Title
Identifying and Resolving Entities in Text

Author
Durrett, Gregory Christopher

Publication Date
2016

Peer reviewed|Thesis/dissertation

Identifying and Resolving Entities in Text

by

Gregory Christopher Durrett

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 John DeNero
Professor David Bamman

Summer 2016

Identifying and Resolving Entities in Text

Abstract

Identifying and Resolving Entities in Text

by

Gregory Christopher Durrett

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Dan Klein, Chair

When automated systems attempt to deal with unstructured text, a key subproblem is identifying the relevant actors in that text—answering the "who" of the narrative being presented. This thesis is concerned with developing tools to solve this NLP subproblem, which we call entity analysis. We focus on two tasks in particular: first, coreference resolution, which consists of within-document identification of entities, and second, entity linking, which involves identifying each of those entities with an entry in a knowledge base like Wikipedia.

One of the challenges of coreference is that it requires dealing with many different linguistic phenomenon: constraints in reference resolution arise from syntax, semantics, discourse, and pragmatics. This diversity of effects to handle makes it difficult to build effective learning-based coreference resolution systems rather than relying on handcrafted features. We show that a set of simple features inspecting surface lexical properties of a document is sufficient to capture a range of these effects, and that these can power an efficient, high-performing coreference system.

Our analysis of our base coreference system shows that some examples can only be resolved successfully by exploiting world knowledge or deeper knowledge of semantics. Therefore, we turn to the task of entity linking and tackle it not in isolation, but instead jointly with coreference. By doing so, our coreference module can draw upon knowledge from a resource like Wikipedia, and our entity linking module can draw on information from multiple mentions of the entity we are attempting to resolve. Our joint model of these tasks, which additionally models semantic types of entities, gives strong performance across the board and shows that effectively exploiting these interactions is a natural way to build better NLP systems.

Having developed these tools, we show that they can be useful for a downstream NLP task, namely automatic summarization. We develop an extractive and compressive automatic summarization system, and argue that one deficiency it has is its inability to use pronouns coherently in generated summaries, as we may have deleted content that contained a pronoun's antecedent. Our entity analysis machinery allows us to place constraints on summarization that guarantee pronoun interpretability: each pronoun must have a valid

antecedent included in the summary or it must be expanded into a reference that makes sense in isolation. We see improvements in our system's ability to produce summaries with coherent pronouns, which suggests that deeper integration of various parts of the NLP stack promises to yield better systems for text understanding.

To my family and chosen family

Contents

List of Figures

List of Tables

Acknowledgments

Looking back on the process of writing this thesis, I marvel at the fact that it's been six years since I started at UC Berkeley; it certainly doesn't feel that way. Part of what has made it fly by is the joy of being around incredible colleagues, friends, and family, who I'll now proceed to thank as thoroughly as I can.

My advisor would say that being at Berkeley isn't so much about working with him as it is about working with the other members of his research group, so I'll start by thanking them. Jonathan Kummerfeld: as classmate, groupmate, and housemate, you've been there every step of the way through this process and changed it so much for the better that I can't really put it into words. Taylor Berg-Kirkpatrick: through our collaborations, I've striven to acquire your fearless command of machine learning and your attitude that anything is possible. David Hall: thanks for inspiring me to try to do things the right way, as well as for all your help with my Scala frustrations. Jacob Andreas: all of our discussions of papers and research have been every bit as valuable as doing the actual research itself. Thanks also to Adam Pauls, who helped me find my feet on my first (real) NLP research paper, and John DeNero, who has mentored me both at Google and Berkeley. Finally, thanks to the rest of the Berkeley NLP group: Mohit Bansal, Percy Liang, Dave Golland, David Burkett, Maxim Rabinovich, Daniel Fried, Mitchell Stern, and Nikita Kitaev. The real work happened in all those conversations that distracted our computer vision colleagues.

Of course, I would be remiss if I did not also single out Dan Klein for being a true inspiration. His dedication to teaching, his confidence in research and disregard for the impossible, his telling me that things would be fine when I was freaking out my second year (he ended up being right), and his uncanny ability to refine and present ideas all make him an amazing role model and teacher. Above all, I always felt that he put my interests before his and was there to help me however I needed, and having your advisor look out for you in such a deep way is a pretty special thing. So I'd like to thank Dan Klein for having me as a student, and as I pursue my academic career, my only hope is that I've osmosed some small fraction of his ability and passion.

On the subject of advising, I would like to thank my two fantastic undergraduates, James Ferguson and Matthew-Francis Landau, for putting up with my haphazard explanations of things and my fumbling attempts to advise them. I'm excited to see what they do in their graduate careers and to run into them at conferences in the future.

I'd also like to thank Regina Barzilay, whose NLP class at MIT first piqued my interest in the subject, as well as Dan Roth, Luke Zettlemoyer, and all the other NLP faculty who have been so friendly and supportive the last several years.

On a different note, being at Berkeley also presented me with the opportunity to play in a fantastic orchestra. I thank David Milnes for taking a chance on me after my lackluster audition my first year, and letting me redeem myself by making at least a bit of music out of all those E♭ clarinet parts over the years. And I'd like to thank my fellow musicians in the orchestra: Lauren Washburn, Cameron Winrow, Lucian Pixley, Mary-Anne Kidwell, Bryson Cwick, Andrea Mich, Eric Price, Chris Wirick, and roughly 100 other people I don't have

space to name, who have all challenged me to be a better musician and helped me feel like a part of something unforgettable.

I also thank my friends and fellow graduate students of 1044 Keith Avenue. Jonathan Kummerfeld, Jon Long, Justine Sherry, Jonathan Kohler, Edgar Solomonik, Michael Cole, and Mollie Schwartz are the best people I could've surrounded myself with for the past six years. And of course, this goes for our 1044 extended family as well: Dave Moore, Paul Pearce, Judy Hoffman, Allie and Ryan Janoch, Pat and Caitlin Virtue, and Ellen Stuart. I can't capture here how much that time has meant to me, but it's been an incredible ride, and I look forward to more.

And last but certainly not least, I thank my family for supporting me on my long journey here. I don't know if I would be on this path if I hadn't looked at my father and thought "you know, being a professor seems pretty cool." And finally, thanks to Emily Cogsdill; I'm not sure what the future holds, but from here, it looks bright.

Chapter 1

Introduction

Language serves many purposes; a crucial one is storytelling. Usually a story has a few main characters, or entities, and we talk about a series of events involving those entities. Understanding a story starts with understanding the entities and what they're doing, but this very first step is already a very difficult one for computers to do. Consider the following example:

> ***Barack Obama*** *signed the Affordable Care Act into law on March 23, 2010. The law is viewed as one of* ***the president****'s signature achievements.* ***Obama*** *has had to defend the law from many legislative and judicial challenges, yet its eventual fate is still uncertain as* ***he*** *prepares to leave office.*

Figure 1.1: A short passage illustrating the importance of entity reference.

