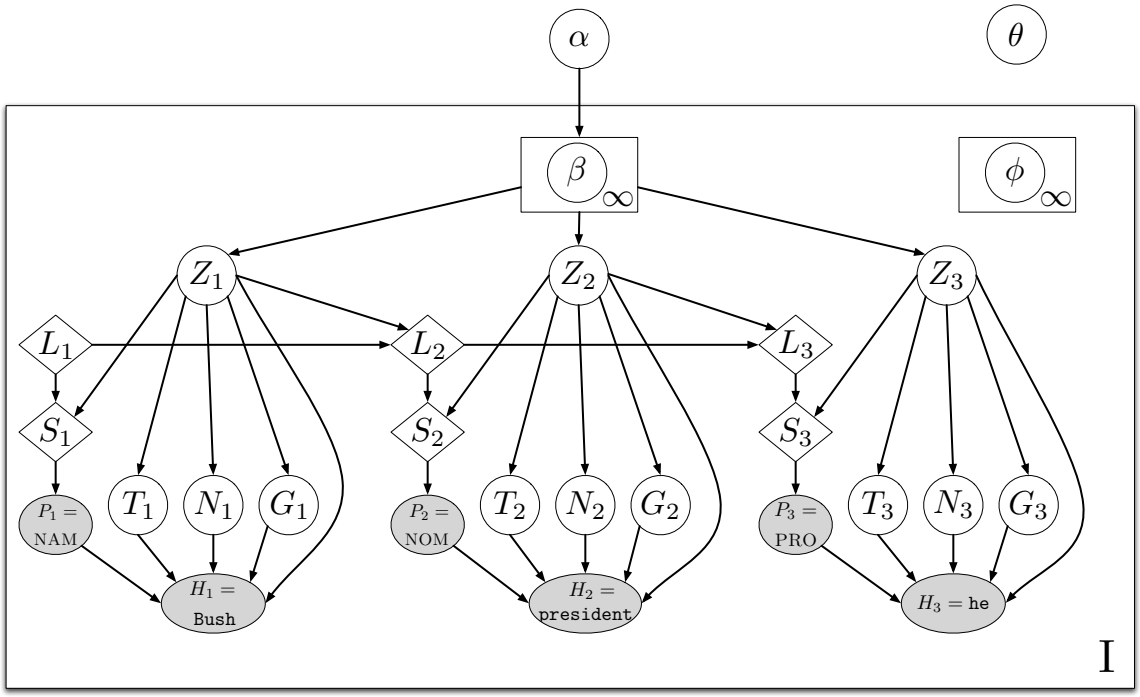


Figure 4.4. Coreference model at the document level with entity properties as well the salience lists used for mention type distributions. The diamond nodes indicate deterministic functions. Shaded nodes indicate observed variables. Although it appears that each mention head node has many parents, for a given mention type, the mention head depends on only a small subset. Note that dependencies involving parameters $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ are omitted for clarity. For instance, any non-pronominal mention head depends only on the entity parent. Diamond-shaped nodes denote variables which are deterministic given their parents.

Because the entity property draws are not (all) observed, we must now sample the unobserved ones as well as the entity indicator indices Z. For instance, we could sample $T_{i,j}$, the entity type of pronominal mention j in document i, using,

$$P(T_{i,j}|\mathbf{Z},\mathbf{N},\mathbf{G},\mathbf{H},\mathbf{T}^{-i,j}) \propto P(T_{i,j}|\mathbf{Z})P(H_{i,j}|\mathbf{T},\mathbf{N},\mathbf{G},\mathbf{H}) \tag{4.5}$$

Salience Feature	Pronoun	Proper	Nominal
Top	0.75	0.17	0.08
High	0.55	0.28	0.17
Mid	0.39	0.40	0.21
Low	0.20	0.45	0.35
None	0.00	0.88	0.12

Table 4.1. Posterior distribution of mention type given salience feature s, chosen by bucketing entity activation rank. Each row depicts the probability distribution over mention types given the salience feature of the entity. This distribution reflects the intuition that pronouns are preferred for entities which have high salience and non-pronominal mentions are preferred for inactive entities.

where the posterior distributions on the right hand side are straightforward to compute since the parameter priors are all finite Dirichlet. Sampling G and N are identical. We opt instead for a different approach during learning. We integrate out (T, G, N) for each mention and instead place variational estimates on (ϕ^t, ϕ^g, ϕ^n) (for each entity) and $\boldsymbol{\theta}$. See Section B.1 for details on precisely how this is done.

Of course we have some (prior) information about the relationship between entity type and pronoun head choice. For example, we expect that `he` is used for mentions with $T =$ *person*. In general, we assume that for each pronominal head we have a list of compatible entity types, which we encode via the prior on $\boldsymbol{\theta}$. We assume $\boldsymbol{\theta}$ is drawn from a Dirichlet distribution where each pronoun head is given a synthetic count of $(1 + \lambda_P)$ for each (t, g, n) where t is compatible with the pronoun and given λ_P otherwise. So while it will be possible in the posterior to use `he` to refer to a non-person, it will be biased towards being used with a person.

This model gives substantially improved predictions: 64.1 MUC-F_1 on our development data. As can be seen in Figure 4.2(b), this model does correct the systematic problem of pronouns being considered their own entities. However, it still does not have a preference for associating pronominal references to entities which are in any way local. Indeed, there is no preference to have a non-pronominal mention precede a pronominal mention.[5]

When using our pairwise pronoun evaluation, the mode yields 48.0 pairwise F_1 (see Section 2.2.1). This substantially outperforms the pronoun resolution of Section 4.3.2, indicating the pronoun head model described here captures pronoun resolution structure more accurately. Although this represents an improvement, there are still several problems with the model. Chief among these are that the model has no preference for generating pronominal references which are near their antecedents.

[5]Although pronominal mentions can be the first to an entity, this is generally not the case.

4.3.4 Adding Salience

We would like our model to capture how mention types are generated for a given entity in a robust and somewhat language independent way. The choice of entities may reasonably be considered to be independent given the mixing weights β, but how we *realize* an entity is strongly dependent on context (Ge et al., 1998). To a first approximation, we use proper and nominal mentions for entities which are not active in the discourse, and pronouns for entities which are (Ge et al., 1998).

Roughly, the salience of entity represents how active or recent it is in a listener's memory (Lappin and Leass, 1994). In this work, we will associate a real-number with each entity that represents its activity. The higher the activity, the more likely an entity is to be active in the memory of a listener. We choose to model the salience, or activity score, for all entities in a discourse.

In order to capture this in our model, we enrich it as shown in Figure 4.4. As we proceed through a document, generating entities and their mentions, we maintain a list of the active entities and their saliences, or activity scores.

Every time an entity is mentioned, we increment its activity score by 1, and every time we move to generate the next mention, all activity scores decay by a constant factor of 0.5. This gives rise to an ordered list of entity activations, L, where the rank of an entity decays exponentially as new mentions are generated. We call this list a *salience list*. Given the current salience list, L, we can associate a rank with each possible entity Z. We discretize these ranks into five buckets: Top (1), High (2-3), Mid (4-6), Low (7+), and None, depending on the absolute rank, if any, of the entity in the salience list. Given the entity choices Z, both the list and S are deterministic (see Figure 4.4). We assume that the mention type M is generated by the salience feature L from a multinomial distribution drawn from a symmetric Dirichlet distribution.

We note that correctly sampling an entity now requires that we incorporate terms for how a change will affect all future salience values. This changes our sampling equation for existing entities:

$$P(Z_{i,j} = z|\mathbf{Z}^{-i,j}) \propto n_z \prod_{j' \geq j} P(M_{i,j'}|S_{i,j'}, \mathbf{Z}) \tag{4.6}$$

where the product ranges over future mentions in the document and $S_{i,j'}$ is the value of future salience feature given the setting of all entities, including setting the current entity $Z_{i,j}$ to z. A similar equation holds for sampling a new entity. Note that, as discussed below, this full product can be truncated as an approximation.

This model gives a 71.5 MUC-F_1 on our development data. Table 4.1 shows the posterior distribution of the mention type given the salience feature. This model fixes many anaphora errors and in particular fixes the second anaphora error in Figure 4.2(c).

On the pairwise pronoun evaluation, the salience model yields 55.2 F_1, a strong improvement over the model from the previous section.

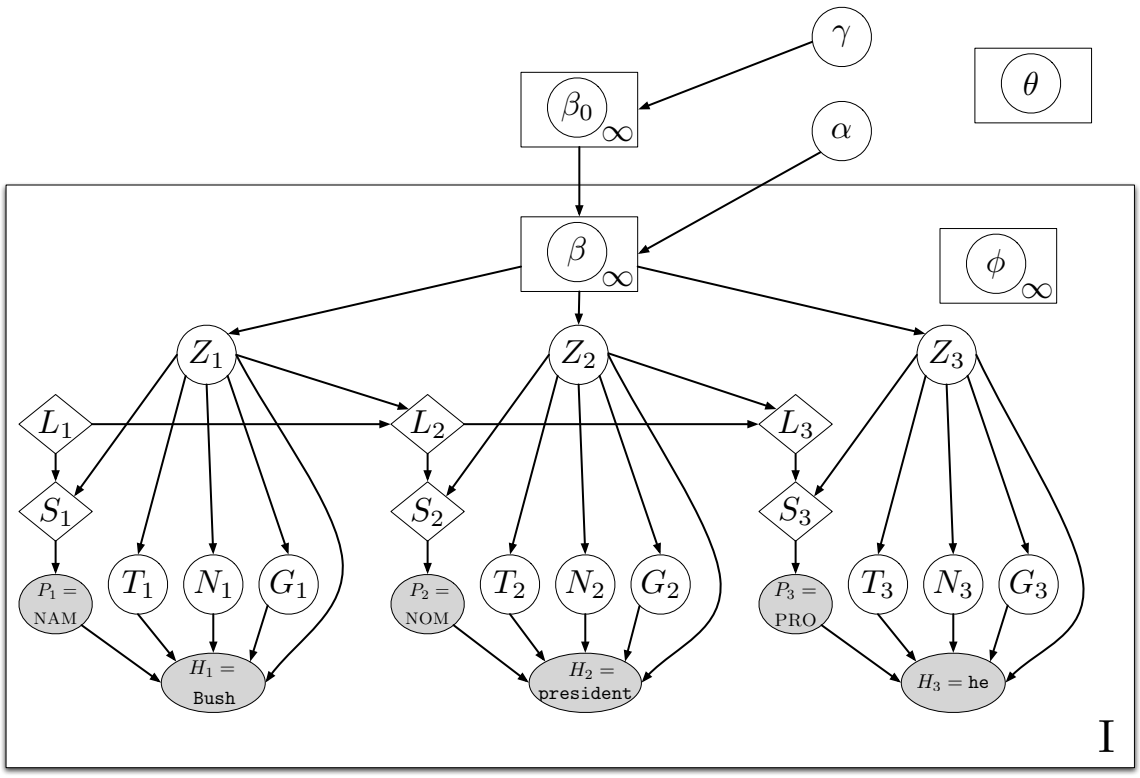


Figure 4.5. Graphical depiction of the hierarchical Dirichlet Process (HDP) reference resolution model described in section 4.3.5. The dependencies between the global entity parameters ϕ and pronoun head parameters $\boldsymbol{\theta}$ on the mention observations are not depicted.

4.3.5 Cross Document Coreference

One advantage to a fully generative approach is that we can allow entities to be shared between documents in a principled way, giving us the capacity to do cross-document entity resolution. Many potential downstream applications such as information extraction must reason about entities across multiple documents. Using an entity resolution system which naturally models the sharing of entities might ultimately benefit such downstream applications. Moreover, sharing across documents pools information about the properties of an entity across documents.[6]

We can easily link entities across a corpus by assuming that the pool of entities is global, with global mixing weights β_0 drawn from a DP prior with concentration parameter γ. Each document uses the same global entities, but each has a document-specific distribution β_i drawn from a DP centered on β_0 with concentration parameter α. Up to the point where entities are chosen, this formulation follows the basic hierarchical Dirichlet process prior of Teh et al. (2006) (see Appendix A.3 for a review). Once the entities are chosen, our model for the realization of the mentions is as before. This model is depicted graphically in Figure 4.5.

[6]The degree of cross-document coreference will vary from corpus to corpus.

Although it is possible to integrate out β_0 as we did the individual β_i, we instead choose for efficiency and simplicity to sample the global mixture distribution β_0 from the posterior distribution $P(\beta_0|\mathbf{Z})$.[7] The mention generation terms in the model and sampler are unchanged.

In the full hierarchical model, our equation (4.4) for sampling entities, ignoring the salience component of section 4.3.4, becomes:

$$P(Z_{i,j} = z|\mathbf{Z}^{-i,j}, \beta_0) \propto \begin{cases} \alpha\beta_0^u, & \text{if } z = z_{new} \\ n_z + \alpha\beta_0^z, & \text{otherwise} \end{cases} \tag{4.7}$$

where β_0^z is the probability of the entity z under the sampled global entity distribution and β_0^u is the unknown component mass of this distribution.

The HDP layer of sharing improves the model's predictions to 72.5 F_1 on our development data. We should emphasize that our evaluation is of course per-document and does not reflect cross-document coreference decisions, only the gains through cross-document sharing (see section 4.6.2). The pronoun pairwise F_1 figure raises to 57.5 F_1.

4.4 Inference Details

Up until now, we've discussed Gibbs sampling, but we are not interested in sampling from the posterior $P(\mathbf{Z}|\mathbf{X})$, but in finding its mode. Although we can use samples from the posterior to approximate the mode, it is not an ideal approach. Instead of sampling directly from the posterior distribution, we instead continually bias our entity samples towards entities with higher posterior probabilities. In particular, we sample entities proportional to exponentiated entity posteriors. The posterior exponent is given by $\exp \frac{ci}{k-1}$, where i is the current round number (starting at $i = 0$), $c = 1.5$ and $k = 20$ is the total number of sampling epochs. This slowly raises the posterior exponent from 1.0 to e^c. This procedure is similar to simulated annealing with the posteriors as a proposal distribution. In our experiments, we found this procedure to outperform simulated annealing.

We also found sampling the T, G, and N variables to be particularly inefficient, so instead we opted to integrate these variables out and instead use variational estimates for $\boldsymbol{\theta}$ and the type, gender, and number distributions for each entity. See Section B.1 for full details. We also found that correctly accounting for the future impact of salience changes was particularly inefficient. However, ignoring those terms entirely made negligible different in final accuracy.[8]

[7] We do not give the details here; see Teh et al. (2006) for details on how to implement this component of the sampler (called "direct assignment" in that reference).

[8] This corresponds to truncating the product in equation (4.6) at $j' = j$.

(a)

Dataset	Num Docs.	Prec.	Recall	F_1
MUC-6	60	80.8	52.8	63.9
+DRYRUN-TRAIN	251	79.1	59.7	68.0
+ENGLISH-NWIRE	381	80.4	62.4	70.3

(b)

Dataset	Prec.	Recall	F_1
ENGLISH-NWIRE	66.7	62.3	64.2
ENGLISH-BNEWS	63.2	61.3	62.3
CHINESE-NWIRE	71.6	63.3	67.2
CHINESE-BNEWS	71.2	61.8	66.2

Table 4.2. Formal Results: Our system evaluated using the MUC model theoretic measure Vilain et al. (1995). The table in (a) is our performance on the thirty document MUC-6 formal test set with increasing amounts of training data. In all cases for the table, we are evaluating on the same thirty document test set which is included in our training set, since our system in unsupervised. The table in (b) is our performance on the ACE 2004 training sets.

4.5 Experiments

We present our formal experiments using the full model developed in section 4.3. As in section 4.3, we use true mention boundaries and evaluate using the MUC F_1 measure (Vilain et al., 1995). All hyper-parameters were tuned on the development set only. The document concentration parameter α was set by taking a constant proportion of the average number of mentions in a document across the corpus. This number was chosen to minimize the squared error between the number of proposed entities and true entities in a document. It was not tuned to maximize the F_1 measure. A document coefficient of 0.4 was chosen. The global concentration coefficient γ was chosen to be a constant proportion of αM, where M is the number of documents in the corpus. We found 0.15 to be a good value using the same least-square procedure. The values for these coefficients were not changed for the experiments in this section.

4.5.1 MUC-6

Our main evaluation is on the standard MUC-6 formal test set.[9] The standard experimental setup for this data is a 30/30 document train/test split. Training our system on all 60

[9]Since the MUC data is not annotated with mention types, we automatically detect this information in the same way as Luo et al. (2004). A mention is proper if it is annotated with NER information. It is a

HEAD	ENT TYPE	GENDER	NUMBER
Bush: 1.0	PERS	MALE	SG
AP: 1.0	ORG	NEUTER	PL
viacom: 0.64, *company*: 0.36	ORG	NEUTER	SG
teamsters: 0.22, *union*: 0.78,	MISC	NEUTER	PL

Table 4.3. Frequent entities occurring across documents along with head distribution and mode of property distributions.

documents (as this is in an unsupervised system, the unlabeled test documents are present at training time) of the training and test set, but evaluating only on the test documents gave 63.9 F_1 and is labeled MUC-6 in table 4.2(a).

One advantage of an unsupervised approach is that we can easily utilize more data when learning a model. We demonstrate the effectiveness of this fact by evaluating on the MUC-6 test documents with increasing amounts of unannotated training data. We first add the 191 documents from the MUC-6 dryrun training set (which were not part of the training data for official MUC-6 evaluation). This model gives 68.0 F_1 and is labeled +DRYRUN-TRAIN in table 4.2(a). We then experiment with adding the ACE ENGLISH-NWIRE training data, which is from different corpora than the MUC-6 test set and from a different time period. This model gives 70.3 F_1 and is labeled +ENGLISH-NWIRE in table 4.2(a).

Our results on this test set are surprisingly comparable to, though slightly lower than, some recent supervised systems. McCallum and Wellner (2005) report 73.4 F_1 on the formal MUC-6 test set, which is reasonably close to our best MUC-6 number of 70.3 F_1. McCallum and Wellner (2005) also report a 91.6 F_1 on only proper nouns mentions. Our system achieves a 89.8 F_1 when evaluation is restricted to only proper mentions.[10] The closest comparable unsupervised system is Cardie and Wagstaff (1999) who use pairwise NP distances to cluster document mentions. They report a 53.6 F_1 on MUC6 when tuning distance metric weights to maximize F_1 on the development set.

4.5.2 ACE 2004

We also perform experiments on ACE 2004 data. Due to licensing restrictions, we did not have access to the ACE 2004 formal development and test sets, and so the results presented are on the training sets.

pronoun if the head is on the list of English pronouns. Otherwise, it is a nominal mention. Note we do not use the NER information for any purpose but determining whether the mention is proper.

[10]The best results we know on the MUC-6 test set using the standard setting are due to Luo et al. (2004) who report a 81.3 F_1 (much higher than others). However, it is not clear this is a comparable number, due to the apparent use of gold NER features, which provide a strong clue to coreference. Regardless, it is unsurprising that their system, which has many rich features, would outperform ours.

We report results on the newswire section (NWIRE in table 4.2b) and the broadcast news section (BNEWS in table 4.2b). Note that for these datasets, our evaluations include the prenominal mention type which we did not consider when developing our model. This mention type is not present in the MUC-6 data.

We also tested our system on the Chinese newswire and broadcast news sections of the ACE 2004 training sets. We note that our relatively higher performance on Chinese compared to English is perhaps due to the lack of prenominal mentions in the Chinese data as well as the presence of fewer pronouns compared to English.

Our ACE results are difficult to compare exactly because we did not have access to the restricted formal test set. However, we can perform a rough comparison between our results on the training data (without using the training coreference annotation) to supervised work which has used the same training data (with coreference annotation) and evaluated on the formal test set. Denis and Baldridge (2007) report 67.1 F_1 and 69.2 F_1 on the English NWIRE and BNEWS respectively using true mention boundaries. While our system underperforms the supervised systems, its accuracy is nonetheless promising.

4.6 Discussion

We discuss general trends and errors of the model presented in this chapter as well as discuss external applications.

4.6.1 Error Analysis

The largest source of error in our system is between coreferent proper and nominal mentions. Emblematic of this kind of error are appositive usages e.g. *George W. Bush, president of the US, visited Idaho.* In our system, there is no pressure to label the proper and nominal mentions as coreferent. Indeed, the system learns that highly salient entities should *not* be realized as nominals. In Chapter 5, we present a system which corrects such errors by using syntactic constraints and a larger source of data. In Chapter 6, we incorporate this information into the kind of model presented in this chapter.