Suppose we want to extract information from this document and understand the events it describes. A naïve system based around a syntactic parser and simple string matching might be able to identify a *signing* event with *Barack Obama* as the agent and *the Affordable Care Act* as the patient. However, we can understand little else from this text using only surface clues—references like *the president*, *its*, and *he* are not resolvable without more sophisticated processing, so we can say very little about the events these references take part in. To have any hope of understanding notions like the law being a signature achievement or the fact that Obama is soon to leave office, we must first be able to resolve these references.

This problem roughly cleaves into two main parts: coreference resolution and entity linking. Coreference resolution as we define it is the problem of identifying spans of text in a document[1] that refer to the same real-world entity. Entity linking, also called named

[1]There are cross-document coreference systems in the literature (Singh et al., 2011; Lee et al., 2012), but treating coreference resolution and entity linking together largely obviates the need to handle cross-document phenomena in the coreference stage; entity canonicalization is mediated by the knowledge base.

entity disambiguation or Wikification,[2] is the task of identifying for each of these spans the entry in a knowledge base that it refers to. Taken together, these two tasks allow you to fully resolve the actors in a text: for example, in our passage above, coreference resolution tells us that all bolded chunks refer to the same thing, and entity linking might resolve this to the entity `https://en.wikipedia.org/wiki/Barack_Obama`, so we can attribute all of these actions to `Barack_Obama` and ad new information to our knowledge base accordingly.

1.1 Contributions of this Thesis

The bulk of this thesis aims to address these problems of coreference resolution and entity linking, and in particular we focus on addressing them with a unified, joint model. After discussing some preliminaries in **Chapter 2**, we introduce the coreference portion of this model in **Chapter 3**. Our contribution here is twofold: first, we devise a simple, data-driven model that captures the heterogeneous phenomena exhibited by coreference (see Section 1.2), and second, we do so in a framework that allows this coreference component to be easily integrated into a joint model of entity resolution.

Chapter 4 discusses the unified model, which jointly tackles the tasks of coreference, entity linking, and semantic typing of entities. Our joint model improves performance on all three tasks compared to its individual component models; a large part of this gain is due to deeper integration of world knowledge from Wikipedia, imported via coarse semantic types. Semantic typing is largely important only insofar as it allows us to do better on the other two, but it does suggest a path towards integration with a full-fledged information extraction pipeline, where typological information can be important for validating extracted entity relations.

Finally, in **Chapter 5**, we return to our original goal of narrative understanding and show how coreference resolution can help us produce better automatic summaries of documents. When generating a summary, we need to take special care that references are clear: a sentence taken out of context might contain pronouns whose antecedents are no longer resolvable, making the summary inherently ambiguous. We draw on our entity resolution machinery to repair these cases, especially pronouns, in the context of an extractive and compressive summarization system. Although we only handle a subset of anaphora resolution, our method provides the first steps towards building summarization systems with heightened discourse and entity awareness.

[2]Some authors (Milne and Witten, 2008) define Wikification as specifically reproducing annotations in Wikipedia rather than simply resolving entities to articles in Wikipedia and using it as a knowledge base, hence why we use the term entity linking in this work.

1.2 Coreference Resolution

Coreference resolution captures a within-document view of what an entity is. A coreference system takes as its input a document and outputs a clustering of *mentions*. A mention is a span of text in the document that refers to an entity in the world, like *Barack Obama* or *he*. In Figure 1.1, there are two coreference clusters containing more than one mention, one consisting of the set {*Barack Obama, the president, Obama, he*} and the other consisting of {*the Affordable Care Act, The law, the law, its*}.

One key feature of coreference that this example highlights the heterogeneity of the task. Unlike in tasks like part-of-speech tagging or syntactic parsing, different subproblems of coreference present radically different phenomena due to the wide range of possible input types. These include:

Proper names An entity might be mentioned by its full name (*Barack Obama*) or by shortened variants (*Obama*). In addition, systems need to be aware of other name variation, nicknames, and potential misspellings in informal text.

Pronouns Once an entity is referred to the first time, subsequent mentions will typically be referred to in abbreviated fashion, especially with pronouns. In English, pronoun resolution requires understanding notions of syntax (the *him* in *John asked him* cannot refer to *John*), discourse (for entity salience), as well as semantic notions of agreement that are reflected in the pronoun, namely number, animacy, and gender.

Common nouns Common nouns fall somewhere between the other two cases: they are less ambiguous than pronouns, but do not exhibit the clear lexical overlap of proper names. These cases are most likely to draw on assumed world knowledge, e.g., that *Barack Obama* is *the president* or that the *Affordable Care Act* is a *law*.

We will discuss each of these in more detail in Chapter 3. Contrary to conventional wisdom, we show that a log-linear model with a small number of feature templates actually can model each of these cases in a relatively uniform way. These feature templates look at surface lexical information in the document and generate large numbers of features, which prove flexible enough to capture most of these heterogeneous aspects of coreference. However, one class of examples that remains difficult is those relying on semantic information and world knowledge, especially resolving nontrivial cases of common nouns (e.g. recognizing that *Barack Obama* is *the president* if we haven't explicitly seen the string *president* earlier in the document). Our joint model in Chapter 4 addresses this shortcoming by integrating with an entity resolution model and consulting Wikipedia, which imports appropriate knowledge into the system to improve performance.

Coreference resolution is a problem with a long history in AI and NLP. Algorithms and computational reasoning about coreference dates back at least 50 years (Hobbs, 1977; Hobbs, 1979) and data-driven approaches have been extensively studied over the past 15 years

(Soon et al., 2001). Coreference has been recognized as a particularly difficult problem for computers to solve and AI challenge problems have been formulated around it (Winograd, 1972; Levesque et al., 2012). Computational approaches have structured the problem in various ways, but the dominant approaches largely score links between mentions as the fundamental units of coreference (Soon et al., 2001; Ng and Cardie, 2002; Stoyanov et al., 2009). Various authors have considered different ways of combining link-based predictions into clusters or treated the task at the entity level (Rahman and Ng, 2009; Raghunathan et al., 2010), though mostly with limited success. We will see that our approach, which maintains no notion of entities at inference time, outperforms much of the past work that does.

1.3 Entity Linking

While coreference is concerned with entity identification within a document, entity linking deals with entity *disambiguation* external to a document. Consider the following example.