Another error of this sort can be seen in figure 4.2, where the *corporation* mention is not labeled coreferent with the *The Weir Group* mention. Examples such as these illustrate the regular (at least in newswire) phenomenon that nominal mentions are used with informative intent, even when the entity is salient and a pronoun could have been used unambiguously. This aspect of nominal mentions is entirely unmodeled in our system.

4.6.2 Global Coreference

Since we do not have labeled cross document coreference data, we cannot evaluate our system's cross document performance quantitatively. However, in addition to observing the within-document gains from sharing shown in section 4.3, we can manually inspect the most frequently occurring entities in our corpora. Table 4.3 shows some of the most frequently occurring entities across the English ACE NWIRE corpus. Note that *Bush* is the most frequent entity, though his (and others') nominal cluster *president* is its own entity. Merging of proper and nominal clusters does occur as can be seen in table 4.3.

4.6.3 Unsupervised NER

An advantage of a fully generative model is that it can be used to answer multiple kinds of queries about unobserved variables. One such unobserved variable in our model that is of interest is the entity type drawn for each proper mention. One such query we can ask is the entity type of each proper mention. Note that the way an entity becomes associated with a particular entity type is only via coreference with pronouns which are in turn correlated with entity types.

We can use our model to perform unsupervised named-entity-recognition (NER) tagging as follows: For each proper mention we use the mode of the generating entity's distribution over entity types. Note that in our model the only way an entity becomes associated with an entity type is by the pronouns used to refer to it.[11] In English, some personal pronouns are only associated with persons. However, it is more difficult to distinguish the non person entities (ORG,LOC,MISC). There are some hints however. The pronoun `we` is typically associated with organizations and the pronoun `there` and `here` are associated with locations. This information is partially hinted in the prior and partially learned (See Section 4.3.3).

If we evaluate our system as an unsupervised NER tagger for the proper mentions in the MUC-6 test set, it yields a per-label accuracy of 61.2% (on MUC labels). Although nowhere near the performance of state-of-the-art systems, this results beats a simple baseline of always guessing PERSON (the most common entity type) which yields 46.4%. This result is interesting given that at the outset, we had no intention of inferring entity types whatsoever.

Elsner et al. (2009) has presented an unsupervised NER model which utilizes the insights presented in this section along with adding a mechanism for using distribution cues. This model yields substantial unsupervised NER accuracies and many of these ideas for unsupervised NER will impact the model presented in Chapter 6.

[11] Ge et al. (1998) exploit a similar idea to assign gender to proper mentions.

4.7 Conclusion

In this chapter we have presented a Bayesian non-parametric model for unsupervised entity reference resolution. The work presented in this chapter was the first unsupervised generative model approach to entity reference resolution. It utilizes the infinite mixture model interpretation of the Dirichlet Process (Teh et al., 2006) in order to flexibly model the number of entities associated with a document. The hierarchichal Dirichlet process extension is used to allow model entities to be shared across documents. Global entities are shared across documents, the number of entities is determined by the model, mention generation is modeled with a sequential salience model and a model of pronoun-entity association. Although our model does not perform quite as well as state-of-the-art supervised systems, its performance is in the same general range despite being unsupervised.

Chapter 5

Simple Entity Reference Resolution with Rich Syntactic and Semantic Features

5.1 Introduction

In Chapter 4, we presented a machine learning heavy approach to entity reference resolution, which made relatively few assumptions and did not directly enforce any known linguistic constraints. Of course entity reference is influenced by a variety of known constraints. Syntactic constraints like the binding theory, the i-within-i filter, and appositive constructions restrict reference by configuration (Lees and Kilma, 1963; Langacker, 1969). Semantic constraints like selectional compatibility (e.g. a *spokesperson* can *announce* things) and subsumption (e.g. *Microsoft* is a *company*) rule out many possible referents (Hobbs, 1977). Finally, discourse phenomena such as salience and centering theory are assumed to heavily influence reference preferences (Grosz et al., 1995). As these varied factors have given rise to a multitude of weak features, recent entity resolution work has focused on how best to learn to combine them using models over reference structures (Culotta et al., 2007; Denis and Baldridge, 2007; Klenner and Ailloud, 2007; Bengston and Roth, 2008).

In this chapter, we break from the standard view. Instead, we consider a vastly more modular system in which coreference is predicted from a deterministic function of a few rich features and constraints. One of the goals of this system is to separate what constraints and cues need to be learned and which are simpler to declaratively incorporate. In particular, we assume a three-step process. First, a self-contained syntactic module carefully represents

syntactic structures using an augmented parser and extracts syntactic paths from mentions to potential antecedents. Some of these paths can be ruled in or out by deterministic but conservative syntactic constraints. Importantly, the bulk of the work in the syntactic module is in making sure the parses are correctly constructed and used, and this module's most important training data is a treebank. Second, a self-contained semantic module evaluates the semantic compatibility of headwords and individual names. These decisions are made from compatibility lists extracted from unlabeled data sources such as newswire and web data. Finally, of the antecedents which remain after rich syntactic and semantic filtering, reference is chosen to minimize tree distance.

This procedure is trivial where most systems are rich, and so does not need any supervised coreference data; in fact, it has no learned or tuned parameters. However, it is rich in important ways which we argue are marginalized in recent entity resolution work. Interestingly, error analysis from our final system shows that its failures are far more often due to syntactic failures (e.g. parsing mistakes) and semantic failures (e.g. missing knowledge) than failure to model discourse phenomena or appropriately weigh conflicting evidence.

One contribution of the system presented in this chapter is the exploration of strong modularity, including the result that this system beats all previous reported unsupervised results and approaches the state of the art in supervised ones.[1] Another contribution of this chapter is the error analysis result that, even with substantial syntactic and semantic richness, the path to greatest improvement appears to be to further improve the syntactic and semantic modules. Finally, we offer our approach as a very strong, yet easy to implement, baseline for future reference resolution research. We make no claim that learning to reconcile disparate features in a joint model offers no benefit, only that it must not be pursued to the exclusion of rich, non-reference analysis. In fact, the observations made from the shortcomings of this system will be incorporated into a richer learned model presented in Chapter 6.

5.2 Experimental Setup

In this chapter, we evaluate our system using the following data sets:

Development: (see Section 5.3)

- **ACE2004-ROTH-DEV**: Dev set split of the ACE 2004 training set utilized in Bengston and Roth (2008). The ACE data also annotates pre-nominal mentions which we map onto nominals. 68 documents and 4,536 mentions.

Testing: (see Section 5.4)

[1]The model presented in Chapter 6 has, at the time of this writing, the best published performance on the end-to-end reference resolution task

- **ACE2004-CULOTTA-TEST**: Test set split of the ACE 2004 training set utilized in Culotta et al. (2007) and Bengston and Roth (2008). Consists of 107 documents.[2]
- **ACE2004-NWIRE**: ACE 2004 Newswire set to compare against Poon and Domingos (2008). Consists of 128 documents and 11,413 mentions; intersects with the other ACE data sets.
- **MUC-6-TEST**: MUC6 formal evaluation set consisting of 30 documents and 2,068 mentions.

Unlabeled: (see Section 5.3.2)

- **BLIPP**: 1.8 million sentences of newswire parsed with the Charniak (2000) parser. No labeled coreference data; used for mining semantic information.
- **WIKI**: 25k articles of English Wikipedia abstracts parsed by the Klein and Manning (2003) parser.[3] No labeled coreference data; used for mining semantic information.

5.3 System Description

In this section we develop our system and report developmental results on the ACE2004-ROTH-DEV dataset (see Section 6.5.1); we report pairwise F_1 figures here, but report on all evaluation metrics (described in Section 2.2) in Section 5.4. At a high level, our system resembles a pairwise coreference model (Soon et al., 2001; Ng and Cardie, 2002; Bengston and Roth, 2008) described in Section 3.1.1; for each mention m_i, we select either a single-best antecedent amongst the previous mentions $m_1, \ldots, m_{i-1}$, or the NULL mention to indicate the underlying entity has not yet been evoked. Mentions are linearly ordered according to the position of the mention head with ties being broken by the larger node coming first. While much research (Ng and Cardie, 2002; Culotta et al., 2007; Haghighi and Klein, 2007; Poon and Domingos, 2008; Finkel and Manning, 2008) has explored how to reconcile pairwise decisions to form coherent clusters, we simply take the transitive closure of our pairwise decision (as in Soon et al. (2001),Ng and Cardie (2002) and Bengston and Roth (2008)) which can and does cause system errors.

In contrast to these pairwise approaches, our pairwise decisions are not made with a learned model which outputs a probability or confidence, but instead for each mention m_i, we select an antecedent amongst $m_1, \ldots, m_{i-1}$ or the NULL mention as follows:

- **Syntactic Constraint:** Based on syntactic configurations, either force or disallow coreference between the mention and an antecedent. Propagate this constraint (see Figure 5.4).

[2]The evaluation set was not made available to non-participants.

[3]Wikipedia abstracts consist of roughly the first paragraph of the corresponding article

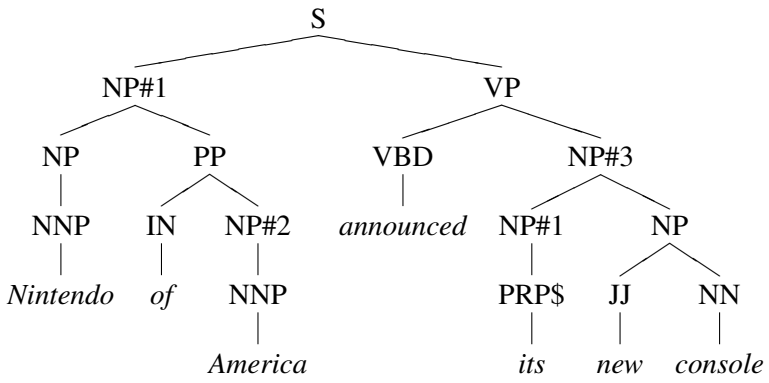


Figure 5.1. Example sentence which demonstrates where using tree distance rather than raw distance can be beneficial for for antecedent identification. In this example, the mention `its` is closest to `America` in raw token distance, but is closer to the NP headed by `Nintendo`, which is the correct antecedent. For clarity, each mention NP is labeled with the underlying entity id.

- **Semantic/Syntactic Filter**: Filter the remaining possible antecedents based upon compatibility with the mention (see Figure 5.2).
- **Selection**: Select the 'closest' mention from the set of remaining possible antecedents (see Figure 5.1) or the NULL antecedent if empty.

Initially, there is no syntactic constraint (improved in Section 5.3.1), the antecedent compatibility filter allows proper and nominal mentions to corefer only with mentions that have the same head (improved in Section 5.3.2), and pronouns have no compatibility constraints (improved in Section 5.3.1). Mention heads are determined by parsing the given mention span with the Stanford parser (Klein and Manning, 2003) and using the Collins head rules (Collins, 1999); Poon and Domingos (2008) showed that using syntactic heads strongly outperformed a simple rightmost headword rule. The mention type is determined by the head POS tag: proper if the head tag is NNP or NNPS, pronoun if the head tag is PRP, PRP$, WP, or WP$, and nominal otherwise.

For the selection phase, we order mentions $m_1, \ldots, m_{i-1}$ according to the position of the head word and select the closest mention that remains after constraint and filtering are applied. This choice reflects the intuition of Hobbs (1979) and Grosz et al. (1995) that speakers only use pronominal mentions when there are not intervening compatible mentions. This system yields a rather low 48.9 pairwise F_1 (see BASE-FLAT in Table 5.2). There are many, primarily recall, errors made choosing antecedents for all mention types which we will address by adding syntactic and semantic constraints.

5.3.1 Adding Syntactic Information

In this section, we enrich the syntactic representation and information in our system to improve results. The model presented in Chapter 4 utilized an extremely crude representation of a mention. Essentially, in that chapter, a mention was associated only with a heuristically-chosen head word. In this chapter, one of our goals will be to have a richer representation of a mention and its context.

Syntactic Salience

We first focus on fixing the pronoun antecedent choices. A common error arose from the use of mention head token distance as a poor proxy for discourse salience. For instance consider the example in Figure 5.1, the mention `America` is closest to `its` in flat mention distance, but syntactically `Nintendo of America` holds a more prominent syntactic position relative to the pronoun which, as Hobbs (1977) argues, is key to discourse salience.

Mapping Mentions to Parse Nodes: In order to use the syntactic position of mentions to determine anaphoricity, we must associate each mention in the document with a parse tree node. We parse all document sentences with the Stanford parser, and then for each evaluation mention, we find the largest-span NP which has the previously determined mention head as its head.[4] We call this node the maximum projection of a given noun. Often, this results in a different, typically larger, mention span than is annotated in the data.

Now that each mention is situated in a parse tree, we utilize the length of the shortest tree path between mentions as our notion of distance. In particular, this fixes examples such as those in Figure 5.1 where the true antecedent has many embedded mentions between itself and the pronoun. This change by itself yields 51.7 pairwise F_1 (see BASE-TREE in Table 5.2), which is small overall, but reduces pairwise pronoun antecedent selection error from 51.3% to 42.5%.

Agreement Constraints

We now refine our compatibility filtering to incorporate simple agreement constraints between coreferent mentions. Since we currently allow proper and nominal mentions to corefer only with matching head mentions, agreement is only a concern for pronouns. Traditional linguistic theory stipulates that coreferent mentions must agree in number, person, gender, and entity type (e.g. animacy). Here, we implement person, number and entity type agreement.[5]

[4] If there is no NP headed by a given mention head, we add an NP over just that word.

[5] Gender agreement, while important for general entity reference resolution, did not contribute to the errors in our largely newswire data sets.

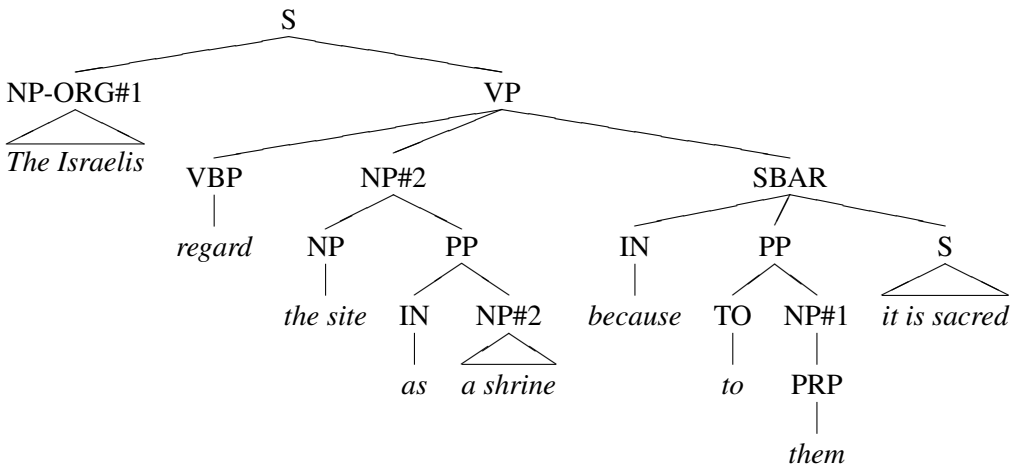


Figure 5.2. Example of a coreference decision fixed by agreement constraints (see Section 5.3.1). The pronoun *them* is closest to *the site* mention, but has an incompatible number feature with it. The closest (in tree distance, see Section 5.3.1) compatible mention is `The Israelis`, which is correct

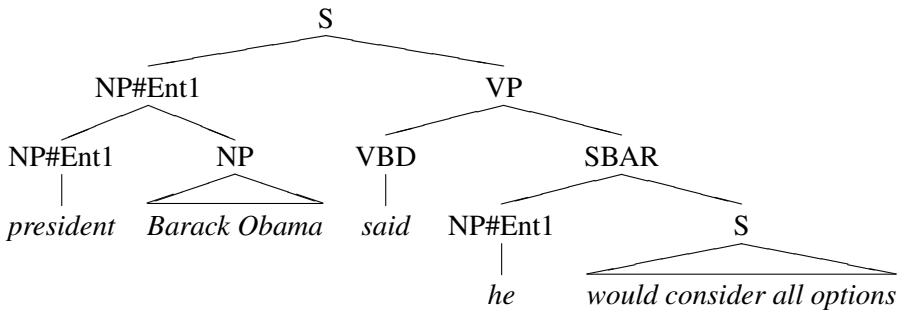
gore	president	florida	state
bush	governor	lebanese	territory
nation	people	arafat	leader
inc.	company	aol	company
nation	country	assad	president

Table 5.1. Most common recall (missed-link) errors amongst non-pronoun mention heads on our development set. Detecting compatibility requires semantic knowledge which we obtain from a large corpus (see Section 5.3.2).

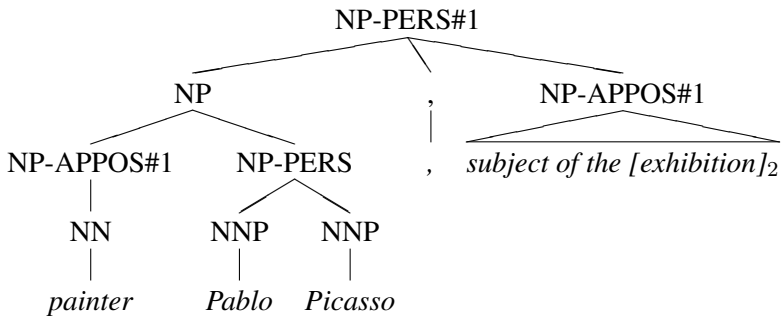
A number feature is assigned to each mention deterministically based on the head and its POS tag. For entity type, we use NER labels. Ideally, we would like to have information about the entity type of each referential NP, however this information is not easily obtainable. Instead, we opt to utilize the Stanford NER tagger (Finkel et al., 2005) over the sentences in a document and annotate each NP with the NER label assigned to that mention head. For each mention, when its NP is assigned an NER label we allow it to only be compatible with that NER label.[6] For pronouns, we deterministically assign a set of compatible NER values (e.g. personal pronouns can only be a PERSON, but *its* can be an ORGANIZATION or LOCATION). Since the NER tagger typically does not label non-proper NP heads, we have no NER compatibility information for nominals.

We incorporate agreement constraints by filtering the set of possible antecedents to those which have compatible number and NER types with the target mention. This yields 53.4 pairwise F_1, and reduces pronoun antecedent errors to 42.5% from 34.4%. An example of the type of error fixed by these agreement constraints is given by Figure 5.2.

[6]Or allow it to be compatible with all NER labels if the NER tagger doesn't predict a label.



(a)



(b)

Figure 5.3. NP structure annotation: In (a) we have the raw parse from the Klein and Manning (2003) parser with the mentions annotated by entity. In (b), we demonstrate the annotation we have added. NER labels are added to all NP according to the NER label given to the head (see Section 5.3.1). Appositive NPs are also annotated. Hashes indicate forced coreferent nodes

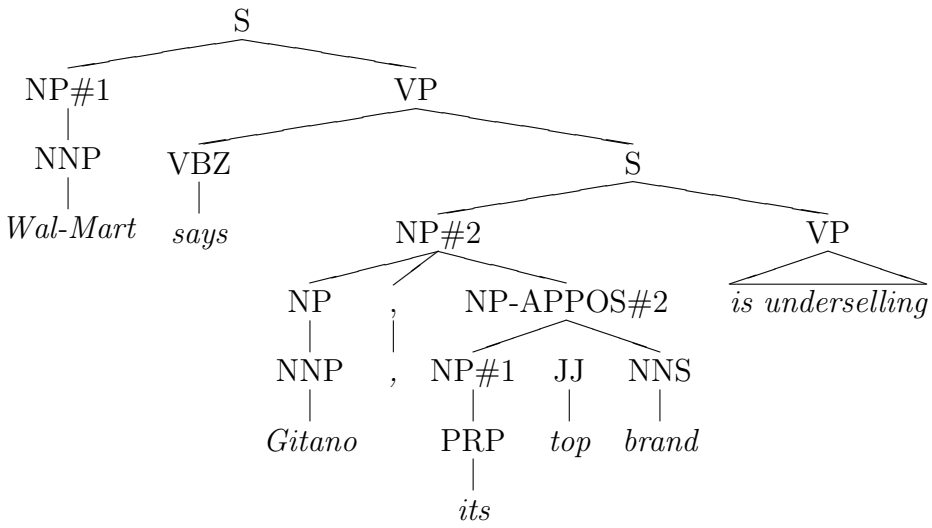


Figure 5.4. Example of interaction between the appositive and i-within-i constraint. The i-within-i constraint disallows coreference between parent and child NPs unless the child is an appositive. Hashed numbers indicate ground truth but are not in the actual trees.