> ***Michael Jordan*** *gave a lecture on probabilistic graphical models as part of a Big Data workshop. The UC Berkeley professor has published numerous papers in the field of statistical machine learning.*

Figure 1.2: Passage illustrating an example mention (*Michael Jordan*) that needs to be resolved to an entry in a knowledge base.

The canonical question of entity linking involves resolving *Michael Jordan* to his corresponding Wikipedia article. There are two natural choices:

- `https://en.wikipedia.org/wiki/Michael_I._Jordan` (the professor)
- `https://en.wikipedia.org/wiki/Michael_Jordan` (the basketball player)

The simplest approach to this task is a most-frequent-class baseline (Cucerzan, 2007; Milne and Witten, 2008): extract all links within Wikipedia and link the current mention to its most frequently linked target. This baseline works surprisingly well (Ratinov et al., 2011), but fails in this case, as occurrences of the string *Michael Jordan* on Wikipedia more frequently link to the basketball player than to the machine learning professor.

In order to go beyond this baseline, there are many clues that one can and should draw upon. One such clue is topic information from nearby words: in Figure 1.2, the presence of *lecture*, *published*, and *statistical* should indicate that we are in the academic domain rather than the sports domain. We can compare the mention's context with the text of the target Wikipedia article using some kind of topic similarity metric, often cosine similarity of vector-based topic representations like tf-idf (Cucerzan, 2007; Ratinov et al., 2011). Another

cue is nearby disambiguations: in Figure 1.2, successfully linking *Big Data* or *probabilistic graphical models* localizes us to a part of the Wikipedia graph structure closer to the machine learning professor than to the basketball player. Such techniques fall under the umbrella of "collective disambiguation" (Hoffart et al., 2011; Ratinov et al., 2011).

Our focus in Chapter 4 is on improving entity linking through tighter integration with coreference and semantic typing. Coreference information has been shown to be effective for this task in a pipelined fashion (Cheng and Roth, 2013), but never in the context of a joint model. In Figure 1.2 *Michael Jordan* is coreferent with *the UC Berkeley professor*, which gives us a direct and useful clue about entity linking, since the first sentence of the Wikipedia article on `Michael_I._Jordan` describes him as a professor. Moreover, entity linking helps import useful semantic information from the perspective of coreference, which suggests that we should handle these tasks in a joint rather than a pipelined fashion.

Compared to tasks like coreference resolution and syntactic parsing, entity linking much less mature and has a less standardized experimental setup. This is especially true because it relies on the presence of a knowledge base. Much of the early work on entity linking focuses on linking to Wikipedia (Cucerzan, 2007; Milne and Witten, 2008), but it has since expanded to other (related) knowledge bases such as YAGO (Hoffart et al., 2011). There are a slew of datasets available for this task, some small ones created primarily for evaluation (Ratinov et al., 2011) and some which are standard datasets for other tasks which entity linking annotations have been added to (Hoffart et al., 2011; Bentivogli et al., 2010).

1.4 Semantic Typing

The third entity analysis task we consider in this work is semantic typing, which consists of labeling mentions with coarse semantic types, e.g. *Barack Obama* is a `Person`. This task is broadly similar to classical named entity recognition (NER); however, a key part of that task is identifying named entity spans, whereas we are more interested in typing pre-identified mentions.

Standard approaches to NER typically use conditional random fields with surface lexical features (Tjong Kim Sang and De Meulder, 2003). Our joint modeling approach in Chapter 4 will allow us to incorporate two additional sources of information. First, a constraint arises from coreference: two coreferent mentions should be the same semantic type, which other authors have captured with predetermined coreference links (Krishnan and Manning, 2006; Ratinov and Roth, 2009). Second, Wikipedia category information can inform us as to a proper noun's semantic type (Nothman et al., 2013). We combine these approaches in a single system and show that they both improve the performance of our semantic typing component, which in turn improves both coreference and entity linking.

1.5 Automatic Summarization

In Chapter 5, we show how the entity analysis machinery that we've built up throughout this work can be used for an actual user-facing task, specifically automatic summarization. We address the task of single-document summarization (McKeown et al., 1995; Marcu, 1998; Mani, 2001), which has been less studied than multi-document summarization in recent years. This task requires finding the best k-word summary (for a pre-specified value of k) of a given document.

Classical approaches to the summarization task have typically focused on sentence extraction, since reproducing whole sentences from the source document saves the system designer from having to deal with a difficult generation problem. To evaluate sentences for inclusion in the summary, criteria such as marginal relevance of including new sentences in the summary (Carbonell and Goldstein, 1998) or bigram frequency information (Gillick and Favre, 2009) serve as simple heuristic measures of content relevance. Beyond simple extraction, past work has additionally integrated sentence compression (Lin, 2003; Martins and Smith, 2009; Berg-Kirkpatrick et al., 2011), leading to models that are extractive and compressive. However, there has been relatively little success in implementing bolder "abstractive" summarization models: most work in this vein is either speculative (Liu et al., 2015) or restricted in scope, e.g. operating at the sentence level (Rush et al., 2015).

Moreover, despite the apparent importance of discourse understanding for this task, there have also been few successes at using any kind of cross-sentence discourse information (Clarke and Lapata, 2010). However, this kind of information can make a difference: from experiments with an extractive and compressive summarization system, we found roughly 60% of summaries included a pronoun whose antecedent failed to be included in the summary. In Chapter 5, we show how using our entity analysis tools can help us fix these cases and lead to more easily understood pronouns. We formulate constraints based on the posteriors from an upstream coreference model: we allow ourselves to expand pronouns into full mentions when we are highly confident about our coreference decisions, and otherwise merely constrain additional context to be included as a way of hedging our bets against coreference errors. Conservative as this is, this approach still gainfully reduces the number of pronoun errors. While this is a limited victory, it represents a successful application of this kind of preprocessing, and we believe it provides a starting point for building a more sophisticated, discourse-aware summarization system.

Chapter 2

Preliminaries: Datasets and Metrics

As the methods we will be developing in this thesis are data-driven, we begin by giving some background on the datasets we use in this work. Specifically, for entity analysis, we will use two main corpora: the OntoNotes dataset (Hovy et al., 2006), used in the CoNLL 2011 and 2012 shared tasks (Pradhan et al., 2011; Pradhan et al., 2012), and the ACE 2005 dataset (NIST, 2005), developed as part of the ACE program. We also touch on the metrics we use to evaluate coreference resolution, as these are not as straightforward as the metrics for the other tasks we discuss in this work.

2.1 OntoNotes

The OntoNotes corpus was created by an ongoing collaborative annotation effort to produce a large, cross-domain, cross-lingual corpus of text with several layers of syntactic and semantic annotation. The English portion, which we focus on in this work, consists of around 1.3 million tokens of mixed domain data: roughly 35% newswire and the rest split roughly evenly between broadcast news, broadcast conversations (i.e., televised interviews), magazine articles, and web data. Systems trained on this mixture of domains should be more robust than systems trained exclusively on newswire data.

The annotations themselves are shown in Figure 2.1. The annotations we will use are the coreference annotations (Chapters 3 and 4) and named entity annotations (Chapter 4). We also use constituency parses in this work, but rather than using the gold constituency parses provided in the data, we instead choose the more realistic approach of using constituency parses produced by an automatic system: in the CoNLL 2011/2012 data, a standardized set of parses produced with the parser of Charniak and Johnson (2005) are provided in separate files. We use those as sources of features as well as to identify mention boundaries.

We see a few interesting things about the coreference annotation in the last column of Figure 2.1. First, we can have nested annotations: *its Igaras pulp and paper mill in Brazil* is a mention of the entity with index 1, but it contains two other mentions, namely *its* (Manville Corp., entity 2) and *Brazil*. Second, "singleton" mentions, or mentions of entities

```
#begin document (nw/wsj/12/wsj_1278); part 000
nw/wsj/12/wsj_1278   0    0       Manville    NNP          (TOP(S(NP*         -    -   -   -     (ORG*   (ARG0*            *         *        (2
nw/wsj/12/wsj_1278   0    1          Corp.    NNP                *)        -    -   -   -        *)       *)           *         *         2)
nw/wsj/12/wsj_1278   0    2          said    VBD             (VP*        say  01   1   -        *     (V*)           *         *         -
nw/wsj/12/wsj_1278   0    3            it    PRP          (SBAR(S(NP*)        -    -   -   -        *   (ARG1*       (ARG0*)        *        (2)
nw/wsj/12/wsj_1278   0    4          will     MD                (VP*        -    -   -   -        *       *    (ARGM-MOD*)        *         -
nw/wsj/12/wsj_1278   0    5         build     VB                (VP*      build  01   1   -        *       *          (V*)         *         -
nw/wsj/12/wsj_1278   0    6             a     DT             (NP(NP*        -    -   -   -        *       *        (ARG1*    (ARG0*        (0
nw/wsj/12/wsj_1278   0    7             $      $           (NML(QP*        -    -   -   -   (MONEY*       *           *         *         -
nw/wsj/12/wsj_1278   0    8            24     CD                 *        -    -   -   -        *       *           *         *         -
nw/wsj/12/wsj_1278   0    9       million     CD                *))       -    -   -   -        *)      *           *         *         -
nw/wsj/12/wsj_1278   0   10         power     NN                 *      power  -   3   -        *       *           *         *         -
nw/wsj/12/wsj_1278   0   11         plant     NN                 *)     plant  -   1   -        *       *           *         *)        -
nw/wsj/12/wsj_1278   0   12            to     TO          (SBAR(S(VP*        -    -   -   -        *       *           *         *         -
nw/wsj/12/wsj_1278   0   13       provide     VB                (VP*    provide  01   1   -        *       *           *       (V*)        -
nw/wsj/12/wsj_1278   0   14   electricity     NN                (NP*)       -    -   -   -        *       *           *    (ARG1*)        -
nw/wsj/12/wsj_1278   0   15            to     IN                (PP*        -    -   -   -        *       *           *    (ARG2*         -
nw/wsj/12/wsj_1278   0   16           its   PRP$             (NP(NP*        -    -   -   -        *       *           *         *    (1|(2)
nw/wsj/12/wsj_1278   0   17        Igaras    NNP                 *        -    -   -   -     (FAC)      *           *         *         -
nw/wsj/12/wsj_1278   0   18          pulp     NN              (NML*      pulp   -   -   -        *       *           *         *         -
nw/wsj/12/wsj_1278   0   19           and     CC                 *        -    -   -   -        *       *           *         *         -
nw/wsj/12/wsj_1278   0   20         paper     NN                 *)     paper  -   1   -        *       *           *         *         -
nw/wsj/12/wsj_1278   0   21          mill     NN                 *)      mill  -   1   -        *       *           *         *         -
nw/wsj/12/wsj_1278   0   22            in     IN                (PP*        -    -   -   -        *       *           *         *         -
nw/wsj/12/wsj_1278   0   23        Brazil    NNP   (NP*)))))))))))))        -    -   -   -     (GPE)      *)          *)        *)    1)|0)
nw/wsj/12/wsj_1278   0   24             .      .                 *))       -    -   -   -        *       *           *         *         -
```

Figure 2.1: Example sentence from a document in the OntoNotes corpus (Hovy et al., 2006) in the CoNLL format (Pradhan et al., 2011). In addition the words and various structural columns, annotations are provided for part-of-speech tags (5th column), constituency parses (6th column), named entity chunks (11th column), and coreference resolution (15th column). Predicate-argument structure is also provided, but we do not use that in this work.

that are only discussed once, are not annotated. We see this from the fact that *Brazil* is not assigned to a coreference cluster;[1] Brazil is never referred to subsequently in this document.

The named entity annotations are provided as labeled chunks in the 11th column. The most salient distinction of the OntoNotes dataset as an NER corpus is the fact that named entities are annotated according to a 17-class tagset, instead of the `PER/ORG/LOC/MISC` tagset that has been used in prior work (Tjong Kim Sang and De Meulder, 2003). Some of these annotations are quite fine-grained: we see `FAC` in Figure 2.1, for Facility.

For the CoNLL 2011 and CoNLL 2012 shared tasks (Pradhan et al., 2011; Pradhan et al., 2012), standard training, development, and testing splits were established for use in coreference evaluation, which we follow in this work. The training, development, and testing data are sampled uniformly across these different domains: for every 10 numbered files, 8 are in train, 1 is in development, and 1 is in test. This means that the domain composition of each set should be almost identical, and we will never encounter data that is truly out-of-domain from the standpoint of a trained system.

2.2 ACE

The OntoNotes corpus is large and relatively high quality, but it has several shortcomings for our purposes, foremost among which is that it does not include entity link annotations. Therefore, we also experiment with the ACE corpus (NIST, 2005), a set of documents from

[1] Two clusters end on the token *Brazil*, but note that there is no separate annotation that starts and ends on that token.

newswire, online newsgroups, and other genres that are annotated primarily with an eye towards information extraction. Annotations are provided in terms of token offsets and are exclusively focused on entities and events. The entity annotations include mention boundaries, entity types, and entity coreference. Unlike OntoNotes, singleton entities are annotated, but unlike OntoNotes, annotation is additionally restricted to only seven entity types: `Person`, `Organization`, `GPE`, `Location`, `Facility`, `Weapon`, and `Vehicle`. Bentivogli et al. (2010) later added entity link annotations by specifying one or more articles in Wikipedia corresponding to every non-pronominal, non-`Weapon`, non-`Vehicle` mention. It is important to note that nominal mentions are linked to articles that describe them rather than the entities that they refer to. For example, if we refer to *Barack Obama* subsequently as *the president*, *the president* is annotated as linking to `President_of_the_United_States` and `President` as opposed to `Barack_Obama`. Other entity linking datasets take a different approach and annotate each mention with the best link choice for the entire entity Hoffart et al., 2011.