Syntactic Configuration Constraints

Our system has so far focused only on improving pronoun anaphora resolution. However, a plurality of the errors made by our system are amongst non-pronominal mentions. There are over twice as many nominal mentions in our development data as pronouns. We take the approach that in order to align a non-pronominal mention to an antecedent without an identical head, we require evidence that the mentions are compatible.

Judging compatibility of mentions generally requires semantic knowledge, to which we return later. However, some syntactic configurations guarantee coreference between mentions. The one exploited most in coreference work (Soon et al., 2001; Ng and Cardie, 2002; Luo et al., 2004; Culotta et al., 2007; Poon and Domingos, 2008; Bengston and Roth, 2008) is the appositive construction. Here, we represent apposition as a syntactic feature of an NP indicating that it is coreferent with its parent NP (e.g. it is an exception to the i-within-i constraint that parent and child NPs *cannot* be coreferent). We deterministically mark a node as NP-APPOS (see Figure 5.3) when it is the third child in of a parent NP whose expansion begins with (NP , NP), and there is not a conjunction in the expansion (to avoid marking elements in a list as appositive).

Role Appositives: During development, we discovered many errors which involved a variant of appositives which we call 'role appositives' (see *painter* in Figure 5.3), where an NP modifying the head NP describes the role of that entity (typically a person entity). There are several challenges to correctly labeling these role NPs as being appositives. First, the NPs produced by Treebank parsers are flat and do not have the required internal structure (see Figure 5.3(a)). While fully solving this problem is difficult, we can heuristically fix many in-

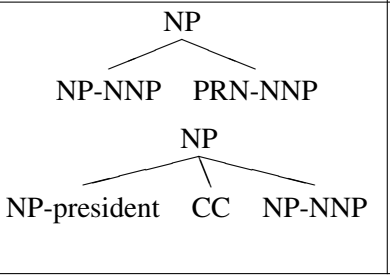
Path	Example
	*America Online Inc. (AOL)* *[President and C.E.O] Bill Gates*

Figure 5.5. Example paths extracted via semantic compatibility mining (see Section 5.3.2) along with example instantiations. In both examples the left child NP is coreferent with the rightmost NP. Each category in the interior of the tree path is annotated with the head word as well as its subcategorization. The examples given here collapse multiple instances of extracted paths.

stances of the problem by placing an NP around maximum length sequences of NNP tags or NN (and JJ) tags within an NP; note that this will fail for many constructions such as *U.S. President Barack Obama*, which is analyzed as a flat sequence of proper nouns. Once this internal NP structure has been added, whether the NP immediately to the left of the head NP is an appositive depends on the entity type. For instance, *Rabbi Ashi* is an apposition but *Iranian army* is not. Again, a full solution would require its own model, here we mark as appositions any NPs immediately to the left of a head child NP where the head child NP is identified as a person by the NER tagger.[7]

We incorporate NP appositive annotation as a constraint during filtering. Any mention which corresponds to an appositive node has its set of possible antecedents limited to its parent. Along with the appositive constraint, we implement the i-within-i constraint that any non-appositive NP cannot be be coreferent with its parent; this constraint is then propagated to any node its parent is forced to agree with. The order in which these constraints are applied is important, as illustrated by the example in Figure 5.4: First the list of possible antecedents for the appositive NP is constrained to only its parent. Now that all appositives have been constrained, we apply the i-within-i constraint, which prevents *its* from having the NP headed by *brand* in the set of possible antecedents, and by propagation, also removes the NP headed by *Gitano*. This leaves the NP *Wal-Mart* as the closest compatible mention.

Adding these syntactic constraints to our system yields 55.4 F_1, a fairly substantial improvement, but many recall errors remain between mentions with differing heads. Resolving such cases will require external semantic information, which we will automatically acquire (see Section 5.3.2).

Predicate Nominatives: Another syntactic constraint exploited in Poon and Domingos (2008) is the predicate nominative construction, where the object of a copular verb (forms of the verb *be*) is constrained to corefer with its subject (e.g. *Microsoft is a company in*

[7] Arguably, we could also consider right modifying NPs (e.g., *[Microsoft [Company]$_1$]$_1$*) to be role appositive, but we do not do so here.

Redmond). While much less frequent than appositive configurations (there are only 17 predicate nominatives in our development set), predicate nominatives are another highly reliable coreference pattern which we will leverage in Section 5.3.2 to mine semantic knowledge. As with appositives, we annotate object predicate-nominative NPs and constrain coreference as before. This yields a minor improvement to 55.5 F_1.

Ordering Syntactic Constraints

We use Figure 5.4 to describe the ordering of syntactic constraints. First, we apply syntactic constraints which force coreference: appositives and predicate nominative constructions. After this pass, the two NPs associated with entity #2 have been marked coreferent. Then when we search for an antecedent for any mentions which have not located an antecedent; for instance, the `its` in Figure 5.4. We then apply the i-within-i filter to remove potential antecedents. When doing this filter, we remove any antecedent which is a parent of the current mention, but also any mention which (according to the forced-coreference pass) has been marked coreferent with that parent. So in the case of Figure 5.4 and the `its` mention: we remove the `its top brand` mention since it is the parent of the mention, but also the mention headed by `Gitano` since it is coreferent with the `brand` mention (the two are in an appositive relation). This leaves the correct antecedent, `Wal-Mart`, as a potential antecedent.

5.3.2 Semantic Knowledge

While appositives and related syntactic constructions can resolve some cases of non-pronominal reference, most cases require semantic knowledge about the various entities as well as the verbs used in conjunction with those entities to disambiguate references (Kehler et al., 2008).

However, given a semantically compatible mention head pair, say *AOL* and *company*, one might expect to observe a reliable appositive or predicative-nominative construction involving these mentions somewhere in a large corpus. In fact, the Wikipedia page for *AOL*[8] has a predicate-nominative construction which supports the compatibility of this head pair: *AOL LLC (formerly America Online) is an American global Internet services and media company operated by Time Warner.*

In order to harvest compatible head pairs, we utilize our BLIPP and WIKI data sets (see Section 5.2), and for each noun (proper or common) and pronoun, we assign a maximal NP mention node for each nominal head as in Section 5.3.1; we then annotate appositive and predicate-nominative NPs as in Section 5.3.1. For any NP which is annotated as an appositive or predicate-nominative, we extract the head pair of that node and its constrained antecedent.

The resulting set of compatible head words, while large, covers a little more than half

[8] `http://en.wikipedia.org/wiki/AOL`

	MUC			B^3			Pairwise			CEAF		
System	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
ACE2004-ROTH-DEV												
BASIC-FLAT	73.5	66.8	70.0	80.6	68.6	74.1	63.6	39.7	48.9	68.4	68.4	68.4
BASIC-TREE	75.8	68.9	72.2	81.9	69.9	75.4	65.6	42.7	51.7	69.8	69.8	69.8
+SYN-COMPAT	77.8	68.5	72.9	84.1	69.7	76.2	71.0	43.1	53.4	69.8	69.8	69.8
+SYN-CONSTR	78.3	70.5	74.2	84.0	71.0	76.9	71.3	45.4	55.5	70.8	70.8	70.8
+SEM-COMPAT	77.9	74.1	75.9	81.8	74.3	77.9	68.2	51.2	58.5	72.5	72.5	72.5
ACE2004-CULOTTA-TEST												
BASIC-FLAT	68.6	60.9	64.5	80.3	68.0	73.6	57.1	30.5	39.8	66.5	66.5	66.5
BASIC-TREE	71.2	63.2	67.0	81.6	69.3	75.0	60.1	34.5	43.9	67.9	67.9	67.9
+SYN-COMPAT	74.6	65.2	69.6	84.2	70.3	76.6	66.7	37.2	47.8	69.2	69.2	69.2
+SYN-CONSTR	74.3	66.4	70.2	83.6	71.0	76.8	66.4	38.0	48.3	69.6	69.6	69.6
+SEM-COMPAT	74.8	**77.7**	**79.6**	79.6	**78.5**	79.0	**57.5**	57.6	57.5	**73.3**	**73.3**	**73.3**
Supervised Results												
Culotta et al. (2007)	-	-	-	86.7	73.2	79.3	-	-	-	-	-	-
Bengston and Roth (2008)	**82.7**	69.9	75.8	**88.3**	74.5	**80.8**	55.4	**63.7**	**59.2**	-	-	-
MUC6-TEST												
+SEM-COMPAT	**87.2**	**77.3**	**81.9**	84.7	**67.3**	**75.0**	**80.5**	**57.8**	**67.3**	72.0	72.0	72.0
Unsupervised Results												
Poon and Domingos (2008)	83.0	75.8	79.2	-	-	-	63.0	57.0	60.0	-	-	-
Supervised Results												
Finkel and Manning (2008)	89.7	55.1	68.3	**90.9**	49.7	64.3	74.1	37.1	49.5	-	-	-
ACE2004-NWIRE												
+SEM-COMPAT	**77.0**	**75.9**	**76.5**	79.4	74.5	76.9	**66.9**	**49.2**	**56.7**	71.5	71.5	71.5
Unsupervised Results												
Poon and Domingos (2008)	71.3	70.5	70.9	-	-	-	62.6	38.9	48.0	-	-	-

Table 5.2. Experimental Results (See Section 5.4): When comparisons between systems are presented, the largest result is bolded. The CEAF measure has equal values for precision, recall, and F_1.

of the examples given in Table 5.1. The problem is that these highly-reliable syntactic configurations are too sparse and cannot capture all the entity information present. For instance, the first sentence of Wikipedia abstract for *Al Gore* is:

> *Albert Arnold "Al" Gore, Jr. is an American environmental activist who served as the 45th Vice President of the United States from 1993 to 2001 under President Bill Clinton.*

The required lexical pattern X `who served as` Y is a general appositive-like pattern that almost surely indicates coreference. Rather than opt to manually create a set of these coreference patterns as in Hearst (1992), Snow et al. (2005) and Phillips and Riloff (2007). We We take a simple bootstrapping technique: given a set of mention pairs extracted from appositives and predicate-nominative configurations, we extract counts over tree fragments between nodes which have occurred in this set of head pairs (see Figure 5.5); the tree fragments are formed by annotating the internal nodes in the tree path with the head word and POS along with the subcategorization. We limit the paths extracted in this way in

several ways: paths are only allowed to go between adjacent sentences and have a length of at most 10. We then filter the set of paths to those which occur more than a hundred times and with at least 10 distinct seed head word pairs.

We allow a non-pronominal mention to match with an antecedent mention if it either is head identical (our old semantic compatibility) or if the mention head and antecedent pair appear on list obtained in this way. The vast majority of the extracted fragments are variants of traditional appositives and predicate-nominatives with some of the structure of the NPs specified. Extracting appositives from large corpora allow this information to be applied in documents which do not explicitly use an appositive. However there are some tree fragments which correspond to the novel coreference patterns (see Figure 5.5) of parenthetical alias as well as conjunctions of roles in NPs.

We apply our extracted tree fragments to our BLIPP and WIKI data sets and extract a set of compatible word pairs which match these fragments; these words pairs will be used to relax the semantic compatibility filter (see the start of the section); mentions are compatible with prior mentions with the same head or with a semantically compatible head word. This yields 58.5 pairwise F_1 (see SEM-COMPAT in Table 5.2) as well as similar improvements across other metrics. It is relatively important that the corpora used to induce semantic compatibility lists include a source such as WIKI since many semantic compatibilities which are considered background information in newswire, for instance, are explicitly stated in WIKI.

By and large the word pairs extracted in this way are correct (in particular we now have coverage for over two-thirds of the head pair recall errors from Table 5.1.) There are however word-pairs which introduce errors. In particular city-state constructions (e.g. `Los Angeles, California`) appears to be an appositive and incorrectly allows our system to have `angeles` as an antecedent for `california`.[9] Another common error is that the `%` symbol is made compatible with a wide variety of common nouns in the financial domain.

5.4 Experimental Results

We present formal experimental results here (see Table 5.2). We first evaluate our model on the ACE2004-CULOTTA-TEST dataset used in the state-of-the-art systems from Culotta et al. (2007) and Bengston and Roth (2008). Both of these systems were supervised systems discriminatively trained to maximize B^3 and used features from many different structured resources including WordNet, as well as domain-specific features (Culotta et al., 2007). Our best B^3 result of 79.0 is broadly in the range of these results. We should note that in our work we use neither the gold mention types (we do not model pre-nominals separately) nor do we use the gold NER tags which Bengston and Roth (2008) does. Across metrics, the syntactic constraints and semantic compatibility components contribute most to the overall final result.

[9] In principle, one can remove these patterns by disallowing appositives between two proper mentions.

	PROPER	NOMINAL	PRONOUN	NULL	TOTAL
PROPER	21/451	8/20	-	72/288	101/759
NOMINAL	16/150	99/432	-	158/351	323/933
PRONOUN	29/149	60/128	15/97	1/2	105/376

Table 5.3. Errors for each type of antecedent decision made by the system on the development set. Each row is a mention type and the column the predicted mention type antecedent. The majority of errors are made in the NOMINAL category. The total number of mentions in each type is given by the denominator in the TOTAL column.

Mention Type	Sem. Compat	Syn. Compat	Head	Intenal NP
NOMINAL	7	-	5	6
PRONOUN	6	3	-	6
PROPER	6	-	3	4
	Prag / Disc.	Process Error	Other	Comment
NOMINAL	2	2	1	2 general appos. patterns
PRONOUN	3	3	3	2 cataphora
PROPER	4	4	1	

Table 5.4. Error analysis on ACE2004-CULOTTA-TEST data by mention type. The dominant errors are in either semantic or syntactic compatibility of mentions rather than discourse phenomena. See Section 5.5.

On the MUC6-TEST dataset, our system outperforms both Poon and Domingos (2008) (an unsupervised Markov Logic Network system which uses explicit constraints) and Finkel and Manning (2008) (a supervised system which uses ILP inference to reconcile the predictions of a pairwise classifier) on all comparable measures.[10] Similarly, on the ACE2004-NWIRE dataset, we also outperform the state-of-the-art unsupervised system of Poon and Domingos (2008).

Overall, we conclude that our system outperforms state-of-the-art unsupervised systems[11] and is in the range of the state-of-the art systems of Culotta et al. (2007) and Bengston and Roth (2008).

5.5 Error Analysis

There are several general trends to the errors made by our system. Table 5.3 shows the number of pairwise errors made on MUC6-TEST dataset by mention type; note these errors

[10]Klenner and Ailloud (2007) took essentially the same approach but did so on non-comparable data.

[11]Poon and Domingos (2008) outperformed Haghighi and Klein (2007) (presented in Chapter 4). Unfortunately, we cannot compare against Ng (2008) since we do not have access to the version of the ACE data used in their evaluation.

are not equally weighted in the final evaluations because of the transitive closure taken at the end. The most errors are made on nominal mentions with pronouns coming in a distant second. In particular, we most frequently say a nominal is NULL when it has an antecedent; this is typically due to not having the necessary semantic knowledge to link a nominal to a prior expression.

In order to get a more thorough view of the cause of pairwise errors, we examined 20 random errors made in aligning each mention type to an antecedent. We categorized the errors as follows:

- Sem. Compat: Missing information about the compatibility of two words e.g. *pay* and *wage*. For pronouns, this is used to mean that we incorrectly aligned a pronoun to a mention with which it is not semantically compatible (e.g. *he* aligned to *board*).
- Syn. Compat: Error in assigning linguistic features of nouns for compatibility with pronouns (e.g. disallowing *they* to refer to *team*).
- Head: Errors involving the assumption that mentions with the same head are always compatible. Includes modifier and specificity errors such as allowing *Lebanon* and *Southern Lebanon* to corefer. This also includes errors of definiteness in nominals (e.g. *the people in the room* and *Chinese people*). Typically, these errors involve a combination of missing syntactic and semantic information.
- Internal NP: Errors involving lack of internal NP structure to mark role appositives (see Section 5.3.1).
- Prag. / Disc.: Errors where discourse salience or pragmatics are needed to disambiguate mention antecedents.
- Process Error: Errors which involved a tokenization, parse, or NER error.

The result of this error analysis is given in Table 5.4; note that a single error may be attributed to more than one cause. Despite our efforts in Section 5.3 to add syntactic and semantic information to our system, the largest source of error is still a combination of missing semantic information or annotated syntactic structure rather than the lack of discourse or salience modeling.

Our error analysis suggests that in order to improve the state-of-the-art in coreference resolution, future research should consider richer syntactic and semantic information than typically used in current systems.

5.6 Conclusion

This chapter has focused on utilizing richer syntactic and semantic representations to improve entity resolution performance rather than on machine learning techniques. The

approach here is not intended as an argument against the more complex,discourse-focused approaches that typify recent work. Instead, we note that rich syntactic and semantic processing vastly reduces the need to rely on discourse effects or evidence reconciliation for reference resolution. Indeed, in Chapter 6 we will further enrich the syntactic and semantic information used by a system and this approach will in fact yield greater error reductions than any other route forward.

Nonetheless, the system described in this chapter, despite being relatively simple and having no tunable parameters or complexity beyond the non-reference complexity of its component modules, manages to outperform state-of-the-art unsupervised coreference resolution and be broadly comparable to state-of-the-art supervised systems.

Chapter 6

Entity Reference Resolution in a Modular Entity-Centered Model

6.1 Introduction

Entity reference resolution systems exploit a variety of information sources, ranging from syntactic and discourse constraints, which are highly configurational, to semantic constraints, which are highly contingent on lexical meaning and world knowledge. Perhaps because configurational features are inherently easier to learn from small data sets, past work has often emphasized them over semantic knowledge.

Of course, all state-of-the-art reference resolution systems have needed to capture semantic compatibility to some degree; indeed, some of the earliest work in reference resolution (Hobbs, 1977) has recognized this need. For instance, the system presented in Chapter 5 mined semantically compatible head words, which yielded performance gains. As an example of nominal headword compatibility, a `president` can be a `leader` but cannot be not an `increase`. Other past systems, including that in Chapter 5, have computed the compatibility of specific headword pairs, extracted either from lexical resources (Ng, 2007; Bengston and Roth, 2008; Rahman and Ng, 2009), web statistics (Yang et al., 2005), or surface syntactic patterns (Haghighi and Klein, 2009). While this pairwise approach to semantic compatibility has high precision, it is neither realistic nor scalable to explicitly enumerate all pairs of compatible word pairs. A more compact approach has been to rely on named-entity recognition (NER) systems to give coarse-grained entity types for each mention (Soon et al., 2001; Ng and Cardie, 2002). In this approach, we can prefer, for instance, a personal pronoun, such as `he`, to take a proper mention that has been tagged as a person by an NER tagger. Unfortunately, current systems use small inventories of types and so provide little constraint.

Another shortcoming of this approach is that NER systems typically tag only proper mentions. They do not for instance know whether `legislator` or `desk` is more likely to be a person. The purely type-based approach also fails to capture specific facts about individual entities (e.g. `Obama` is a `president`). In general, coreference errors in state-of-the-art systems are frequently due to poor models of semantic compatibility (Haghighi and Klein, 2009).

In this chapter, we present an unsupervised generative entity reference resolution model, broadly along the lines of Chapter 4, which exploits semantic information as well as discourse and syntactic configurations. This model combines the strengths of entity-centered approaches (see Section 3.1.2) with pairwise approaches (see Section 3.1.1).

The semantic consistency of an entity will be ensured by a novel entity-centered semantic module. This semantic module exploits a large inventory of distributional entity types, including standard NER types like PERSON and ORGANIZATION, as well as more refined types like WEAPON and VEHICLE. For each type, distributions over typical heads, modifiers, and governors are learned from large amounts of unlabeled data, capturing type-level semantic information (e.g. `spokesman` is a likely head for a PERSON). In particular, this generalizes the semantic compatibility approach described in Chapter 5. In addition to representing information about general types, our model also represents information about specific entities. Each entity inherits from a type but captures entity-level semantic information (e.g. `giant` may be a likely head for the Microsoft entity but not all ORGs).