As with OntoNotes, we use automatic preprocessing to determine syntactic parses; we do this with the Berkeley Parser (Petrov et al., 2006). We additionally have to tokenize and sentence-split the data, which we do with the tools bundled with Reconcile (Stoyanov et al., 2010). Once we do this process and project the annotations onto our new tokens, the ACE corpus annotations can be treated identically to the OntoNotes annotations.

2.3 Coreference Evaluation Metrics

Coreference is a difficult task to evaluate. Fundamentally, we are comparing a predicted clustering of mentions to a gold-standard clustering of a (potentially different) set of gold-standard mentions. However, there is no "most natural" accuracy metric or one that can be easily motivated from first principles. Almost any metric will inherently make value judgments about coreference that are difficult to judge irrespective of a downstream task: for example, should erroneously breaking off one mention from a large cluster incur a larger penalty than breaking off a mention from a small cluster? Perhaps the best evidence of there being difficult design choices is that the MUC metric was discovered to give very high scores to systems that output very few clusters, even when those clusterings are clearly incorrect (Luo, 2005).

The most standard approach, although not the most elegant one, is to use an arithmetic mean of three metrics; this is how the evaluation was conducted for the CoNLL 2011 Shared Task (Pradhan et al., 2011). We refer the reader to (Pradhan et al., 2014) for unified formal definitions of each of the metrics. Each of these metrics has different strengths and weaknesses, so we describe them in a high level in the following few paragraphs. However, it is worth noting that in non-pathological cases (e.g., when we are dealing with relatively small variations on credible coreference systems), these three metrics typically track one another quite closely.

MUC MUC was originally proposed in Vilain et al. (1995). MUC recall penalizes a clustering based on the number of clusters that a given reference cluster is split into; precision is the same metric with the roles of the reference and prediction reversed. Critically, we obtain 100% recall by putting all mentions into a single cluster, and precision does not overly suffer. As a result, high MUC scores are generally obtained by improving recall and clustering mentions more aggressively, hence MUC is more recall-oriented.

B^3 B^3 was designed to address the shortcomings of MUC, and was first presented in Bagga and Baldwin (1998). This metric makes several different design choices than MUC does, and is based on intersections between all pairs of clusters in such a way that a mention's importance is proportional to the size of the cluster that contains it. Making a single large cluster therefore ends up reducing precision almost to zero and hurts overall B^3 score substantially. In this sense, B^3 is the oppose of MUC and is more precision-oriented.

CEAF_e CEAF_e (Luo, 2005) is a more sophisticated metric to compute. Given a gold coreference cluster and a predicted coreference cluster, we can score their overlap by simply taking the size of the intersection over the average of the two cluster sizes. However, actually scoring a set of coreference partitions requires finding a maximum matching between the gold and the prediction and scoring the aligned clusters according to that alignment. This process makes the metric more complex and non-decomposable, but it is not subject to the same biases as MUC and B^3.

Chapter 3

Simple Learning-Based Coreference Resolution[1]

3.1 Introduction

Coreference resolution is a multi-faceted task: humans resolve references by exploiting contextual and grammatical clues, as well as semantic information and world knowledge, so capturing each of these will be necessary for an automatic system to fully solve the problem. As we established in Section 1.2, there are also many different kinds of coreference phenomena an automatic system needs to be able to address. Acknowledging this complexity, coreference systems, either learning-based (Bengtson and Roth, 2008; Stoyanov et al., 2010; Haghighi and Klein, 2010; Rahman and Ng, 2011b) or rule-based (Haghighi and Klein, 2009; Lee et al., 2011), draw on diverse information sources and complex heuristics to resolve pronouns, model discourse, determine anaphoricity, and identify semantically compatible mentions. However, this leads to systems with many heterogenous parts that can be difficult to interpret or modify.

We build a learning-based, mention-synchronous coreference system that aims to use the simplest possible set of features to tackle the various aspects of coreference resolution. Though they arise from a small number of simple templates, our features are numerous, which works to our advantage: we can both implicitly model important linguistic effects and capture other patterns in the data that are not easily teased out by hand. As a result, our data-driven, homogeneous feature set is able to achieve high performance despite only using surface-level document characteristics and shallow syntactic information. We win "easy victories" without designing features and heuristics explicitly targeting particular phenomena.

Though our approach is successful at modeling syntax, we find semantics to be a much more challenging aspect of coreference. Our base system uses only two recall-oriented features on nominal and proper mentions: head match and exact string match. Building on these features, we critically evaluate several classes of semantic features which intuitively should

[1] An early version of this chapter appeared in Durrett and Klein (2013).

prove useful but have had mixed results in the literature, and we observe that they are ineffective for our system. However, these features are beneficial when gold mentions are provided to our system, leading us to conclude that the large number of system mentions extracted by most coreference systems (Lee et al., 2011; Fernandes et al., 2012) means that weak indicators cannot overcome the bias against making coreference links. Capturing semantic information in this shallow way is an "uphill battle" due to this structural property of coreference resolution.

Nevertheless, using a simple architecture and feature set, our final system outperforms the two best publicly available English coreference systems, the Stanford system (Lee et al., 2011) and the IMS system (Björkelund and Farkas, 2012), by wide margins: 3.5% absolute and 1.9% absolute, respectively, on the CoNLL metric.

3.2 Experimental Setup

Throughout this chapter, we use the datasets from the CoNLL 2011 shared task[2] (Pradhan et al., 2011), which is derived from the OntoNotes corpus (Hovy et al., 2006). When applicable, we use the standard automatic parses and NER tags for each document. All experiments use system mentions except where otherwise indicated. For each experiment, we report MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), and CEAF_e (Luo, 2005), as well as their average, the CoNLL metric. All metrics are computed using version 5 of the official CoNLL scorer.[3]

3.3 A Mention-Synchronous Framework

We first present the basic architecture of our coreference system, independent of a feature set. Unlike binary classification-based coreference systems where independent binary decisions are made about each pair (Soon et al., 2001; Bengtson and Roth, 2008; Versley et al., 2008; Stoyanov et al., 2010), we use a log-linear model to select at most one antecedent for each mention or determine that it begins a new cluster (Denis and Baldridge, 2008). In this mention-ranking or mention-synchronous framework, features examine single mentions to evaluate whether or not they are anaphoric and pairs of mentions to evaluate whether or not they corefer. While other work has used this framework as a starting point for entity-level systems (Luo et al., 2004; Rahman and Ng, 2009; Haghighi and Klein, 2010), we will show that a mention-synchronous approach is sufficient to get state-of-the-art performance on its own.

[2]This dataset is identical to the English portion of the CoNLL 2012 data, except for the absence of a small pivot text.

[3]Note that this version of the scorer implements a modified version of B^3, described in Cai and Strube (2010), that was used for the CoNLL shared tasks. The implementation of CEAF_e is also not exactly as described in Luo et al. (2004), but for completeness we include this metric as well.

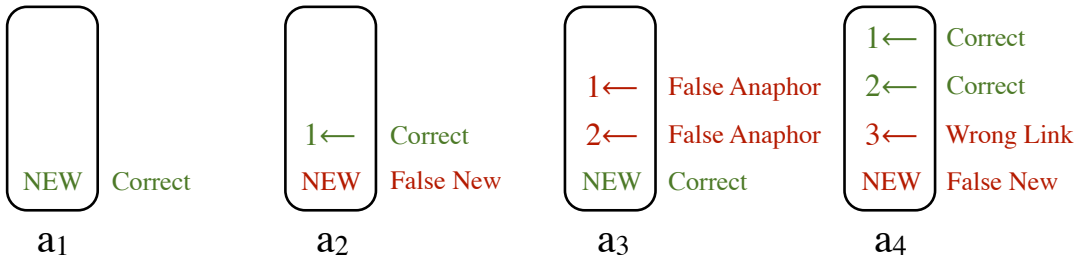


Figure 3.1: The basic structure of our coreference model. The ith mention in a document has i possible antecedence choices: link to one of the $i-1$ preceding mentions or begin a new cluster. We place a distribution over these choices with a log-linear model. Structurally different kinds of errors are weighted differently to optimize for final coreference loss functions; error types are shown corresponding to the decisions for each mention.

3.3.1 Mention Detection

Our system first identifies a set of predicted mentions from text annotated with parses and named entity tags. We extract three distinct types of mentions: proper mentions from all named entity chunks except for those labeled as QUANTITY, CARDINAL, or PERCENT, pronominal mentions from single words tagged with PRP or PRP$, and nominal mentions from all other maximal NP projections. These basic rules are similar to those of Lee et al. (2011), except that their system uses an additional set of filtering rules designed to discard instances of pleonastic *it*, partitives, certain quantified noun phrases, and other spurious mentions. In contrast to this highly engineered approach and to systems which use a trained classifier to compute anaphoricity separately (Rahman and Ng, 2009; Björkelund and Farkas, 2012), we aim for the highest possible recall of gold mentions with a low-complexity method, leaving us with a large number of spurious system mentions that we will have to reject later.

3.3.2 Coreference Model

Figure 3.1 shows the mention-ranking architecture that serves as the backbone of our coreference system. Assume we have extracted n mentions from a document x, where x denotes the surface properties of a document and any precomputed information. The ith mention in a document has an associated random variable a_i taking values in the set $\{1, \ldots, i-1, \text{NEW}\}$; this variable specifies mention i's selected antecedent or indicates that it begins a new coreference chain. A setting of the a_i, denoted by $\mathbf{a} = (a_1, ..., a_n)$, implies a unique set of coreference chains C that serve as our system output.

We use a log linear model of the conditional distribution $p(a|x)$ as follows:

$$p(\mathbf{a}|x) \propto \exp\left(\sum_{i=1}^{n} \mathbf{w}^\top \mathbf{f}(i, a_i, x)\right)$$

where $\mathbf{f}(i, a_i, x)$ is a feature function that examines the coreference decision a_i for mention i with document context x. When $a_i = \text{NEW}$, the features fired indicate the suitability of the given mention to be anaphoric or not; when $a_i = j$ for some j, the features express aspects of the pairwise linkage, and can examine any relevant attributes of the anaphor i or the antecedent j, since information about each mention is contained in x.

Inference in this model is efficient: because $\log p(\mathbf{a}|x)$ decomposes linearly over mentions, we can compute $a_i = \arg\max_{a_i} p(a_i|x)$ separately for each mention and return the set of coreference chains implied by these decisions.

3.3.3 Learning

During learning, we optimize for conditional log-likelihood augmented with a parameterized loss function. The main complicating factor in this process is that the supervision in coreference consists of a gold clustering C^* defined over gold mentions. This is problematic for two reasons: first, because the clustering is defined over gold mentions rather than our system mentions, and second, because a clustering does not specify a full antecedent structure of the sort our model produces. We can address the first of these problems by imputing singleton clusters for mentions that do not appear in the gold standard; our system will then simply learn to put spurious mentions in their own clusters. Singletons are always removed before evaluation because the OntoNotes corpus does not annotate them, so in this way we can neatly dispose of spurious mentions. To address the lack of explicit antecedents in C^*, we simply sum over all possible antecedent structures licensed by the gold clusters.

Formally, we will maximize the conditional log-likelihood of the set $\mathcal{A}(C^*)$ of antecedent vectors a for a document that are *consistent* with the gold annotation.[4] Consistency for an antecedent choice a_i under gold clusters C^* is defined as follows:

1. If $a_i = j$, a_i is consistent iff mentions i and j are present in C^* and are in the same cluster.

2. If $a_i = \text{NEW}$, a_i is consistent off mention i is not present in C^*, or it is present in C^* and has no gold antecedents, or it is present in C^* and none of its gold antecedents are among the set of system predicted mentions.

Given t training examples of the form (x_k, C_k^*), we write the following likelihood function:

$$\ell(\mathbf{w}) = \sum_{k=1}^{t} \log\left(\sum_{\mathbf{a}^* \in \mathcal{A}(C_k^*)} p'(\mathbf{a}^*|x_k)\right) + \lambda\|\mathbf{w}\|_1$$

[4]Because of this marginalization over latent antecedent choices, our objective is non-convex.

where $p'(\mathbf{a}|x_k) \propto p(\mathbf{a}|x_k)\exp(l(\mathbf{a}, C_k^*))$ with $l(\mathbf{a}, C^*)$ being a real-valued loss function. The loss here plays an analogous role to the loss in structured max-margin objectives; incorporating it into a conditional likelihood objective is a technique called softmax-margin (Gimpel and Smith, 2010).