Separately from the type-entity semantic module, a log-linear discourse model captures discourse and syntactic configurational effects which past work has successfully captured (Soon et al., 2001; Ng and Cardie, 2002; Bengston and Roth, 2008; Haghighi and Klein, 2009). This discourse component resembles an unsupervised pairwise system, where a given mention selects an antecedent (or is the first mention of a new entity). Crucially, this pairwise approach is only used to model the discourse and syntactic patterns (and constraints) of coreference rather than to be responsible for generating the text of mentions. Mention text is generated by the semantic module to ensure global semantic consistency.

Finally, a mention model assembles each textual mention by selecting semantically appropriate words from the entities and types. Despite being almost entirely unsupervised, our model yields the best reported end-to-end results on a range of standard reference resolution data sets.

6.2 Key Abstractions

In this section we describe the key abstractions of our model. Illustrations of each abstractions can be found in Figure 6.1.

Mentions: A mention is an observed textual reference to a latent real-world entity. Mentions are associated with nodes in a parse tree and are typically realized as NPs. There

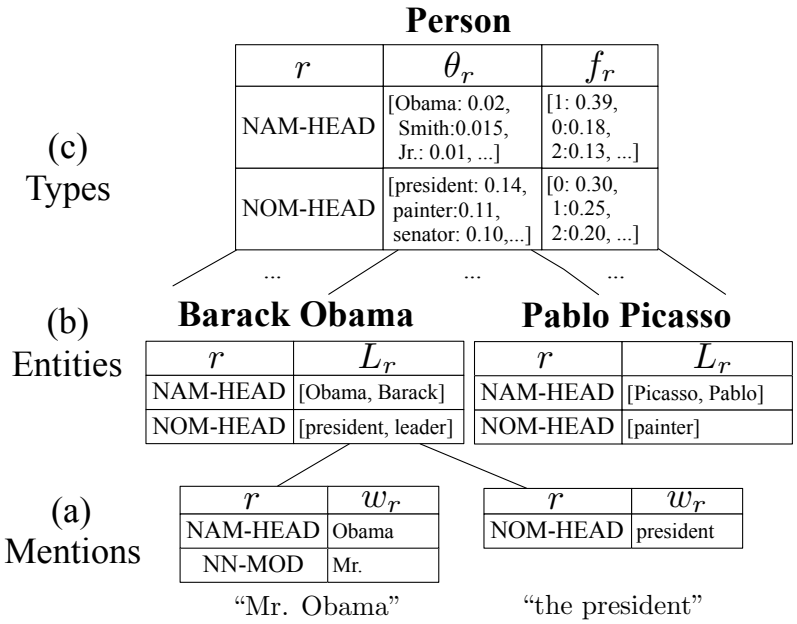


Figure 6.1. The key abstractions of our model (Section 6.2). (a) Mentions map properties (r) to words (w_r). (b) Entities map properties (r) to word lists (L_r). (c) Types map properties (r) to distributions over property words (θ_r) and the fertilities of those distributions (f_r). For (b) and (c), we only illustrate a subset of the properties.

are three basic forms of mentions:[1] proper (denoted NAM), nominal (NOM), and pronominal (PRO). We will often describe proper and nominal mentions together as *referring* mentions.

We represent each mention M as a collection of key-value pairs. The keys are called *properties* and the values are words. For example, the left mention in Figure 6.1(a) represents the string `Mr. Obama`. Under our representation, this mention consists of two key-value pairs. It has a proper head property, denoted NAM-HEAD, with value `Obama`. As well as a nominal modifier, NN-MOD, with value `Mr.`. The set of properties we consider, denoted $\mathcal{R}$, includes several varieties of heads, modifiers, and governors (see Section 6.5.2 for details). Not every mention has a value for every property.

Entities: An entity is a specific individual or object in the world. Entities are always latent in text. Where a mention has a single word for a given property, an entity has a *list* of signature words. Formally, entities are mappings from properties $r \in \mathcal{R}$ to lists L_r of 'canonical' words which that entity uses for that property. For instance in Figure 6.1(b), the list of nominal heads for the Barack Obama entity includes `president`. This captures the tendency of an entity to reuse a small number of 'cannonical' words. For instance, for a person entity, the proper head property,NAM-HEAD, typically corresponds to the last name of the entity (at least in newswire the tendency is to use the last name to refer to a person). The nominal head property, NOM-HEAD, typically refers to the role or job a person entity plays; for the *Barack Obama* entity, nominal words include `president` and `leader`. Other

[1]In this chapter, to avoid confusion with entity types, we use *mention form* to refer to *mention type*.

person entities can certainly be evoked with these words, but these words are frequently used to refer to the *Barack Obama* entity.

Types: Entity reference resolution systems often make a mention / entity distinction; for instance, this distinction is present in Chapter 4 as well as other entity-based approaches Section 3.1.2. In this chapter, we extend this hierarchy to include *types*, which represent classes of entities (PERSON, ORGANIZATION, and so on). The primary purpose of types is to allow the sharing of properties across entities and mediate the generation of entities in our model (see Section 6.3.1 for an elaboration). See Figure 6.1(c) for a concrete example.

Whereas an entity represents a mapping from properties to a list of words, a type is a mapping from properties to *distributions* over words. Concretely, we represent each type τ as a mapping between properties r and pairs of multinomial distributions (θ_r, f_r). Together, these distributions control the property word lists L_r for entities of that type.

θ_r is a unigram distribution of words that are semantically licensed for property r. For instance in the PERSON type, the θ_r for the NAM-HEAD property represents, roughly, a distribution over possible last names for people. Similarly, the θ_r for the NOM-HEAD property is a distribution over possible roles and generic nominal descriptions for a person (see Figure 6.1).

f_r is a "fertility" distribution over the integers that characterizes entity list lengths. For example, for the type PERSON, θ_r for proper heads is quite flat (there are many last names) but f_r is peaked at 1 (people have a single last name). On the other hand, the f_r for the NOM-HEAD distribution is more likely to generate lists of length two or more, since it is more likely to have two or more words frequently associated with a person entity.

6.3 Generative Model

We now describe our generative model. At the parameter level, we have one parameter group for the types $\boldsymbol{\tau} = (\phi, \tau_1, \ldots, \tau_t)$, where ϕ is a multinomial prior over a fixed number t over types and the $\{\tau_i\}$ are the parameters for each individual type, described in greater detail below. A second group comprises log-linear parameters $\boldsymbol{\pi}$ over discourse choices, also described below. Together, these two groups are drawn according to $P(\boldsymbol{\tau}|\boldsymbol{\lambda})P(\boldsymbol{\pi}|\sigma^2)$, where $\boldsymbol{\lambda}$ and σ^2 are a small number of scalar hyper-parameters described fully in Section 6.4.

Conditioned on the parameters $(\boldsymbol{\tau}, \boldsymbol{\pi})$, a document is generated as follows: A *semantic module* generates a sequence $\mathbf{E}$ of entities. $\mathbf{E}$ is in principle infinite, though during inference only a finite number are ever instantiated. A *discourse module* generates a vector $\mathbf{Z}$ which assigns an entity index Z_i to each mention position i. Finally, a *mention generation module* independently renders the sequence of mentions ($\mathbf{M}$) from their underlying entities. The syntactic position and structure of mentions are treated as observed, including the mention forms (pronominal, etc.). We use $\mathbf{X}$ to refer to this ungenerated information. Our model

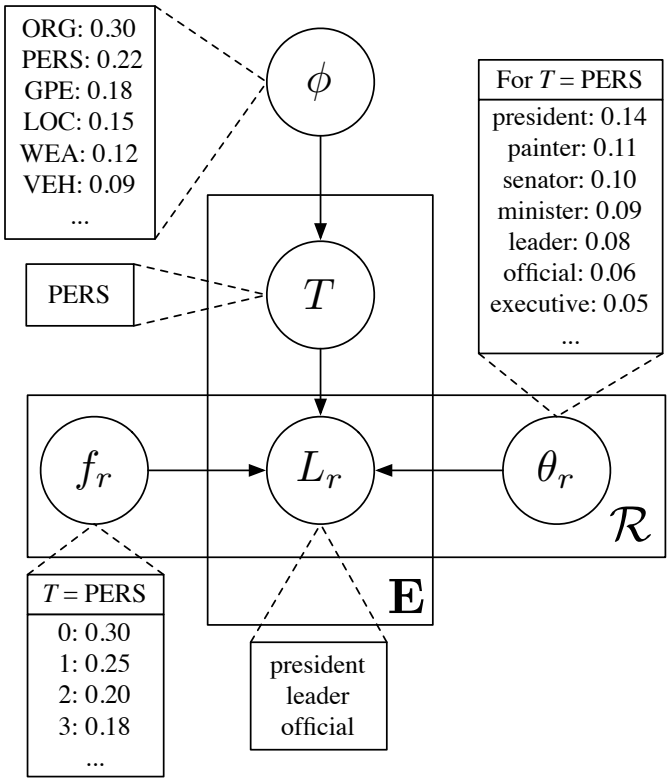


Figure 6.2. Depiction of the entity generation process (Section 6.3.1). Each entity draws a type (T) from ϕ, and, for each property $r \in \mathcal{R}$, forms a word list (L_r) by choosing a length from T's f_r distribution and then independently drawing that many words from T's θ_r distribution. Example values are shown for the person type and the nominal head property (NOM-HEAD).

decomposes as follows:

$$
\begin{aligned}
&P(\mathbf{E}, \mathbf{Z}, \mathbf{M} | \boldsymbol{\tau}, \boldsymbol{\pi}, \mathbf{X}) = \\
&\quad P(\mathbf{E}|\boldsymbol{\tau}) \text{ [Semantic, Section 6.3.1]} \\
&\quad P(\mathbf{Z}|\boldsymbol{\pi}, \mathbf{X}) \text{ [Discourse, Section 6.3.2]} \\
&\quad P(\mathbf{M}|\mathbf{Z}, \mathbf{E}, \boldsymbol{\tau}) \text{ [Mention, Section 6.3.3]}
\end{aligned}
$$

We detail each of these components in subsequent sections.

6.3.1 Semantic Module

The semantic module is responsible for the probability term $P(\mathbf{E}|\boldsymbol{\tau})$. Specifically, the semantic module generates a sequence of entities from the underlying type parameters $\boldsymbol{\tau}$. The entities are generated independently,

$$P(\mathbf{E}|\boldsymbol{\tau}) = \prod_{E \in \mathbf{E}} P(E|\boldsymbol{\tau}) \tag{6.1}$$

Each entity E is generated independently and consists of a type indicator T, as well as a collection $\{L_r\}_{r\in\mathcal{R}}$ of word lists for each property. We use L to denote this collection of word lists and so an entity consists of the pair (T, L).

Each entity, $E = (T, L)$ is generated as follows:

Entity Generation

Draw entity type $T \sim \phi$

For each mention property $r \in \mathcal{R}$,

Fetch $\{(f_r, \theta_r)\}$ for $\mathcal{T}_T$

Draw word list length $|L_r| \sim f_r$

Draw $|L_r|$ words from $w \sim \theta_r$

We can also represent this generative process as follows:

$$P(E = (T, L)|\boldsymbol{\tau}) = P(T|\phi) \prod_{r\in\mathcal{R}} P(L_r|\tau_T) \tag{6.2}$$

$$= P(T|\phi) \prod_{r\in\mathcal{R}} \left(P(|L_r||f_r) \prod_{w\in L_r} P(w|\theta_r) \right) [(f_r, \theta_r) \text{ from } \tau_T] \tag{6.3}$$

See Figure 6.2 for an illustration of this process. Each word list L_r is generated by first drawing a list length from f_r and then independently populating that list from the property's word distribution θ_r.[2] Past work has employed broadly similar distributional models for unsupervised NER of proper mentions (see Collins and Singer (1999) and Elsner et al. (2009)). However, to our knowledge, this is the first work to also label nominal expressions as well to incorporate such a model into an entity reference process.

6.3.2 Discourse Module

The discourse module (depicted in Figure 6.3) is responsible for choosing an entity to evoke at each of the n mention positions. Recall that in our model, the number of mention positions as well as their syntactic positions are treated as observed (denoted $\mathbf{X}$). Formally, this module generates an entity assignment vector $\mathbf{Z} = (Z_1, \ldots, Z_n)$, where Z_i indicates the entity index for the ith mention position. Most linguistic inquiry characterizes NP anaphora by the pairwise relations that hold between a mention and its antecedent (Hobbs, 1979; Kehler et al., 2008). Our discourse module utilizes this pairwise perspective to define each Z_i in terms of an intermediate "antecedent" variable A_i. A_i either points to a previous antecedent mention position ($A_i < i$) and "steals" its entity assignment or begins a new

[2]There is one exception: the sizes of the proper and nominal head property lists are jointly generated, but their word lists are still independently populated.

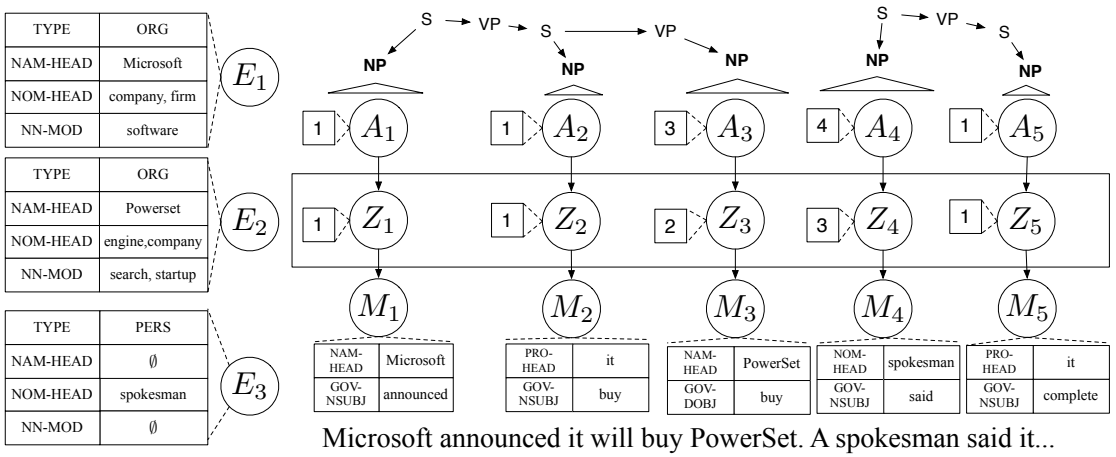


Figure 6.3. Depiction of the discourse module (Section 6.3.2), which generates the entity assignment vector $\mathbf{Z}$ as well as the mention module (Section 6.3.3), which is responsible for rendering mentions conditioned on entity assignments ($\mathbf{Z}$) and entities ($\mathbf{E}$).

entity ($A_i = i$). The choice of A_i is parametrized by affinities $s_{\boldsymbol{\pi}}(i, j; \mathbf{X})$ between mention positions i and j. Formally, this process is described as:

Entity Assignment

For each mention position, $i = 1, \ldots, n$,

Draw antecedent position $A_i \in \{1, \ldots, i\}$:

$$P(A_i = j | X) \propto s_{\boldsymbol{\pi}}(i, j; X)$$

$$Z_i = \begin{cases} Z_{A_i}, \text{ if } A_i < i \\ K + 1, \text{ otherwise} \end{cases}$$

Here, K denotes the number of entities allocated in the first i-1 mention positions. For each mention position, the discourse module either: (1) selects a prior antecedent and "steals" its entity assignment ($A_i < i$), or (2) begins a new entity ($A_i = i$). This choice is parametrized by the antecedent affinities $s_{\boldsymbol{\pi}}(i, j; \mathbf{X})$ between mention positions i and j. This process is an instance of the sequential distance-dependent Chinese Restaurant Process (DD-CRP) of Blei and Frazier (2009). During inference, we variously exploit both the A and Z representations (Section 6.4).

For nominal and pronoun mentions, there are several well-studied anaphora cues, including centering (Grosz et al., 1995), nearness (Hobbs, 1977), and deterministic constraints, which have all been utilized in prior coreference work (Soon et al., 2001; Ng and Cardie, 2002). In order to combine these cues, we take a log-linear, feature-based approach and parametrize $s_{\boldsymbol{\pi}}(i, j; X) = \exp\{\boldsymbol{\pi}^\top \mathbf{f}_{\mathbf{X}}(i, j)\}$, where $\mathbf{f}_{\mathbf{X}}(i, j)$ is a feature vector over mention positions i and j, and $\boldsymbol{\pi}$ is a parameter vector; the features may freely condition on $\mathbf{X}$. We utilize the following features between a mention and an antecedent: tree distance, sentence

distance, and the syntactic positions (subject, object, and oblique) of the mention and antecedent. Features for starting a new entity include: a definiteness feature (extracted from the mention's determiner), the top CFG rule of the mention parse node, its syntactic role, and a bias feature. These features are conjoined with the mention form (nominal or pronoun). Additionally, we restrict pronoun antecedents to the current and last two sentences, and the current and last three sentences for nominals. Additionally, we disallow nominals from having direct pronoun antecedents.

In addition to the above, if a mention is in a deterministic coreference configuration, as defined in Haghighi and Klein (2009), we force it to take the required antecedent. In general, antecedent affinities learn to prefer close antecedents in prominent syntactic positions. We also learn that new entity nominals are typically indefinite or have SBAR complements (captured by the CFG feature).

In contrast to nominals and pronouns, the choice of entity for a proper mention is governed more by entity frequency than antecedent distance. We capture this by setting $s_{\boldsymbol{\pi}}(i, j; \mathbf{X})$ in the proper case to 1 for past positions and to a fixed α otherwise. [3]

6.3.3 Mention Module

Once the semantic module has generated entities and the discourse model selects entity assignments, each mention M_i generates word values for a set of observed properties R_i:

Mention Generation

For each mention M_i, $i = 1, \ldots, n$

Fetch $(T, \{L_r\}_{r \in \mathcal{R}})$ from E_{Z_i}

Fetch $\{(f_r, \theta_r)\}_{r \in \mathcal{R}}$ from $\mathcal{T}_T$

For $r \in R_i$:

$$w \sim (1 - \alpha_r)\textsc{Uniform}(L_r) + (\alpha_r)\theta_r$$

For each property r, there is a hyper-parameter α_r which interpolates between selecting a word from the entity list L_r and drawing from the underlying type property distribution θ_r. Intuitively, a small value of α_r indicates that an entity prefers to re-use a small number of words for property r. This is typically the case for proper and nominal heads as well as modifiers. At the other extreme, setting α_r to 1 indicates the property isn't particular to the entity itself, but rather only on its type. We set α_r to 1 for pronoun heads as well as for the governor of the head properties.

[3] As Blei and Frazier (2009) notes, when marginalizing out the A_i in this trivial case, the DD-CRP reduces to the traditional CRP (Pitman, 2002), so our discourse model roughly matches Haghighi and Klein (2007) for proper mentions.

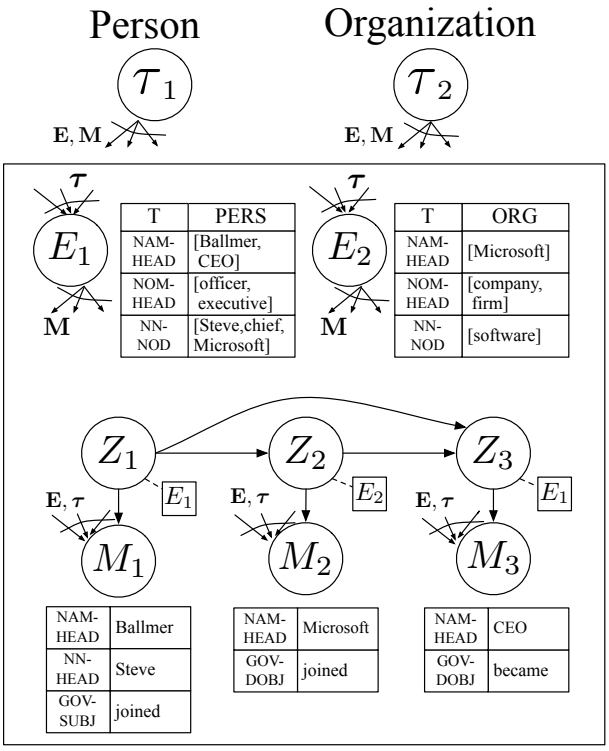


Figure 6.4. Depiction of the discourse module (Section 6.3.2); each random variable is annotated with an example value. For each mention position, an entity assignment (Z_i) is made. Conditioned on entities (E_{Z_i}), mentions (M_i) are rendered (Section 6.3.3). The arrow symbol denotes that a random variable is the parent of all $\mathbf{Y}$ random variables.

6.4 Learning and Inference

We provide a brief description of learning and inference for our model. For a fuller description, see Section C. Our learning procedure involves finding parameters and assignments which are likely under our model's posterior distribution $P(\mathbf{E}, \mathbf{Z}, \boldsymbol{\tau}, \boldsymbol{\pi} | \mathbf{M}, \mathbf{X})$. The model is modularized in such a way that running EM on all variables simultaneously would be very difficult. Therefore, we adopt a variational approach which optimizes various subgroups of the variables in a round-robin fashion, holding approximations to the others fixed. We first describe the variable groups, then the updates which optimize them in turn.

Decomposition: We decompose the entity variables $\mathbf{E}$ into types, $\mathbf{T}$, one for each entity, and word lists, $\mathbf{L}$, one for each entity and property. We decompose the mentions $\mathbf{M}$ into *referring* mentions (propers and nominals), $\mathbf{M}^r$, and pronominal mentions, $\mathbf{M}^p$ (with sizes n_r and n_p respectively). The entity assignments $\mathbf{Z}$ are similarly divided into $\mathbf{Z}^r$ and $\mathbf{Z}^p$ components. For pronouns, rather than use $\mathbf{Z}^p$, we instead work with the corresponding antecedent variables, denoted $\mathbf{A}^p$, and marginalize over antecedents to obtain $\mathbf{Z}^p$.

With these variable groups, we would like to approximation our model posterior $P(\mathbf{T}, \mathbf{L}, \mathbf{Z}^r, \mathbf{A}^p, \boldsymbol{\tau}, \boldsymbol{\pi} | \mathbf{M}, \mathbf{X})$ using a simple factored representation. Our variational approxi-

mation takes the following form:

$$Q(\mathbf{T}, \mathbf{L}, \mathbf{Z}^r, \mathbf{A}^p, \boldsymbol{\tau}, \boldsymbol{\pi}) = \delta_r(\mathbf{Z}^r, \mathbf{L}) \left(\prod_{k=1}^{n} q_k(T_k)\right) \left(\prod_{i=1}^{n_p} r_i(A_i^p)\right) \delta_s(\boldsymbol{\tau})\delta_d(\boldsymbol{\pi})$$

We use a mean field approach to update each of the RHS factors in turn to minimize the KL-divergence between the current variational posterior and the true model posterior. The δ_r, δ_s, and δ_d factors place point estimates on a single value, just as in hard EM. Updating these factors involves finding the value which maximizes the model (expected) log-likelihood under the other factors. For instance, the δ_s factor is a point estimate of the type parameters, and is updated with:[4]

$$\delta_s(\boldsymbol{\tau}) \leftarrow \operatorname{argmax}_{\boldsymbol{\tau}} \mathbb{E}_{Q_{-\delta_s}} \ln P(\mathbf{E}, \mathbf{Z}, \mathbf{M}, \boldsymbol{\tau}, \boldsymbol{\pi}) \tag{6.4}$$

where $Q_{-\delta_s}$ denotes all factors of the variational approximation except for the factor being updated. The r_i (pronoun antecedents) and q_k (type indicator) factors maintain a soft approximation and so are slightly more complex. For example, the r_i factor update takes the standard mean field form:

$$r_i(A_i^p) \propto \exp\{\mathbb{E}_{Q_{-r_i}} \ln P(\mathbf{E}, \mathbf{Z}, \mathbf{M}, \boldsymbol{\tau}, \boldsymbol{\pi})\} \tag{6.5}$$

We briefly describe the update for each additional factor, omitting details for space (see Appendix C).

Updating type parameters $\delta_s(\boldsymbol{\tau})$: The type parameters $\boldsymbol{\tau}$ consist of several multinomial distributions which can be updated by normalizing expected counts as in the EM algorithm. The prior $P(\boldsymbol{\tau}|\boldsymbol{\lambda})$ consists of several finite Dirichlet draws for each multinomial, which are incorporated as pseudocounts.[5] Given the entity type variational posteriors $\{q_k(\cdot)\}$, as well as the point estimates of the $\mathbf{L}$ and $\mathbf{Z}^r$ elements, we obtain expected counts from each entity's attribute word lists and referring mention usages.

Updating discourse parameters $\delta_d(\boldsymbol{\pi})$: The learned parameters for the discourse module rely on pairwise antecedent counts for assignments to nominal and pronominal mentions.[6] Given these expected counts, which can be easily obtained from other factors, the update reduces to a weighted maximum entropy problem, which we optimize using LBFGS. The prior $P(\boldsymbol{\pi}|\sigma^2)$ is a zero-centered normal distribution with shared diagonal variance σ^2, which is incorporated via L2 regularization during optimization.

[4]Of course during learning, the argmax is performed over the entire document collection, rather than a single document.

[5]See software release for full hyper-parameter details.

[6]Propers have no learned discourse parameters.

Updating referring assignments and word lists $\delta_r(\mathbf{Z}^r, \mathbf{L})$: The word lists are usually concatenations of the words used in nominal and proper mentions and so are updated together with the assignments for those mentions. Updating the $\delta_r(\mathbf{Z}^r, \mathbf{L})$ factor involves finding the referring mention entity assignments, $\mathbf{Z}^r$, and property word lists $\mathbf{L}$ for instantiated entities which maximize $\mathbb{E}_{Q_{-\delta_r}} \ln P(\mathbf{T}, \mathbf{L}, \mathbf{Z}^r, \mathbf{A}^p, \mathbf{M}, \boldsymbol{\tau}, \boldsymbol{\pi})$. We actually only need to optimize over $\mathbf{Z}^r$, since for any $\mathbf{Z}^r$, we can compute the optimal set of property word lists $\mathbf{L}$. Essentially, for each entity we can compute the L_r which optimizes the probability of the referring mentions assigned to the entity (indicated by $\mathbf{Z}^r$). In practice, the optimal L_r is just the set of property words in the assigned mentions. [7] Of course enumerating and scoring all $\mathbf{Z}^r$ hypotheses is intractable, so we instead utilize a left-to-right sequential beam search. Each partial hypothesis is an assignment to a prefix of mention positions and is scored as though it were a complete hypothesis. Hypotheses are extended via adding a new mention to an existing entity or creating a new one. For our experiments, we limited the number of hypotheses on the beam to the top fifty and did not notice an improvement in model score from increasing beam size.

Updating pronominal antecedents $r_i(A_i^p)$ and entity types $q_k(T_k)$: These updates are straightforward instantiations of the mean-field update (C.4).

To produce our final coreference partitions, we assign each referring mention to the entity given by the δ_r factor and each pronoun to the most likely entity given by the r_i.

6.4.1 Factor Staging

In order to facilitate learning, some factors are initially set to fixed heuristic values and only learned in later iterations. Initially, the assignment factors δ_r and $\{r_i\}$ are fixed. For δ_r, we use a deterministic entity assignment $\mathbf{Z}^r$, similar to the Haghighi and Klein (2009)'s SYN-CONSTR setting: each referring mention is coreferent with any past mention with the same head or in a deterministic syntactic configuration (appositives or predicative nominatives constructions). Forcing appositive coreference is essential for tying proper and nominal entity type vocabulary. The $\{r_i\}$ factors are heuristically set to place most of their mass on the closest antecedent by tree distance. During training, we proceed in stages, each consisting of 5 iterations:

Stage	Learned	Fixed	B^3All
1	$\delta_s, \delta_d, \{q_k\}$	$\{r_i\}$,δ_r	74.6
2	$\delta_s, \delta_d, \{q_k\}, \delta_r$	$\{r_i\}$	76.3
3	$\delta_s, \delta_d, \{q_k\}, \delta_r, \{r_i\}$	–	78.0

We evaluate our system at the end of stage using the B^3All metric on the A05CU development set (see Section 6.5 for details).

[7] For any property where $\alpha_r = 1$ (Section 6.3.3), the empty list is trivially optimal.

System	MUC P	MUC R	MUC F1	B^3All P	B^3All R	B^3All F1	B^3None P	B^3None R	B^3None F1	Pairwise F_1 P	Pairwise F_1 R	Pairwise F_1 F1
ACE2004-STOYANOV-TEST												
Stoyanov et al. (2009)	-	-	62.0	-	-	76.5	-	-	75.4	-	-	-
Haghighi and Klein (2009)	**67.5**	61.6	64.4	77.4	69.4	73.2	77.4	67.1	71.3	58.3	44.5	50.5
THIS WORK	67.4	**66.6**	**67.0**	**81.2**	**73.3**	**77.0**	**80.6**	**75.2**	**77.3**	59.2	50.3	54.4
ACE2005-STOYANOV-TEST												
Stoyanov et al. (2009)	-	-	67.4	-	-	73.7	-	-	72.5	-	-	-
Haghighi and Klein (2009)	73.1	58.8	65.2	82.1	63.9	71.8	81.2	61.6	70.1	66.1	37.9	48.1
THIS WORK	**74.6**	**62.7**	**68.1**	**83.2**	**68.4**	**75.1**	**82.7**	**66.3**	**73.6**	**64.3**	**41.4**	**50.4**
ACE2005-RAHMAN-TEST												
Rahman and Ng (2009)	75.4	64.1	69.3	-	-	-	**54.4**	70.5	61.4	-	-	-
Haghighi and Klein (2009)	72.9	60.2	67.0	53.2	73.1	61.6	52.0	72.6	60.6	57.0	44.6	50.0
THIS WORK	**77.0**	**66.9**	**71.6**	**55.4**	**74.8**	**63.8**	54.0	**74.7**	**62.7**	**60.1**	**47.7**	**53.0**

Table 6.1. Experimental results with system mentions. All systems except Haghighi and Klein (2009) and current work are fully supervised. The current work outperforms all other systems, supervised or unsupervised. For comparison purposes, the B^3None variant used on A05RA is calculated slightly differently than other B^3None results; see Rahman and Ng (2009).

6.5 Experiments

We considered the challenging end-to-end system mention setting, where in addition to predicting mention partitions, a system must identify the mentions themselves and their boundaries automatically. Our system deterministically extracts mention boundaries from parse trees (Section 6.5.2). We utilized no coreference annotation during training, but did use minimal prototype information to prime the learning of entity types (Section 6.5.3).

6.5.1 Datasets

For evaluation, we used standard coreference data sets derived from the ACE corpora:

- **A04CU**: Train/dev/test split of the newswire portion of the ACE 2004 training set[8] utilized in Culotta et al. (2007), Bengston and Roth (2008) and Stoyanov et al. (2009). Consists of 90/68/38 documents respectively.
- **A05ST**: Train/test split of the newswire portion of the ACE 2005 training set utilized in Stoyanov et al. (2009). Consists of 57/24 documents respectively.
- **A05RA**: Train/test split of the ACE 2005 training set utilized in Rahman and Ng (2009). Consists of 482/117 documents respectively.

For all experiments, we evaluated on the dev and test sets above. To train, we included the text of all documents above, though of course not looking at either their mention boundaries or reference annotations in any way. We also trained on the following much larger unlabeled datasets utilized in Haghighi and Klein (2009):

[8]Due to licensing restriction, the formal ACE test sets are not available to non-participants.

- **BLLIP**: 5k articles of newswire parsed with the Charniak (2000) parser.
- **WIKI**: 8k abstracts of English Wikipedia articles parsed by the Berkeley parser (Petrov et al., 2006). Articles were selected to have subjects amongst the frequent proper nouns in the evaluation datasets.

6.5.2 Mention Detection and Properties

Mention boundaries were automatically detected as follows: For each noun or pronoun (determined by parser POS tag), we associated a mention with the maximal NP projection of that head or that word itself if no NP can be found.[9] This procedure recovers over 90% of annotated mentions on the A05CU dev set, but also extracts many unannotated "spurious" mentions (for instance events, times, dates, or abstract nouns) which are not deemed to be of interest by the ACE annotation conventions. For evaluation, we suppress entities when the mode of the variational posterior over entity type is not amongst the prototyped entity types. This process reduces roughly half mention detection precision error by half and rarely incorrectly suppresses correct mention.

Mention properties were obtained from parse trees using the the Stanford typed dependency extractor (de Marneffe et al., 2006). The mention properties we considered are the mention head (annotated with mention type), the typed modifiers of the head, and the governor of the head (conjoined with the mention's syntactic position). We discard determiners, but make use of them in the discourse component (Section 6.3.2) for NP definiteness.

6.5.3 Prototyping Entity Types

While it is possible to learn type distributions in a completely unsupervised fashion, we found it useful to prime the system with a handful of important types. Rather than relying on fully supervised data, we took the approach of Haghighi and Klein (2006). For each type of interest, we provided a (possibly-empty) prototype list of proper and nominal head words, as well as a list of allowed pronouns. For instance, for the PERSON type we might provide:

NAM	Bush, Gore, Hussein
NOM	president, minister, official
PRO	he, his, she, him, her, you, ...

The prototypes were used as follows: Any entity with a prototype on any proper or nominal head word attribute list (Section 6.3.1) was constrained to have the specified type; i.e. the q_k factor (Section 6.4) places probability one on that single type. Similarly to Haghighi and Klein (2007) and Elsner et al. (2009), we biased these types' pronoun distributions to the allowed set of pronouns.

[9]The maximal projection of a word in a parse tree is the highest node in the parse tree which has the given word as its head as determined by Collins' head rules (Collins, 1999).

In general, the choice of entity types to prime with prototypes is a domain-specific question. For experiments here, we utilized the types which are annotated in the ACE coreference data: person (PERS), organization (ORG), geo-political entity (GPE), weapon (WEA), vehicle (VEH), location (LOC), and facility (FAC). Since the person type in ACE conflates individual persons with groups of people (e.g., *soldier* vs. *soldiers*), we added the group (GROUP) type and generated a prototype specification.[10]

We obtained our prototype list by extracting at most four common proper and nominal head words from the newswire portions of the 2004 and 2005 ACE training sets (A04CU and A05ST); we chose prototype words to be minimally ambiguous with respect to type.[11] When there are not at least three proper heads for a type (WEA for instance), we did not provide any proper prototypes and instead strongly biased the type fertility parameters to generate empty NAM-HEAD lists.

Because only certain semantic types were annotated under the arbitrary ACE guidelines, there are many mentions which do not fall into those limited categories. We therefore prototype (refinements of) the ACE types and then add an equal number of unconstrained "other" types which are automatically induced. A nice consequence of this approach is that we can simply run our model on *all* mentions, discarding at evaluation time any which are of non-prototyped types.

6.5.4 Evaluation

We evaluated on multiple coreference resolution metrics, as no single one is clearly superior, particularly in dealing with the system mention setting. We utilized MUC (Vilain et al., 1995), B^3All (Stoyanov et al., 2009), B^3None (Stoyanov et al., 2009), and Pairwise F_1 (see Section 2.2.1 for a fuller description of these metrics). The B^3All and B^3None are B^3 variants (Bagga and Baldwin, 1998) that differ in their treatment of spurious mentions. For Pairwise F1, precision measures how often pairs of predicted coreferent mentions are in the same annotated entity. We eliminated any mention pair from this calculation where both mentions were spurious.[12]

6.5.5 Results

Table 6.1 shows our results. We compared to two state-of-the-art supervised coreference systems. The Stoyanov et al. (2009) numbers represent their THRESHOLD_ESTIMATION setting and the Rahman and Ng (2009) numbers represent their highest-performing cluster ranking model. We also compared to the strong deterministic system of Haghighi and Klein

[10]These entities are given the GROUP subtype in the ACE annotation.

[11]Meaning those headwords were assigned to the target type for more than 75% of their usages.

[12]Note that we are still penalized for marking a spurious mention coreferent with an annotated one.

(2009).[13] Across all data sets, our model, despite being largely unsupervised, consistently outperforms these systems, which are the best previously reported results on end-to-end coreference resolution (i.e. including mention detection). Performance on the A05RA dataset is generally lower because it includes articles from blogs and web forums where parser quality is significantly degraded.

While Bengston and Roth (2008) do not report on the full system mention task, they do report on the more optimistic setting where mention detection is performed but non-gold mentions are removed for evaluation using an oracle. On this more lenient setting, they report 78.4 B^3 on the A04CU test set. Our model yields 80.3.

6.6 Analysis

We now discuss errors and improvements made by our system. One frequent source of error is the merging of mentions with explicitly contrasting modifiers, such as *new president* and *old president*. While it is not unusual for a single entity to admit multiple modifiers, the particular modifiers *new* and *old* are incompatible in a way that *new* and *popular* are not. Our model does not represent the negative covariance between these modifiers.

We compared our output to the deterministic system of Haghighi and Klein (2009). Many improvements arise from correctly identifying mentions which are semantically compatible but which do not explicitly appear in an appositive or predicate-nominative configuration in the data. For example, *analyst* and *it* cannot corefer in our system because *it* is not a likely pronoun for the type PERSON.

While the focus of our model is coreference resolution, we can also isolate and evaluate the type component of our model as an NER system. We test this component by presenting our learned model with boundary-annotated non-pronominal entities from the A05ST dev set and querying their predicted type variable T. Doing so yields 83.2 entity classification accuracy under the mapping between our prototyped types and the coarse ACE types. Note that this task is substantially more difficult than the unsupervised NER in Elsner et al. (2009) because the inventory of named entities is larger (7 vs. 3) and because we predict types over nominal mentions that are more difficult to judge from surface forms. In this task, the plurality of errors are confusions between the GPE (geo-political entity) and ORG entity types, which have very similar distributions.

[13] Haghighi and Klein (2009) reports on true mentions; here, we report performance on automatically detected mentions.

Chapter 7

An Entity-Level Approach to Information Extraction

7.1 Introduction

Template-filling information extraction (IE) systems must merge information across multiple sentences to identify all role fillers of interest. For instance in the MUC4 terrorism event extraction task, the entity filling the *individual perpetrator* role often occurs multiple times, variously as proper, nominal, or pronominal mentions. However, most template-filling systems (Freitag and McCallum, 2000; Phillips and Riloff, 2007) assign roles to individual textual mentions using only local context as evidence, leaving aggregation for post-processing. While prior work has acknowledged that entity reference resolution and discourse analysis are integral to accurate role identification, to our knowledge no model has been proposed which jointly models these phenomena.

In this chapter, we present an application of the model presented in Chapter 6 to template-filling information extraction. We present an entity-centered approach to template-filling IE problems. This model jointly merges surface mentions into underlying entities (coreference resolution) and assigns roles to those discovered entities. In the generative process proposed here, document entities are generated for each template role, along with a set of non-template entities. These entities then generate mentions in a process sensitive to both lexical and structural properties of the mention. Our joint coreference and entity-level model outperforms a discriminative mention-level baseline. Moreover, since this model is generative, it can naturally incorporate unannotated data, which further increases accuracy.

Figure 7.1(a) shows an example *template-filling* task from the corporate acquisitions

(a) Template

SELLER	BUSINESS	ACQUIRED	PURCHASER
CSR Limited	Oil and Gas	Delhi Fund	Esso Inc.

(b) Document

[**S** CSR] has said that [**S** it] has sold [**S** its] [**B** oil interests] held in [**A** Delhi Fund]. [**P** Esso Inc.] did not disclose how much [**P** they] paid for [**A** Dehli].

Figure 7.1. Example of Corporate Acquisitions role-filling task. In (a), an example template specifying the entities playing each domain role. In (b), an example document with coreferent mentions sharing the same role label. Note that pronoun mentions provide direct clues to entity roles.