Our loss function $l(\mathbf{a}, C^*)$ is a weighted linear combination of three error types, examples of which are shown in Figure 3.1. A false anaphor (FA) error occurs when a_i is chosen to be anaphoric when it should start a new cluster. A false new (FN) error occurs in the opposite case, when a_i wrongly indicates a new cluster when it should be anaphoric. Finally, a wrong link (WL) error occurs when the antecedent chosen for a_i is the wrong antecedent (but a_i is indeed anaphoric). Our final parameterized loss function is a weighted sum of the counts of these three error types:

$$l(\mathbf{a}, C^*) = \alpha_{\text{FA}}\text{FA}(\mathbf{a}, C^*) + \alpha_{\text{FN}}\text{FN}(\mathbf{a}, C^*) + \alpha_{\text{WL}}\text{WL}(\mathbf{a}, C^*)$$

where $\text{FA}(\mathbf{a}, C^*)$ gives the number of false anaphor errors in prediction a with gold chains C^* (FN and WL are analogous). By setting α_{FA} low and α_{FN} high relative to α_{WL}, we can counterbalance the high number of singleton mentions and bias the system towards making more coreference linkages. We set $(\alpha_{\text{FA}}, \alpha_{\text{FN}}, \alpha_{\text{WL}}) = (0.1, 3.0, 1.0)$ and $\lambda = 0.001$ and optimize the objective using AdaGrad (Duchi et al., 2011).

3.4 Easy Victories from Surface Features

Our primary goal with this work is to show that a high-performance coreference system is attainable with a small number of feature templates that use only surface-level information sources. These features will be general-purpose and capture linguistic effects to the point where standard heuristic-driven features are no longer needed in our system.

3.4.1 Surface Features and Conjunctions

Our SURFACE feature set only considers the following properties of mentions and mention pairs:

- Mention type (nominal, proper, or pronominal)
- The complete string of a mention
- The semantic head of a mention
- The first word and last word of each mention
- The word immediately preceding and the word immediately following a mention
- Mention length, in words

Feature name	Count
Features on the current mention	
[ANAPHORIC] + [HEAD WORD]	41371
[ANAPHORIC] + [FIRST WORD]	18991
[ANAPHORIC] + [LAST WORD]	19184
[ANAPHORIC] + [PRECEDING WORD]	54605
[ANAPHORIC] + [FOLLOWING WORD]	57239
[ANAPHORIC] + [LENGTH]	4304
Features on the antecedent	
[ANTECEDENT HEAD WORD]	57383
[ANTECEDENT FIRST WORD]	24239
[ANTECEDENT LAST WORD]	23819
[ANTECEDENT PRECEDING WORD]	53421
[ANTECEDENT FOLLOWING WORD]	55718
[ANTECEDENT LENGTH]	4620
Features on the pair	
[EXACT STRING MATCH (T/F)]	47
[HEAD MATCH (T/F)]	46
[SENTENCE DISTANCE, CAPPED AT 10]	2037
[MENTION DISTANCE, CAPPED AT 10]	1680

Table 3.1: Our SURFACE feature set, which exploits a small number of surface-level mention properties. Feature counts for each template are computed over the training set, and include features generated by our conjunction scheme (not explicitly shown in the table; see Figure 3.2), which yields large numbers of features at varying levels of expressivity.

- Two distance measures between mentions (number of sentences and number of mentions)

Table 3.1 shows the SURFACE feature set. Features that look only at the current mention fire on all decisions ($a_i = j$ or $a_i = \text{NEW}$), whereas features that look at the antecedent in any way (the latter two groups of features) only fire on pairwise linkages ($a_i \neq \text{NEW}$).

Two conjunctions of each feature are also included: first with the "type" of the mention being resolved (either NOMINAL, PROPER, or, if it is pronominal, the citation form of the pronoun), and then additionally with the antecedent type (only if the feature is over a pairwise link). This conjunction process is shown in Figure 3.2. Note that features that just examine the antecedent will end up with conjunctions that examine properties of the current mention as well, as shown with the ANT. HEAD feature in the figure.

Finally, we found it beneficial for our lexical indicator features to only fire on words occurring at least 20 times in the training set; for rare words, we use the part of speech of the word instead.

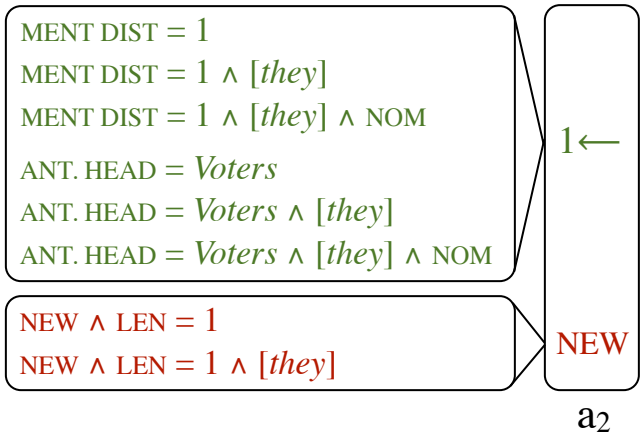


Figure 3.2: Demonstration of the conjunction scheme we use. Each feature on anaphoricity is conjoined with the type (NOMINAL, PROPER, or the citation form if it is a pronoun) of the mention being resolved. Each feature on a mention pair is additionally conjoined with the types of the current and antecedent mentions.

	MUC	B^3	CEAF_e	Avg.
STANFORD	60.46	65.48	47.07	57.67
IMS	62.15	65.57	46.66	58.13
SURFACE	**64.39**	**66.78**	**49.00**	**60.06**

Table 3.2: Results for our SURFACE system, the STANFORD system, and the IMS system on the CoNLL 2011 development set. Complete results are shown in Table 3.7. Despite using limited information sources, our system is able to substantially outperform the other two, the two best publicly-available English coreference systems. Bolded values are significant with $p < 0.05$ according to a bootstrap resampling test.

The performance of our system is shown in Table 3.2. We contrast our performance with that of the Stanford system (Lee et al. (2011), the winner of the CoNLL 2011 shared task) and the IMS system (Björkelund and Farkas (2012), the best publicly available English coreference system). Despite its simplicity, our SURFACE system is sufficient to outperform these sophisticated systems: the Stanford system uses a cascade of ten rule-based sieves each of which has customized heuristics, and the IMS system uses a similarly long pipeline consisting of a learned referentiality classifier followed by multiple resolvers, which are run in sequence and rely on the outputs of the previous resolvers as features.