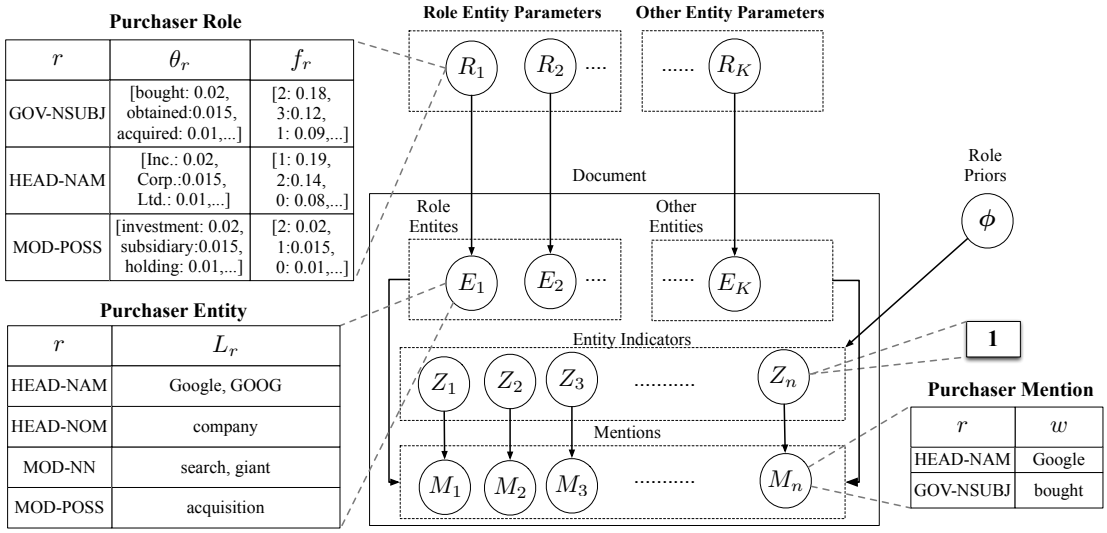


Figure 7.2. Graphical model depiction of our generative model described in Section 7.2. Sample values are illustrated for key parameters and latent variables.

domain (Freitag, 1998).[1] We have a template of K roles (PURCHASER, AMOUNT, etc.) and we must identify which entity (if any) fills each role (*CSR Limited*, etc.). Often such problems are modeled at the mention level, directly labeling individual mentions as in Figure 7.1(b). Indeed, in this data set, the mention-level perspective is evident in the gold annotations, which ignore pronominal references. However, roles in this domain appear in several locations throughout the document, with pronominal references often being the "give away" mentions. Therefore, Section 7.2 presents a model in which entities are explicitly modeled, naturally merging information across all mention types and explicitly representing latent structure very much like the entity-level template structure from Figure 7.1(a).

[1]In Freitag (1998), some of these fields are split in two to distinguish a full versus abbreviated name, but we ignore this distinction. Also we ignore the *status* field as it doesn't apply to entities and its meaning is not consistent.

7.2 Model

We describe our generative model for a document, which has many similarities to the coreference-only model presented in Chapter 6, but which integrally models template role-fillers. We review the basic abstractions we use and modify them to the current application:

Mentions: A mention is an observed textual reference to a latent real-world entity. Mentions are associated with nodes in a parse tree and are typically realized as NPs. There are three basic forms of mentions: proper (NAM), nominal (NOM), and pronominal (PRO). As in Chapter 6, each mention M is represented as collection of key-value pairs. The keys are called *properties* and the values are words. The set of properties utilized here, denoted $\mathcal{R}$, are the same as Haghighi and Klein (2010) and consist of the mention head, its dependencies, and its governor. See Figure 7.2 for a concrete example. Mention types are trivially determined from mention head POS tag. All mention properties and their values are observed.

Entities: An entity is a specific individual or object in the world. Entities are always latent in text. Where a mention has a single word for each property, an entity has a *list* of signature words. Formally, entities are mappings from properties $r \in \mathcal{R}$ to lists L_r of "canonical" words which that entity uses for that property.

Roles: Our model performs role-filling by assuming that each entity is drawn from an underlying role. These roles include the K template roles as well as "junk" roles to represent entities which do not fill a template role (see Section 7.4.2). Each role R is represented as mapping from between properties r and pairs of multinomials (θ_r, f_r). Together, these distributions control the lists L_r for entities which instantiate the role. θ_r is a unigram distribution of words for property r that are semantically licensed for the role (e.g. being the subject of "acquired" for the ACQUIRED role). f_r is a "fertility" distribution over the integers that characterizes entity list lengths.

Note that our notion or role corresponds to the entity type presented in Chapter 6. While the representations of these two elements are identical, the key difference is that in this work, since we will use labeled data, the parameters of our roles will be biased to distinguish different kinds of entities.

We first present a broad sketch of our model's components and then detail each in a subsequent section. We temporarily assume that all mentions belong to a template role-filling entity; we lift this restriction in Section 7.4.2. First, a semantic component generates a sequence of entities $\mathbf{E} = (E_1, \ldots, E_K)$, where each E_i is generated from a corresponding role R_i. We use $\mathbf{R} = (R_1, \ldots, R_K)$ to denote the vector of template role parameters. Note that this work assumes that there is a one-to-one mapping between entities and roles; in particular, at most one entity can fill each role which is appropriate for the domain considered here.

Once entities have been generated, a discourse component generates which entities will be evoked in each of the n mention positions. We represent these choices using *entity indicators* denoted by $\mathbf{Z} = (Z_1, \ldots, Z_n)$. This component utilized a learned global prior ϕ over roles. The Z_i indicators take values in $1, \ldots, K$ indicating the entity number (and thereby the role) underlying the ith mention position. Finally, a mention generation component renders each mention conditioned on the underlying entity and role. Formally, our decomposition is

similar to the Chapter 6:

$$P(\mathbf{E}, \mathbf{Z}, \mathbf{M}|\mathbf{R}, \phi) =$$

$$\left(\prod_{i=1}^{K} P(E_i|R_i)\right) \quad \text{[Semantic, Sec. 7.2.1]}$$

$$\left(\prod_{j=1}^{n} P(Z_j|\mathbf{Z}_{<j}, \phi)\right) \quad \text{[Discourse, Sec. 7.2.2]}$$

$$\left(\prod_{j=1}^{n} P(M_j|E_{Z_j}, R_{Z_j})\right) \quad \text{[Mention, Sec. 7.2.3]}$$

7.2.1 Semantic Component

Each role R generates an entity E as follows: for each mention property r, a word list, L_r, is drawn by first generating a list length from the corresponding f_r distribution in R.[2] This list is then populated by independent draw from R's unigram distribution θ_r. Formally, for each $r \in \mathcal{R}$, an entity word list is drawn according to,[3]

$$P(L_r|R) = P(\texttt{len}(L_r)|f_r) \prod_{w \in L_r} P(w|\theta_r)$$

7.2.2 Discourse Component

The discourse component draws the entity indicator Z_j for the jth mention according to,

$$P(Z_j|\mathbf{Z}_{<j}, \phi) = \begin{cases} P(Z_j|\phi), \text{ if non-pronominal} \\ \sum_{j'} \delta_{Z_j}(Z_{j'}) P(j'|j), \text{ o.w.} \end{cases}$$

When the jth mention is non-pronominal, we draw Z_j from ϕ, a global prior over the K roles. When M_j is a pronoun, we first draw an antecedent mention position j', such that $j' < j$, and then we set $Z_j = Z_{j'}$. The antecedent position is selected according to the distribution,

$$P(j'|j) \propto \exp\{-\gamma \textsc{TreeDist}(\text{j'},\text{j})\}$$

where $\textsc{TreeDist}(j',j)$ represents the tree distance between the parse nodes for M_j and $M_{j'}$.[4] Mass is restricted to antecedent mention positions j' which occur earlier in the same sentence or in the previous sentence. The sole parameter γ is fixed at 0.1. Were the focus of this task correctly identifying pronoun antecedents, a more complex model would be necessary.

[2]There is one exception: the sizes of the proper and nominal head property lists are jointly generated, but their word lists are still independently populated.

[3]While in principle, this process can yield word lists with duplicate words, we constrain the model during inference to not allow that to occur.

[4]Sentence parse trees are merged into a right-branching document parse tree. This allows us to extend tree distance to inter-sentence nodes.

7.2.3 Mention Generation

Once the entity indicator has been drawn, we generate words associated with mention conditioned on the underlying entity E and role R. For each mention property r associated with the mention, a word w is drawn utilizing E's word list L_r as well as the multinomials (f_r, θ_r) from role R. The word w is drawn according to,

$$P(w|E,R)=(1-\alpha_r)\frac{\mathbf{1}\left[w \in L_r\right]}{\texttt{len}(L_r)}+\alpha_r P(w|\theta_r)$$

For each property r, there is a hyper-parameter α_r which interpolates between selecting a word uniformly from the entity list L_r and drawing from the underlying role distribution θ_r. Intuitively, a small α_r indicates that an entity prefers to re-use a small number of words for property r. This is typically the case for proper and nominal heads as well as modifiers. At the other extreme, setting α_r to 1 indicates the property isn't particular to the entity itself, but rather always drawn from the underlying role distribution. We set α_r to 1 for pronoun heads as well as for the governor of the head properties.

7.3 Learning and Inference

Since we will make use of unannotated data (see Section 7.4), we utilize a variational EM algorithm to learn parameters $\mathbf{R}$ and ϕ. The E-Step requires the posterior $P(\mathbf{E}, \mathbf{Z}|\mathbf{R}, \mathbf{M}, \phi)$, which is intractable to compute exactly. We approximate it using a surrogate variational distribution of the following factored form:

$$Q(\mathbf{E}, \mathbf{Z}) = \left(\prod_{i=1}^{K} q_i(E_i)\right)\left(\prod_{j=1}^{n} r_j(Z_j)\right)$$

Each $r_j(Z_j)$ is a distribution over the entity indicator for mention M_j, which approximates the true posterior of Z_j. Similarly, $q_i(E_i)$ approximates the posterior over entity E_i which is associated with role R_i. As is standard, we iteratively update each component distribution to minimize KL-divergence, fixing all other distributions:[5]

$$\begin{aligned} q_i &\leftarrow \arg\min_{q_i} KL(Q(\mathbf{E}, \mathbf{Z})|P(\mathbf{E}, \mathbf{Z}|\mathbf{M}, \mathbf{R}, \phi) \\ &\propto \exp\{\mathbb{E}_{Q/q_i} \ln P(\mathbf{E}, \mathbf{Z}|\mathbf{M}, \mathbf{R}, \phi))\} \end{aligned}$$

For example, the update for a non-pronominal entity indicator component $r_j(\cdot)$ is given by: [6]

$$\begin{aligned} \ln r_j(z) &\propto \mathbb{E}_{Q/r_j} \ln P(\mathbf{E}, \mathbf{Z}, \mathbf{M}|\mathbf{R}, \phi) \\ &\propto \mathbb{E}_{q_z} \ln \left(P(z|\phi)P(M_j|E_z, R_z)\right) \\ &= \ln P(z|\phi) + \mathbb{E}_{q_z} \ln P(M_j|E_z, R_z) \end{aligned}$$

[5] See Liang and Klein (2007) for a primer on variational inference.

[6] For simplicity of exposition, we omit terms where M_j is an antecedent to a pronoun.

	Ment Acc.	Ent. Acc.
INDEP	60.0	43.7
JOINT	64.6	54.2
JOINT+PRO	**68.2**	**57.8**

Table 7.1. Results on corporate acquisition tasks with given role mention boundaries. We report mention role accuracy and entity role accuracy (correctly labeling all entity mentions).

A similar update is performed on pronominal entity indicator distributions, which we omit here for space. The update for variational entity distribution is given by:

$$\begin{aligned}
\ln q_i(e_i) &\propto \mathbb{E}_{Q/q_i} \ln P(\mathbf{E}, \mathbf{Z}, \mathbf{M}|\mathbf{R}, \phi) \\
&\propto \mathbb{E}_{\{r_j\}} \ln \left(P(e_i|R_i) \prod_{j:Z_j=i} P(M_j|e_i, R_i) \right) \\
&= \ln P(e_i|R_i) + \sum_j r_j(i) \ln P(M_j|e_i, R_i)
\end{aligned}$$

It is intractable to enumerate all possible entities e_i (each consisting of several sets of words). We instead limit the support of $q_i(e_i)$ to several sampled entities. We obtain entity samples by sampling mention entity indicators according to r_j. For a given sample, we assume that E_i consists of the non-pronominal head words and modifiers of mentions such that Z_j has sampled value i. We utilize 200 such samples.

During each E-Step we perform 5 iterations of updating each variational factor, which results in an approximate posterior distribution. Using expectations from this approximate posterior, our M-Step is relatively straightforward. The role parameters R_i are computed from the $q_i(e_i)$ and $r_j(z)$ distributions, and the global role prior ϕ from the non-pronominal components of $r_j(z)$.

We note that inference is far simpler in this model relative to Chapter 6 since we assume a fixed number of entities, one for each possible role.

7.4 Experiments

We present results on the corporate acquisitions task, which consists of 600 annotated documents split into a 300/300 train/test split. We use 50 training documents as a development set. In all documents, proper and (usually) nominal mentions are annotated with roles, while pronouns are not. We preprocess each document identically to Haghighi and Klein (2010): we sentence-segment using the OpenNLP toolkit, parse sentences with the Berkeley

	Role Id			Overall		
	P	R	F_1	P	R	F_1
INDEP	79.0	65.5	71.6	48.6	40.3	44.0
JOINT+PRO	**80.3**	69.2	74.3	53.4	46.4	49.7
BEST	80.1	**70.1**	**74.8**	**57.3**	**49.2**	**52.9**

Table 7.2. Results on corporate acquisitions data where mention boundaries are not provided. Systems must determine which mentions are template role-fillers as well as label them. Role Id only evaluates the binary decision of whether a mention is a template role-filler or not. overall includes correctly labeling mentions. Our best system, see Section 7.4, adds extra unannotated data to our joint+pro system.

Parser (Petrov et al., 2006), and extract mention properties from parse trees as well as from the Stanford Dependency Extractor (de Marneffe et al., 2006). [7]

7.4.1 Gold Role Boundaries

We first consider the simplified task where role mention boundaries are given. We map each labeled token span in training and test data to a parse tree node that shares the same head. In this setting, the role-filling task is a collective classification problem, since we know each mention is filling some role.

As our baseline, INDEP, we built a maximum entropy model which independently classifies each mention's role. It uses features as similar as possible to the generative model (and more), including the head word, typed dependencies of the head, various tree features, governing word, and several conjunctions of these features as well as coarser versions of lexicalized features. This system yields 60.0 mention labeling accuracy (see Table 7.1). The primary difficulty in classification is the disambiguation amongst the acquired, seller, and purchaser roles, which have similar internal structure, and differ primarily in their semantic contexts. The accuracy is 95.5 if the distinctions between these three roles are ignored. Our entity-centered model, JOINT in Table 7.1, has no latent variables at training time in this setting, since each role maps to a unique entity. This model yields 64.6, outperforming INDEP.[8]

During development, we noted that often the most direct evidence of the role of an entity was associated with pronoun usage (see the first `it` in Figure 7.1). Training our model with pronominal mentions, whose roles are latent variables at training time, improves accuracy to 68.2. While this approach incorrectly assumes that all pronouns have antecedents amongst our given mentions, this did not appear to degrade performance.

[7] We would've liked to have utilized the MUC-4 dataset, however this data does not have casing information, and our approach relies heavily on syntactic parsing.

[8] We use the mode of the variational posteriors $r_j(Z_j)$ to make predictions (see Section 7.3).

7.4.2 Full Task

We now consider the more difficult setting where role mention boundaries are not provided at test time. In this setting, we automatically extract mentions from a parse tree using a heuristic approach. Our mention extraction procedure yields 95% recall over annotated role mentions and 45% precision. Following Patwardhan and Riloff (2009), we match extracted mentions to labeled spans if the head of the mention matches the labeled span. Using extracted mentions as input, our task is to label some subset of the mentions with template roles. Since systems can label mentions as non-role bearing, only recall is critical to mention extraction. To adapt INDEP to this setting, we first use a binary classifier trained to distinguish role-bearing mentions. The baseline then classifies mentions which pass this first phase as before. Similar to Haghighi and Klein (2010), we add 'junk' roles to our model to flexibly model entities that do not correspond to annotated template roles. During training, extracted mentions which are not matched in the labeled data have posteriors which are constrained to be amongst the 'junk' roles.

We first evaluate role identification (ROLE ID in Table 7.2), the task of identifying mentions which play some role in the template. The binary classifier for INDEP yields 71.6 F_1. Our JOINT+PRO system yields 74.3. On the task of identifying and correctly labeling role mentions, our model outperforms INDEP as well (OVERALL in Table 7.2). As our model is generative, it is straightforward to utilize totally unannotated data. We added 700 fully unannotated documents from the mergers and acquisitions portion of the Reuters 21857 corpus. Training JOINT+PRO on this data as well as our original training data yields the best performance (BEST in Table 7.2).[9]

To our knowledge, the best results on this dataset are from Siefkes (2008), who report 45.9 weighted F_1, including the STATUS field we ignore. Since their performance on the STATUS field (56.3) exceeds this average, it is likely fair to compare their 45.9 F_1 (with STATUS) with our BEST system (without STATUS) evaluated in their slightly stricter way. Our BEST system yields 51.1.[10]

7.5 Conclusion

In this chapter, we have presented a joint generative model of reference resolution (from Chapter 6) and template-filling information extraction. This model makes role decisions at the entity, rather than at the mention level. This approach naturally incorporates information across multiple mentions, incorporates unannotated data, and yields strong performance. This chapter demonstrated that the entity reference resolution model developed in Chapter 6 can be used to improve the performance on an external downstream application.

[9] We scaled expected counts from the unlabeled data so that they did not overwhelm those from our (partially) labeled data.

[10] We deterministically select mention base NP tokens, excluding determiners, which almost always matches annotation. This is similar to the post-processing performed in Chapter 6.

Chapter 8

Conclusion

A persistent theme over the last fifteen years is the decentralization of information. More than ever, the information diet of the average consumer has substantially diversified; content about people, organizations, and events is not only spread across multiple discourses, but multiple sources (blogs, newswire, tweets, conversations, etc.). The task of disambiguating references to these objects will become increasingly essential to allow a user to discover or navigate large amounts of information. Models capable of 'multiplexing' this data in a largely unsupervised fashion will be of particular need.

This dissertation has presented advancements in entity reference resolution, which, hopefully, might benefit that end. One of its central contributions is that entity reference structure can be effectively modeled in an unsupervised fashion, allowing our approach to be deployed to novel domains and, with some alteration, other languages. Furthermore, we hoped we have demonstrated that entity-based approaches to such problems afford many potential benefits: ensure consistency amongst mention usages, a coherent way to share entity across documents, a mechanism for doing potential downstream applications (such as information extraction in Chapter 7). While much progress has been made in reference resolution, there remains many improvements to be made. There is still not a model which incorporated deeper semantic and coherency tendencies (such as those described in Hobbs (1979)), there is not a satisfactory account of nominals and their communicative intent, nor does this approach resolve references to events.

The generative models presented here can be used for a variety of NLP tasks where reasoning about entities or events is part of the task.[1] Another possible thread of future research is inducing a richer representation of entity properties. The model presented in Chapter 6 uses syntactic relations ,such the proper head, for semantic properties such as last names. It might be possible with limited supervision to incorporate richer semantic properties in this model. Beyond improving entity reference resolution performance, the entity properties induced by this model would be of independent use.

[1]Indeed, Bejan et al. (2009) present an event-coreference model based upon the work in Chapter 4.

Our model design has been inspired by a long history of keen linguistic insight and analysis as well as impressive advancements in structured machine learning techniques. There is a perceived conflict in NLP between work which exercises significant linguistic insight (typically in feature design or careful processing) and that which uses sophisticated machine learning. We hope that one thing we have demonstrated is that elements are not in conflict, and that modern machine learning techniques can facilitate the expression of linguistic wisdom.

Bibliography

David J. Aldous. Exchangeability and related topics. In *École d'été de probabilités de Saint-Flour, XIII—1983*, volume 1117 of *Lecture Notes in Math.*, pages 1–198. Springer, Berlin, 1985.

Amit Bagga and Breck Baldwin. Algorithms for Scoring Coreference Chains. In *Linguistic Coreference Workshop at The First International Conference on Language Resources and Evaluation (LREC)*, pages 563–566, 1998.

Cosmin Adrian Bejan, Matthew Titsworth, Andrew Hickl, and Sanda Harabagiu. Nonparametric Bayesian models for unsupervised event coreference resolution. In *Advances in Neural Information Processing Systems 22 (NIPS)*, 2009.

Eric Bengston and Dan Roth. Understanding the value of features for corefernce resolution. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2008.

Taylor Berg-Kirkpatrick, Alexandre Bouchard-Cote, John DeNero, and Dan Klein. Painless unsupervised learning with features. In *North American Chapter of the Association of Computational Linguistics (NAACL)*, 2010.

Shane Bergsma, Dekang Lin, Google Inc, and Randy Goebel. Distributional Identification of Nonreferential Pronouns. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 10–18, 2008.

Indrajit Bhattacharya and Lise Getoor. A Latent Dirichlet Model for Unsupervised Entity Resolution. In *Society of Industrial and Applied Mathematics (SIAM) conference on data mining*, 2006.

David Blei and Peter I. Frazier. Distance Dependent Chinese Restaurant Processes. In *International Conference on Machine Learning (ICML)*, 2009.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research (JMLR)*, 3:993–1022, 2003.

Claire Cardie and Kiri Wagstaff. Noun phrase coreference as clustering. In *Empirical Methods in Natural Language Processing (EMNLP)*, 1999.

Nathanael Chambers and Dan Jurafsky. Unsupervised Learning of Narrative Schemas and their Participants. In *Joint Conference of the 47th Annual Meeting of the ACL and the*

4th International Joint Conference on Natural Language Processing of the AFNLP, pages 602–610. Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics (ACL), 2009.

Eugene Charniak. Maximum Entropy Inspired Parser. In *North American Chapter of the Association of Computational Linguistics (NAACL)*, 2000.

Eugene Charniak and Micha Elsner. EM works for pronoun anaphora resolution. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL-09), Athens, Greece*, 2009.

Mike Collins. *Head-Driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania, 1999.

Aaron Culotta, Michael Wick, Robert Hall, and Andrew McCallum. First-order probabilistic models for coreference resolution. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*, 2007.

Hal Daume and Daniel Marcu. A Bayesian model for supervised clustering with the Dirichlet process prior. *Journal of Machine Learning Research (JMLR)*, 2005.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. Generating typed dependency parses from phrase structure trees. In *Linguistic Coreference Workshop at The First International Conference on Language Resources and Evaluation (LREC)*, 2006. URL `http://nlp.stanford.edu/pubs/LREC06_dependencies.pdf`.

Pascal Denis. *New Learning Models For Robust Reference Resolution*. PhD thesis, University of Texas at Austin, 2007.

Pascal Denis and Jason Baldridge. Global, joint determination of anaphoricity and coreference resolution using integer programming. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*, 2007.

Micha Elsner, Eugene Charniak, and Mark Johnson. Structured generative models for unsupervised named-entity clustering. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*, pages 164–172, 2009.

Christiane Fellbaum. *WordNet An Electronic Lexical Database*. The MIT Press, Cambridge, MA ; London, 1998. URL `http://mitpress.mit.edu/catalog/item/default.asp?ttype=2&tid=8106`.

Ivan P. Fellegi and Alan B. Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64:1183–1210, 1969.

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

Jenny Finkel and Christopher Manning. Enforcing transitivity in coreference resolution. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2008.

Jenny Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2005.

Jenny Finkel, Trond Grenager, and Christopher Manning. The infinite tree. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2007.

Dayne Freitag. *Machine Learning for Information Extraction in Informal Domains*. PhD thesis, Carnegie Mellon University (CMU), 1998.

Dayne Freitag and Andrew McCallum. Information extraction with hmm structures learned by stochastic estimation. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2000.

Niyu Ge, John Hale, and Eugene Charniak. A statistical approach to anaphora resolution. In *Sixth Workshop on Very Large Corpora*, 1998.

T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Suppl. 1):5228–5235, 2004.

Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. Centering: A framework for modelling the local coherence of discourse. *Computational Linguistics*, 21(2):203–225, 1995.

Aria Haghighi and Dan Klein. Prototype-driven learning for sequence models. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*. Association for Computational Linguistics, 2006.

Aria Haghighi and Dan Klein. Unsupervised coreference resolution in a nonparametric Bayesian model. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Association for Computational Linguistics, 2007.

Aria Haghighi and Dan Klein. Simple coreference resolution with rich syntactic and semantic features. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2009.

Aria Haghighi and Dan Klein. Coreference resolution in a modular, entity-centered model. In *North American Association of Computational Linguistics (NAACL)*, 2010.

Marti Hearst. Automatic acquisition of hyponyms from large text corpora. In *Conference on Natural Language Learning (COLING)*, 1992.

Jerry R. Hobbs. Resolving pronoun references. *Lingua*, 44:311–338, 1977.

Jerry R. Hobbs. Coherence and Coreference. *Cognitive Science*, 3:67–90, 1979.

Andrew Kehler, Laura Kertz, Hannah Rohde, and Jeffrey Elman. Coherence and coreference revisited. *Journal of Semantics (Special Issue on Processing Meaning)*, 2008.

Dan Klein and Chris Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, 2003.

Manfred Klenner and Etienne Ailloud. Optimization in coreference resolution is not needed: A nearly-optimal algorithm with intensional constraints. In *Recent Advances in Natural Language Processing*, 2007.

R. Langacker. On pronominalization and the chain of command. *Modern Studies of Language*, pages 160–186, 1969.

Shalom Lappin and Herbert J. Leass. An algorithm for Pronominal Anaphora Resolution. *Computational Linguistics*, 20(4):535–561, 1994.

R. Lees and E. Kilma. Rules for English pronominalization. *Language*, 39:17–28, 1963.

Percy Liang and Dan Klein. Structured Bayesian nonparametric models with variational inference (tutorial). In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2007.

Xiaoqiang Luo. On coreference resolution performance metrics. In *Human Language Technology and Empirical Methods in Natural Language Processing (HLT-EMNLP)*, 2005.

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

Andrew McCallum and Ben Wellner. Conditional models of identity uncertainty with application to noun coreference. In *Advances in Neural Information Processing Systems 17 (NIPS)*, 2005.

Joseph F. McCarthy. *A Trainable Approach To Coreference Resolution For Information Extraction*. PhD thesis, University of Massachusetts at Amherst, 1996.

Brian Milch, Bhaskara Marthi, Stuart Russell, David Sontag, Daniel L. Ong, and Andrey Kolobov. BLOG: Probabilistic models with unknown objects. In *Proceedings of the International Joint Conferences of Artificial Intelligence (IJCAI)*, 2005.

Christoph Muller. Automatic detection of non- referential it in spoken multi-party dialog. In *Proceedings of the Annual Meeting European Association of Computational Linguistics (EACL)*, pages 49–56, 2006.

Ani Nenkova. Entity-driven Rewrite for Multi-document Summarization. In *International Joint Conference on Natural Language Processing (IJCNLP)*, 2008.

Vincent Ng. Machine learning for coreference resolution: From local classification to global ranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, 2005.

Vincent Ng. Shallow semantics for coreference resolution. In *IJCAI'07: Proceedings of the 20th International Joint Conference on Artificial intelligence*, pages 1689–1694, 2007.

Vincent Ng. Unsupervised models for coreference resolution. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2008.

Vincent Ng and Claire Cardie. Improving Machine Learning Approaches to Coreference Resolution. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002.

NIST. The ACE evaluation plan. 2004. URL `http://www.itl.nist.gov/iad/mig//tests/ace/2004/doc/ace04-evalplan-v7.pdf`.

Hanna Pasula, Bhaskara Marthi, Brian Milch, Stuart Russell, and Ilya Shpitser. Identity Uncertainty and Citation Matching. In *Advances Neural Information Processing Systems 15 (NIPS)*, 2003.

Siddharth Patwardhan and Ellen Riloff. A unified model of phrasal and sentential evidence for information extraction. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2009.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 433–440, Sydney, Australia, July 2006. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P/P06/P06-1055`.

William Phillips and Ellen Riloff. Exploiting role-identifying nouns and expressions for information extraction. In *Recent Advances in Natural Language Processing (RANLP)*, 2007.

Jim Pitman. Combinatorial stochastic processes. In *Lecture Notes for St. Flour Summer School*, 2002.

Hoifung Poon and Pedro Domingos. Joint unsupervised coreference resolution with Markov Logic. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2008.

Altaf Rahman and Vincent Ng. Supervised models for coreference resolution. In *Proceedings of the 2009 Conference on Empirical Conference in Natural Language Processing*, 2009.

Deepak Ravichandran and Eduard Hovy. Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 41–47, 2002.

Matthew Richardson and Pedro Domingos. Markov Logic networks. *Machine Learning Journal (MLJ)*, 62:107–136, 2006.

Jayaram Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4: 639–650, 1994.

Candace L Sidner. Towards a computational theory of definite anaphora comprehension in english discourse. Technical report, MIT, Cambridge, MA, USA, 1979.

Christian Siefkes. *An Incrementally Trainable Statistical Approach to Information Extraction: Based on Token Classification and Rich Context Model*. VDM Verlag, Saarbrücken, Germany, Germany, 2008. ISBN 363900146X, 9783639001464.

Rion Snow, Dan Jurafsky, and Andrew Y. Ng. Learning syntactic patterns for automatic hypernym discovery. In *Advances in Neural Information Processing Systems 17 (NIPS)*, 2005.

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544, 2001.

Veselin Stoyanov, Nathan Gilbert, Claire Cardie, and Ellen Riloff. Conundrums in noun phrase coreference resolution: Making sense of the state-of-the-art. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2009.

Yee W. Teh, Michael Jordan, Matthew Beal, and David Blei. Hierarchical Dirichlet Processes. *Journal of the American Statistical Association*, 101, 2006.

Jose L. Vicedo and Antonio Ferrandez. Importance of pronominal anaphora resolution in question answering systems. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 555–562, 2000.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. A model-theoretic coreference scoring scheme. In *Message Understanding Conference 6 (MUC-6)*, 1995.

Michael Wick, Aaron Culotta, Khashayar Rohanimanesh, and Andrew McCallum. An Entity Based Model for Coreference Resolution. In *Society of Industrial and Applied Mathematics (SIAM)*, 2009.

Xiaofeng Yang, Jian Su, and Chew Lim Tan. Improving pronoun resolution using statistics-based semantic compatibility information. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2005.

Appendix A

Overview of Dirichlet Process and Extensions

This appendix presents a brief overview of the hierarchichal Dirichlet process (HDP) as well as the Dirichlet Process (DP) which are used in Chapters 4 and 6. Where appropriate, examples and terminology will be specific to the task of entity reference resolution.

A.1 Dirichlet Process

The **Dirichlet process** (DP) is a distribution over distributions. The DP is parametrized by an underlying base distribution G_0 as well as a scalar concentration parameter α. We use $G \sim DP(G_0, \alpha)$ to denote that G is a draw from the DP. While there are many equivalent descriptions of the DP, the one we initially use is the *stick-breaking construction* due to Sethuraman (1994).

Each draw G from a DP is a discrete distribution over a countably infinite number of samples $\eta_1, \ldots, \eta_n, \ldots$. Each sample, or *atom*, η_i is drawn from the base distribution G_0. We use $\boldsymbol{\eta}$ to denote this infinite sequence of samples. The form that G takes is given by,

$$G(\eta) = \sum_{i=1}^{\infty} \beta_i \mathbf{1}[\eta = \eta_i] \tag{A.1}$$

where the $\{\beta_i\}_{i=1}^{\infty}$ coefficients give the probability of selecting sample η_i. Each sample η_i in turn is drawn from the base distribution G_0. The $\{\beta_i\}$ coefficients form a valid distribution over the natural numbers: $\beta_i \geq 0$ and $\sum_{i=1}^{\infty} \beta_i = 1$. When one uses a DP, typically each η_i represents the parameters of a cluster. Thus β_i reflects the probability of selecting the ith cluster. Essentially, a draw from G represents drawing a cluster index (the β_i selected) and returning the parameters (η_i) associated with that cluster.

The $\{\beta_i\}_{i=1}^{\infty}$ probabilities are themselves random variables. The random process used to generate these mixture probabilities is where the stick-breaking construction gets its name. First, we draw a collection of intermediate random variables $\{\beta'_i\}_{i=1}^{\infty}$, where each

$\beta'_i \sim Beta(1, \alpha)$.[1] Then we set β_i to be,

$$\beta_i = \beta'_i \prod_{k=1}^{i-1}(1 - \beta'_k) \tag{A.2}$$

Intuitively, the probability of selecting the ith cluster is given by not selecting the first $i-1$ elements (this corresponds to the product in Equation A.2) and then selecting the $i$the component. Note that the α concentration parameter will tend to produce smaller β'_i random variables, which in turn yields β_i random variables.

We will use $\boldsymbol{\beta}$ to denote the countably infinite sequence $\{\beta_i\}$ and $\boldsymbol{\beta} \sim StickBreak(\alpha)$ denote the random process discussed above for computing the integer distribution encoded in $\boldsymbol{\beta}$. Similarly, we will use $\boldsymbol{\eta}$ to represent a countable infinite sequence of samples from the base distribution G_0. Since these two sequences characterize a DP draw (see Equation A.1), we will also denote a DP draw by $(\boldsymbol{\beta}, \boldsymbol{\eta}) \sim DP(\alpha, G_0)$.

A.1.1 Chinese Restaurant Process

The Chinese restaurant process (CRP) is a representation of the DP (Aldous, 1985) that is useful for performing Markov chain Monte Carlo (MCMC) inference. The CRP is a random process for clustering a collection of items, or, to keep with the metaphor, customers. The CRP has a single hyper-parameter, α, which, as we will see, is connected to the concentration parameter from Section A.1. For each customer, we assign the customer to either one of the current clusters (or tables) or have the customer start a new table. Let us denote the assignment of customer to a table by Z_i for the ith customer. We use $Z_{1:n}$ to denote the sequence of customer table assignments, where $Z_{1:n}$ is shorthand for the sequence $(Z_1, \ldots, Z_n)$. We let $K = \max_{i=1,\ldots,n} Z_i$ denote the current (random) number of existing tables. Consider a new customer. The CRP distribution over the next customer seat assignment, Z_{n+1}, is given by,

$$P(Z_{n+1} = k | Z_{1:n}, \alpha) \propto \begin{cases} n_k & \text{where } k \leq K \\ \alpha & \text{where } k = K + 1 \end{cases} \tag{A.3}$$

where n_k is the number of $Z_1, \ldots, Z_n$ which have the value k. Essentially, when assigning a cluster to a new datum, the chance of joining a cluster is proportional to the number of datums in the cluster as well as some remaining mass (proportional to α) for a customer to join a new table $(K + 1)$.

The CRP is related to the DP since the posterior in Equation A.3 represents cluster assignment in the DP where the $\boldsymbol{\beta}$ parameters have been integrated out. Concretely, suppose that $\boldsymbol{\beta} \sim StickBreak(\alpha)$ and for each customer, we draw an assignment $Z \sim Multinomial(\boldsymbol{\beta})$. If the value of Z is $\leq K$, we assign the customer to that table. Otherwise, we assign the customer to a new $K + 1$ table. Then the posterior probability

[1] The Beta distribution is a prior over binomial probabilities. The expected value from $Beta(1, \alpha)$ is $\frac{1}{1+\alpha}$.

of $P(Z_{n+1}|Z_{1:n}, \alpha)$ is also given by Equation A.3.[2] Note that in this expression, we have integrated out the unobserved $\boldsymbol{\beta}$ parameters.

A.1.2 Infinite Mixture Model

The CRP is useful for models which use the DP as an infinite mixture (clustering) model as we do in Section 4.3.2. In that setting, each cluster corresponds to an entity. Each entity is associated with parameters ϕ^h, a multinomial distribution over possible mention heads. Our goal with this model is to obtain samples from the posterior $P(\mathbf{Z}|\mathbf{H})$, where $\mathbf{Z}$ denote the entity assignments $Z_{1:n}$ for the n mentions and $\mathbf{H}$ denotes the n mention heads $H_{1:n}$. We use ϕ^h_i to denote the parameters for entity E_i. Formally, the infinite mixture model is described by,

Draw $(\boldsymbol{\beta}, \boldsymbol{\phi}^h) \sim DP(\alpha, G_0)$

For each mention head $i = 1, \ldots, n$,

Draw $Z_i|\boldsymbol{\beta} \sim Multinomial(\boldsymbol{\beta})$

$H_i|Z_i \sim Multinomial(\phi^h_{Z_i})$

The G_0 base distribution is responsible for generating new entity parameters ϕ^h. In this model, we take G_0 to be DIRICHLET(λ_H, V) as discussed in Section 4.3.2;[3] also V is the size of the head vocabulary. Our goal during inference is to obtain samples from $P(\mathbf{Z}|\mathbf{H}, \alpha, \lambda_H)$. We use Gibbs sampling to accomplish this and sample each Z_i random variable, treating the rest as fixed. Since, in this model, the H_i data are exchangeable,[4] we can pretend that the head we are sampling comes last. Without loss of generality, we can assume we are always sampling the entity indicator Z_{n+1} for the last head H_{n+1}. The posterior over Z_{n+1} is given by,

$$P(Z_{n+1}|H_{1:n+1}, Z_{1:n}, \alpha, \lambda_H) \propto P(Z_{n+1}|Z_{1:n}, \alpha)P(H_{n+1}|Z_{1:n+1}, H_{1:n}, \lambda_H) \tag{A.4}$$

The first term on the right hand side can be computed using the CRP representation from Equation A.3. The head probability can be computed by integrating over the entity parameters for Z_{n+1},

$$P(H_{n+1}|Z_{1:n+1}, H_{1:n}, \lambda_H) = \int P(\phi^h|Z_{1:n+1}, H_{1:n}, \lambda_H)P(H_{n+1}|\phi^h)d\phi^h \tag{A.5}$$

This decomposition holds since all heads (H_i) are independent given the head parameters ϕ^h. Since $P(H_{n+1}|\phi^h)$ is multinomial and $P(\phi^h)$ is Dirichlet, we can exploit conjugacy, which yields

$$P(H_{n+1} = w|Z_{1:n+1}, H_{1:n}, \lambda_H) \propto n_w + \lambda_H \tag{A.6}$$

[2]Note that there is implicitly a permutation over cluster indices being used to ensure that cluster indices appear in increasing order without any 'gaps'.

[3]The λ_H, as described in Section 4.3.2, is given by e^{-4}.

[4]Formally exchangeability means the probability of $H_{1:n}$ is the same for any permutation.

where $n_w = |\{i : Z_i = Z_{n+1}, H_i = w\}|$ is the number of heads given the same entity as Z_{n+1} with head w.

By repeatedly sampling cluster assignments, we are guaranteed that we will be sampling from the true posterior over entity assignments.

A.2 Distance-Dependent Chinese Restaurant Process

In Chapter 6, the discourse module used a variant of the Chinese restaurant process (CRP) presented in Section A.1.1. This variant is called the Distance-dependent CRP and originates from Blei and Frazier (2009). Recalling the setup from Section A.1.1: Suppose that there are n customers and we have assigned each a cluster index Z_i. We use $Z_{1:n}$ denote the sequence of existing cluster assignments. Consider the cluster assignment Z^* to the $n+1$th customer. We also assume that we have a set of distances $\{d_{i,j}\}$ between customer i and customer j.[5] Before selecting a cluster assignment Z_{n+1}, the distance-dependent CRP (DD-CRP) first selects an antecedent A_{n+1}, which takes values between $\{1, \ldots, n, n+1\}$. The antecedent A_{n+1} is selected according to the customer distances,[6]

$$P(A_{n+1} = j|\alpha) \propto \begin{cases} \exp\{-d_{n+1,j}\}, & \text{if } j < n+1 \\ \alpha, & \text{if } j = n+1 \end{cases} \tag{A.7}$$

Once the antecedent has been chosen, the cluster assignment is selected according to,

$$Z_{n+1} = \begin{cases} Z_{A_{n+1}}, & \text{if } A_{n+1} < n+1 \\ K+1, & \text{otherwise} \end{cases} \tag{A.8}$$

The cluster assignment is stolen from the antecedent, or if there isn't an antecedent, the customer spawns a new cluster. Note that the DD-CRP reduces to the standard CRP if all the distances are $d_{i,j} = 0$.[7]

A.2.1 Relationship to Discourse Module in Section 6.3.2

The discourse module described in Section 6.3.2 utilizes the DD-CRP in a particular way. Rather than directly utilize customer distances, we instead parametrize the 'affinities' $\exp\{-d_{i,j}\}$ using a standard log-linear probability model. We also parametrize the choice of starting a new cluster, which diverges from Blei and Frazier (2009)'s presentation. To our knowledge, there is no existing work which learns parameters for the DD-CRP distances, but it is a natural extension.

[5]These distances don't necessarily come from a distance metric, but only need to satisfy non-negativity and a relaxed version of the triangle inequality: $d_{i,j} = 0$ and $d_{j,k}$ imply $d_{i,k} = 0$.

[6]In Blei and Frazier (2009), he uses an arbitrary function to decay distances. This generality isn't necessary here, so we do not utilize it.

[7]This would yield a uniform choice amongst prior antecedents; thus the probability of selecting a particular cluster is proportional to the number of elements currently in the cluster, matching Equation A.3.

A.3 Hierarchichal Dirichlet Process

The hierarchichal Dirichlet process (HDP), described more fully in Teh et al. (2006), is a straightforward extension of the DP (Section A.1). This extension is used in the context of our cross-document entity reference resolution model (See Section 4.3.5). At a high level, the DP is used to model clustering datasets where each datum is generated from an underlying cluster component. The HDP is intended to be used when you have multiple clustering datasets and the cluster components may be shared between different datasets. For our application of entity reference resolution, the utility of sharing an entity (a clustering component) across datums (mentions) in different datasets (documents).

The technical means by which this is achieved is relatively straightforward. Recall the infinite mixture model described in Section A.1.2. In that model, we make a single draw $(\boldsymbol{\beta}, \boldsymbol{\eta})$ from $DP(\alpha, G_0)$ which is used to generate data. Suppose we have m different clustering datasets and for each dataset we draw independent $G^{(1)}, \ldots, G^{(m)}$ all from the same underlying distribution $DP(\alpha, G_0)$. What it means to 'share' a clustering component between these is that some of cluster component parameters η has mass amongst multiple $G^{(i)}$ distributions. Since for each $G^{(i)}$, the $\boldsymbol{\eta}$ parameters are sampled from G_0, it suffices that G_0 have a non-zero probability of generating the same atom η multiple times. In general, for real-valued distributions, this is not the case.[8] However, if the underlying G_0 is itself a DP draw, then the probability of drawing the same atom multiple times is non-zero; in fact, depending on the setting of the concentration parameter, it can be made very likely. The HDP model thus draws a global distribution $G_0 \sim DP(\alpha_0, H)$ using global concentration parameter α_0 and base distribution H. Then for each clustering dataset, a DP draw is made from $DP(\alpha, G_0)$ as in Section A.1.2.

[8] For instance, the uniform distribution from $[0, 1]$ has probability zero of ever selecting the same value twice.

Appendix B

Learning and Inference Details for Chapter 4 Model

This appendix presents learning and inference details for the model presented in Chapter 4. Specifically, we provide details for the models in Section 4.3.3 and 4.3.5. Learning for the simpler models from Chapter 4 is fully explained in those sections.

B.1 Details for Pronoun Head Model

This model enriches the representation for the parameters, ϕ, of an entity. Each entity draws multinomial distributions: ϕ^h (non-pronominal head distribution), ϕ^t (type distribution), ϕ^g (gender distribution), and ϕ^n (number distribution). The head distribution ϕ^h is drawn from a symmetric Dirichlet with concentration parameter λ_H.[1] The feature distributions (ϕ^t, ϕ^g, ϕ^t) are drawn from symmetric Dirichlet distributions with the appropriate sizes (4,3,2 respectively) with a shared concentration parameter. We use $\phi = (\phi^h, \phi^t, \phi^g, \phi^t)$ to collectively denote the parameters for an entity.

We now describe the mention generation module more fully. Each mention $M = (P, H, T, G, N)$ consists of a mention type (P), head (H), entity type (T), gender (G), and number (N). In addition to entity parameters ϕ, there are also pronoun parameters $\boldsymbol{\theta}$. Assuming that ϕ are the entity parameters for a given mention, mention generation decomposes as,

$$P(H, T, G, N|\phi, \boldsymbol{\theta}) = P(T|\phi^t)P(G|\phi^g)P(N|\phi^n) \begin{cases} P(H|\phi^h) & \text{if not pronominal} \\ P(H|T, G, N, \boldsymbol{\theta}) & \text{otherwise} \end{cases} \tag{B.1}$$

Essentially, the features (T, G, N) are generated from underlying entity parameters. Then, generating the mention head depends on whether the mention is pronominal or not. If it is not pronominal, it comes from the entity-specific head distribution ϕ^h, otherwise, a pronoun head

[1] As mentioned in Section 4.3.1, all Dirichlet hyper-parameters are e^{-4} unless stated otherwise.

is generated from the pronoun parameters $\boldsymbol{\theta}$ conditioned on features (T, G, N). In performing model inference, rather than sampling (T, G, N), we integrate out these random variables. Because of this, we can no longer analytically integrate out the (ϕ^t, ϕ^g, ϕ^t) parameters nor the pronoun parameters $\boldsymbol{\theta}$. We instead opt to estimate these parameters. Specifically, for each entity we have estimates for $(\hat{\phi}^t, \hat{\phi}^g, \hat{\phi}^n)$ as well as for the global pronoun parameters $\hat{\boldsymbol{\theta}}$. We use $\hat{\boldsymbol{\phi}}$ to denote the collection of estimated entity feature parameters.[2]

We now describe how this affects sampling the entity indicator Z for a given mention M. We will of course condition on other entity indicator samples $Z_{1:n}$ and mention heads $H_{1:n}$. If the mention is non-pronominal, the (T, G, N) are independent of the head H and can be effectively ignored. In this case, sampling is identical as in Section 4.3.2. If the mention is pronominal on the other hand, we must sum over the latent (T, G, N) variables. Of course not all (T, G, N) tuples are legal settings; for instance, a non-person ($T \neq$ PERSON) must have a neuter gender ($G =$ NEUTER). We use $\mathcal{V}$ to denote these valid tuples. In English the valid tuple constraints are:

- If G is not NEUTER then N cannot be plural
- If T is not PERSON then N must be NEUTER
- If T is PERSON then N cannot be NEUTER

The posterior probability of assigning a given pronoun mention to entity z ($Z = z$) is given by:

$$P(Z = z|Z_{1:n}, H_{1:n}, \alpha, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\phi}}) \propto P(Z = z|Z_{1:n}, \alpha)P(H|Z = z, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\phi}}) \tag{B.2}$$

$$= P(Z = z|Z_{1:n}, \alpha) \tag{B.3}$$

$$\left(\sum_{(T,G,N)\in\mathcal{V}} P(T|\hat{\phi}^t_z)P(G|\hat{\phi}^g_z)P(N|\hat{\phi}^n_z)P(H|T, G, N, \hat{\boldsymbol{\theta}}) \right) \tag{B.4}$$

The first term on the right hand side can be computed as before using Equation A.3. Each of the terms in the summation in the second product is simply a 'lookup' in the appropriate parameters. Of course, some pronouns have observed values amongst the (T, G, N) variables; for instance, we know all values for the pronoun `he` (PERSON,MALE,SINGLE). In such cases, we limit the summation over tuples consistent with the observed pronoun feature values.

At the end of each sampling round, we re-estimate the $(\hat{\phi}^t, \hat{\phi}^g, \hat{\phi}^n)$ parameters associated with each entity as well as the global pronoun parameters $\hat{\boldsymbol{\theta}}$. Note that the only statistics relevant to these distributions are the (T, G, N) counts associated with each mention usage. The $\boldsymbol{\theta}$ parameters depend on the counts of these triples and the each entity's $(\hat{\phi}^t, \hat{\phi}^g, \hat{\phi}^n)$ estimates depend only on the appropriate T,G, or N count. We compute expected counts over (T, G, N), denoted $C(T, G, N)$, for each pronoun mention as follows:

$$C(T, G, N) \propto P(T|\hat{\phi}^t)P(G|\hat{\phi}^g)P(N|\hat{\phi}^n)P(H|T, G, N, \hat{\boldsymbol{\theta}}) \tag{B.5}$$

[2] ϕ^h is not included in $\hat{\phi}$ since these parameters can still be analytically integrated out and do not require estimation.

This computation uses current parameter estimates. The counts $C(T, G, N)$ are normalized over valid tuples for each pronoun mention. These counts are used to re-estimate parameters for the next round of sampling entity assignments.

This represents what was done in Haghighi and Klein (2007). It is of course possible to instead view this inference scheme as a variational approximation, where point-mass estimates are made for all parameters. Without much difficulty, we can amend this approach to accommodate full variational estimates, yielding a mixed MCMC and variational approach. In our experiments, this gave no significant difference.

B.2 Cross Document Model

When developing the cross-document model in Section 4.3.5, we utilize the direct sampler described in Teh et al. (2006). At a high level in this sampler, the CRP representation (see Section A.1.1) is used at the document level and the stick-breaking representation (Section A.1) is used at the global level.

In this approach, we directly sample the global entity index for each entity assignment. In order to do this, while we marginalize out CRP parameters at the document level, the β random variables in Figure 4.5, we instead sample the global entity mixture probabilities, β_0 random variables in Figure 4.5. In describing this model in Section 4.3.5, we left details out about how to sample the global entity distribution β_0 given entity assignments at the document level.[3]

There is a subtle issue in sampling β_0 from document entity counts. Suppose at the document level we have two mentions assigned to a given entity. We do not know from this information alone, whether both of those counts originated from the document DP drawing from the parent global distribution, or whether one came from the global distribution and the other from the 'reuse' of the entity at the document level. Given only the document entity counts, we do not maintain this information and must sample it. Suppose we observe m counts of an entity within a document. We can sample the number of times the use of the entity was the result of a parent draw. We can sample this quantity, denoted m', as follows:

$m' \leftarrow 1$
for $i = 2$ to m **do**
  if CoinFlip($\frac{\alpha_0}{\alpha_0 + i - 1}$) **then**
    $m' \leftarrow m' + 1$
  end if
  `return` m'
end for

These entity counts m' are aggregated for all entities across all documents. We use $m_1, \ldots, m_K$ to denote the summed counts for all K instantiated entities (K is of course

[3]This sampling scheme is also present in Finkel et al. (2007) as well.

random but determined by entity assignments). Then we sample

$$(\beta_0^0, \ldots, \beta_0^1, \beta_0^u) \sim \textsc{Dirichlet}(m_1, \ldots, m_K, \alpha_0) \tag{B.6}$$

where β_0^i is the global entity probability for entity i and β_0^u is the mass left in the global distribution for unobserved entities. These probabilities are used in the direct sampler equations in Equation 4.7.

Appendix C

Learning and Inference Details for Chapter 6 Model

This appendix presents inference details from Section 6.4, repeating some content as necessary. Our learning procedure involves finding parameters and assignments which are likely under our model's posterior distribution $P(\mathbf{E}, \mathbf{Z}, \boldsymbol{\tau}, \boldsymbol{\pi} | \mathbf{M}, \mathbf{X})$. We approximate this posterior using variational inference.

Our variational approximation decomposes model variables to facilitate learning. We split each entity E into a type T and a collection of lists L. Each entity L consists of a collection of lists L_r for each property. We use $\mathbf{T}$ for the collection of types and $\mathbf{L}$ for the collection of all entity word lists. Note that there at most n entities in a document since there are that many mentions. the entity variables. We decompose the mentions $\mathbf{M}$ into *referring* mentions (propers and nominals), $\mathbf{M}^r$, and pronominal mentions, $\mathbf{M}^p$ (with sizes n_r and n_p respectively). The entity assignments $\mathbf{Z}$ are similarly divided into $\mathbf{Z}^r$ and $\mathbf{Z}^p$ components (with sizes n_r and n_p respectively). For pronouns, rather than use $\mathbf{Z}^p$, we instead work with the corresponding antecedent variables, denoted $\mathbf{A}^p$, and marginalize over antecedents to obtain $\mathbf{Z}^p$.

Given these variable decomposition, our variational approximation takes the form,

$$P(\mathbf{E}, \mathbf{Z}, \boldsymbol{\tau}, \boldsymbol{\pi} | \mathbf{M}, \mathbf{X}) \approx Q(\mathbf{E}, \mathbf{Z}, \boldsymbol{\tau}, \boldsymbol{\pi}) \tag{C.1}$$

$$= \delta_r(\mathbf{Z}^r, \mathbf{L}) \delta_s(\boldsymbol{\tau}) \delta_d(\boldsymbol{\pi}) \tag{C.2}$$

$$\left(\prod_{k=1}^{n} q_k(T_k) \right) \left(\prod_{i=1}^{n_p} r_i(A_i^p) \right) \tag{C.3}$$

The δ_r, δ_s, and δ_d factors place point estimates on a single value, just as in hard EM. The $\{q_i\}$ and $\{r_j\}$ are full variational factor distributions over the appropriate random variable. Mean-field inference optimizes each factor in turn, keeping the others fixed. Each update to a factor q takes the form:

$$q \leftarrow \arg\min_{q} KL(Q(\mathbf{E}, \mathbf{Z}, \boldsymbol{\tau}, \boldsymbol{\pi}) | P(\mathbf{E}, \mathbf{Z}, \boldsymbol{\tau}, \boldsymbol{\pi} | \mathbf{M}, \mathbf{X})) \tag{C.4}$$

In the case of a mean-field decomposition, this update takes the form:

$$q(x) \propto \exp\left\{\mathbb{E}_{Q/q} \ln P(\mathbf{E}, \mathbf{Z}, \boldsymbol{\tau}, \boldsymbol{\pi}, \mathbf{M}, x|\mathbf{X})\right\} \tag{C.5}$$

where Q/q represents the variational approximation factors except the currently updated q. In essence, the new variational probability of the $q(x)$ is proportional to plugging x into the log-likelihood of the model and computing the expected log-likelihood given the other variational factor estimates. For a point-mass factor, the update must just find the setting of x which maximizes the right hand side above. The above updates are simplified in practice since any term in the model likelihood which do not depend on the updated variables can be safely ignored. The only model factors that are relevant are those in the markov blanket of the random variable.

We now fully provide more details regarding each factor update:

Updating type parameters $\delta_s(\boldsymbol{\tau})$: The type parameters $\boldsymbol{\tau}$ consist of several multinomial distributions which can be updated by normalizing expected counts as in the EM algorithm. The prior $P(\boldsymbol{\tau}|\boldsymbol{\lambda})$ consists of several finite Dirichlet draws for each multinomial, which are incorporated as pseudocounts. The relevant distribution hyper-parameters are: The θ_r parameters are drawn from a symmetric Dirichlet distribution with a concentration of 1. The f_r distribution is drawn from a 21-dimensional asymmetric Dirichlet with concentrations set to $10^{-0.5k}$, where $k = 0, \ldots, 20$ is the list length. This initially encourages all lists to be short. The ϕ distribution is drawn from a symmetric Dirichlet with concentration 10 to encourage the type prior not to be too skewed.

We fully describe how expected counts are obtained for parameters (θ_r, f_r) for a fixed property r (say NAM-HEAD) and for a fixed type T. Suppose we have the $\delta_r(\mathbf{Z}^r, \mathbf{L})$ estimate for all entities. This factor encodes concrete referring mention entity assignments and the (deterministically obtained word lists $\mathbf{L}$, see below for more details). For each entity list $L^{(i)} \in \mathbf{L}$ and for each word $w \in L_r^{(i)}$, we accrue a $q_i(T)$ count of that word to reflect the uncertainty over the entity type. Similarly, for each list $L_r^{(i)}$ we accrue an observation of the list length $|L_r^{(i)}|$ towards the expected sufficient statistics for the f_r distribution. For each referring mention, we look at the (possibly empty) set of word(s) associated with property r. Then for the entity list L_r associated with that entity and property, we compute the expected count,

$$\frac{(\alpha_r)\theta_r}{\textsc{Uniform}(L_r) + (\alpha_r)\theta_r}$$

and accrue that towards the expected counts for the new θ_r estimate. This count reflects the expected count of whether given word usage came from the entity list or from the type unigram distribution. Note that if the property r has $\alpha_r = 0$, we do not bother with this step. Likewise, when $\alpha_r = 1$, as is the case for pronoun and governor properties, we increment the appropriate θ_r sufficient statistics by $q_i(T)$.

In the particular case of a pronoun mention m_j^p. We loop over potential antecedents A_j^p and potential types T and increment the appropriate θ_r distributions by $r_j(A_j^p) q_{Z^r_{A_j^p}}(T)$.

Updating discourse parameters $\delta_d(\boldsymbol{\pi})$: The learned parameters for the discourse module rely on pairwise antecedent counts for assignments to nominal and pronominal mentions.[1] Given these expected counts, which can be easily obtained from other factors, the update reduces to a weighted maximum entropy problem, which we optimize using LBFGS. This problem is very similar to the sort described by Berg-Kirkpatrick et al. (2010).

For a single pronoun, we present the form this problem takes. Consider a pronoun. We have a current estimate $r_j(A^p)$ over the potential antecedents. We treat the variational posterior over A_j^p as fractional observed counts. We use $\hat{c}(A_j^p)$ to denote these 'empirical' counts. As a model, our discourse component, for any setting of the parameters $\boldsymbol{\pi}$ has a model distribution over A_j^p given by,

$$P_{\boldsymbol{\pi}}(A_j^p = i) \propto \exp\{\boldsymbol{\pi}^T f(i,j)\} \tag{C.6}$$

for antecedent positions i.[2] Then we select $\boldsymbol{\pi}$ to maximize the model log-likelihood of our observed counts:

$$\arg\max_{\boldsymbol{\pi}} \sum_i \hat{c}(i) \log P_{\boldsymbol{\pi}}(A_j^p = i) \tag{C.7}$$

Of course when we update $\boldsymbol{\pi}$, we collect counts from all antecedent choices from all pronominal and nominal mentions. For pronominal mentions, we take such expected counts directly from $r_i(A_i^p)$. For nominals mentions, which may only have proper and nominal antecedents, we compute $Q_{-\delta_d}(A_i^r)$, which involves normalizing the antecedent prior over antecedent mentions assigned to the same entity as the current mention.

We also assume that $P(\boldsymbol{\pi}|\sigma^2)$ is a zero-centered normal distribution with shared diagonal variance σ^2, which is incorporated via L2 regularization during optimization of the above criterion. We set $\sigma^2 = 0.5$ for all experiments.

Updating $\delta_r(\mathbf{Z}^r, \mathbf{L})$: We utilize a sequential beam search to find an approximation to the optimal referring mention entity assignments as well as the entity word lists of each instantiated entity. At the ith step of the search, we maintain a set of hypotheses over $\mathbf{Z}^r_{<i} = (Z_1^r, \ldots, Z_{i-1}^r)$. We associate a $\mathbf{Z}^r_{<i}$ hypothesis, with the optimal entity lists, $\mathbf{L}$, given the hypothesis entity assignments. Essentially, given entity assignments to mentions, we can compute the optimal word list for each attribute based on attribute word frequencies in assigned mentions and current parameters $\boldsymbol{\tau}$. This depends on the fact that all attributes associated with pronoun mentions have α_r set to 1 (Section 6.3.3), and thus have no entity word lists. We also marginalize over antecedent random variables A_i. We extend this hypothesis to one over $\mathbf{Z}^r_{<i+1}$, by adding M_i^r to an existing entity in $\mathbf{Z}^r_{<i}$ or by starting a new one. We can compute the score of these hypotheses efficiently by only computing the delta score associated with incorporating a new mention into an existing hypothesis. For our experiments, we limited the number of hypotheses on the beam to the top fifty and did not notice an improvement in model score from increasing beam size. Many of these

[1] Propers have no learned discourse parameters.

[2] Recall pronouns are not allowed to start their own cluster and can only select from mention positions and the last two sentences.

computations can done efficiently by caching and sharing computations amongst hypotheses with similar partition structure.

Updating pronominal antecedents $r_i(A_i^p)$ and entity types $q_k(T_k)$: These updates are straightforward instantiations of the mean-field update (C.4). The pronoun antecedent update is given by,

$$\ln r_i(A_i^p = j) \propto \ln P(A_i^p = j|\boldsymbol{\pi}) + \sum_{k=1}^{n} Q_{-r_i}(Z_j = k)\mathbb{E}_{q_k} \log P(M_i|E_k)$$

The $Q_{-r_i}(Z_j = k)$ term is the probability of the potential antecedent M_j being assigned to entity k. If M_j is a referring mention then $Q_{-r_i}(Z_j = k) = \mathbf{1}[Z_j^r = k]$, using the factor $\delta(\mathbf{Z}^r, \mathbf{L})$; there is only one entity that has mass for this mention in the variational posterior approximation. If the antecedent is a pronoun, we can recursively compute the entity assignment distribution by looking at this antecedent's antecedent. Since pronouns are not allowed to initiate a pronoun, this process terminates.

The type factor is applied to each of the entities and is given by,

$$\ln q_k(T_k = t) \propto \log P(T_k = t|\phi) + \log P(L^{(k)}|T_k = t, \boldsymbol{\tau}) \sum_{i=1}^{n} Q_{-q_k}(Z_i = k) \log P(M_i|L^{(k)}, T_k = t, \tau_t)$$

The summation term loops over all mentions and weighs the variation probability of that mention being assigned the current entity. The $\log P(M_i|L^{(k)}, T_k = t, \tau_t)$ term utilizes the current estimate $L^{(k)}$ of the entity lists associated with the entity.

Updates are performed until convergence. To produce our final coreference partitions, we assign each referring mention to the entity given by the δ_r factor and each pronoun to the most likely entity given by the r_i.