3.4.2 Data-Driven versus Heuristic-Driven Features

Why are the SURFACE features sufficient to give high coreference performance, when they do not make apparent reference to important linguistic phenomena? The main reason is that they actually do capture the same phenomena as standard coreference features, just implicitly. For example, rather than having rules targeting person, number, gender, or animacy of mentions, we use conjunctions with pronoun identity, which contains this information. Rather than explicitly writing a feature targeting definiteness, our indicators on the first word of a mention will capture this and other effects. And finally, rather than targeting centering theory (Grosz et al., 1995) with rule-based features identifying syntactic positions (Stoyanov et al., 2010; Haghighi and Klein, 2010), our features on word context can identify configurational clues like whether a mention is preceded or followed by a verb, and therefore whether it is likely in subject or object position.[5]

Not only are data-driven features able to capture the same phenomena as heuristic-driven features, but they do so at a finer level of granularity, and can therefore model more patterns in the data. To contrast these two types of features, we experiment with three ablated versions of our system, where we replace data-driven features with their heuristic-driven counterparts:

1. Instead of using an indicator on the first word of a mention (1STWORD), we instead fire a feature based on that mention's manually-computed definiteness (DEF).

2. Instead of conjoining features on pronominal-pronominal linkages with the citation form of each pronoun (PRONCONJ), we only conjoin with a PRONOUN indicator and add features targeting the person, number, gender, and animacy of the two pronouns (AGR).

3. Instead of using our context features on the preceding and following word (CONTEXT), we use manual determinations of when mentions are in subject, direct object, indirect objection, or oblique position (POSN).

All rules for computing person, number, gender, animacy, definiteness, and syntactic position are taken from the system of Lee et al. (2011).

Table 3.3 shows each of the target ablations, as well as the SURFACE system with the DEF, AGR, and POSN features added. While the heuristic-driven feature always help over the corresponding ablated system, they cannot do the work of the fine-grained data-driven features. Most tellingly, though, none of the heuristic-driven features give statistically significant improvements on top of the data-driven features we have already included, indicating that we are at the point of diminishing returns on modeling those specific phenomena. While this does not preclude further engineering to take better advantage of other syntactic constraints, our simple features represent an "easy victory" on this subtask.

[5]Heuristic-driven approaches were historically more appropriate, since past coreference corpora such as MUC and ACE were smaller and therefore more prone to overfitting feature-rich models. However, the OntoNotes dataset contains thousands of documents, so having support for features is less of a concern.

	MUC	B^3	CEAF_e	Avg.
SURFACE	64.39	66.78	49.00	60.06
−1STWORD	63.32	66.22	47.89	59.14
+DEF−1STWORD	63.79	66.46	48.35	59.53
−PRONCONJ	59.97	63.46	47.94	57.12
+AGR−PRONCONJ	63.54	66.10	48.72	59.45
−CONTEXT	60.88	64.66	47.60	57.71
+POSN−CONTEXT	62.45	65.44	48.08	58.65
+DEF+AGR+POSN	64.55	66.93	48.94	60.14

Table 3.3: CoNLL metric scores on the development set, for the three different ablations and replacement features described in Section 3.4.2. Feature types are described in the text; + indicates inclusion of that feature class, − indicates exclusion. Each individual shallow indicator appears to do as well at capturing its target phenomenon as the hand-engineered features, while capturing other information as well. Moreover, the hand-engineered features give no benefit over the SURFACE system.

3.5 Uphill Battles on Semantics

In Section 3.4, we gave a simple set of features that yielded a high-performance coreference system; this high performance is possible because features targeting only superficial properties in a fine-grained way can actually model complex linguistic constraints. However, while our existing features capture syntactic and discourse-level phenomena surprisingly well, they are not effective at capturing semantic phenomena like type compatibility. We will show that due to structural aspects of the coreference resolution problem, even a combination of several shallow semantic features from the literature fails to adequately model semantics.

3.5.1 Analysis of the Surface System

What can the SURFACE system resolve correctly, and what errors does it still make? To answer this question, we will split mentions into several categories based on their observable properties and the gold standard coreference information, and examine our system's accuracy on each mention subclass in order to more thoroughly characterize its performance.[6] These categories represent important distinctions in terms of the difficulty of mention resolution for our system.

We first split mentions into three categories by their *status* in the gold standard: singleton (unannotated in the OntoNotes corpus), starting a new entity with at least two mentions, or

[6]This method of analysis is similar to that undertaken in Stoyanov et al. (2009) and Rahman and Ng (2011b), though we split our mentions along different axes, and can simply evaluate on accuracy because our decisions do not directly imply multiple links, as they do in binary classification-based systems (Stoyanov et al., 2009) or in entity-mention models (Rahman and Ng, 2011b).

	Nominal/Proper				Pronominal	
	1st w/head		2nd+ w/head			
Singleton	99.7%	18.1K	85.5%	7.3K	66.5%	1.7K
Starts Entity	98.7%	2.1K	78.9%	0.7K	48.5%	0.3K
Anaphoric	7.9%	0.9K	75.5%	3.9K	72.0%	4.4K

Table 3.4: Analysis of our SURFACE system on the development set. We characterize each predicted mention by its status in the gold standard (singleton, starting a new entity, or anaphoric), its type (pronominal or nominal/proper), and by whether its head has appeared as the head of a previous mention. Each cell shows our system's accuracy on that mention class as well as the size of the class. The biggest weakness of our system appears to be its inability to resolve anaphoric mentions with new heads (bottom-left cell).

anaphoric. It is important to note that while singletons and mentions starting new entities are outwardly similar in that they have no antecedents, and the prediction should be the same in either case (NEW), we treat them as distinct because the factors that impact the coreference decision differ in the two cases. Mentions that start new clusters are semantically similar to anaphoric mentions, but may be marked by heaviness or by a tendency to be named entities, whereas singletons may be generic or temporal NPs which might be thought of as coreferent in a loose sense, but are not included in the OntoNotes dataset due to choices in the annotation standard.

Second, we divide mentions by their type, pronominal versus nominal/proper; we then further subdivide nominals and propers based on whether or not the head word of the mention has appeared as the head of a previous mention in the document.

Table 3.4 shows the results of our analysis. In each cell, we show the fraction of mentions that we correctly resolve (i.e., for which we make an antecedence decision consistent with the gold standard), as well as the total number of mentions falling into that cell. First, we observe that there are a surprisingly large number of singleton mentions with misleading head matches to previous mentions (often recurring temporal nouns phrases, like *July*). The features in our system targeting anaphoricity are useful for exactly this reason: the more bad head matches we can rule out based on other criteria, the more strongly we can rely on head match to make correct linkages.

Our system is most noticeably poor at resolving anaphoric mentions whose heads have not appeared before. The fact that exact match and head match are our only recall-oriented features on nominals and propers is starkly apparent here: when we cannot rely on head match, as is true for this mention class, we only resolve 7.9% of anaphoric mentions correctly.[7] Many of the mentions in this category can only be correctly resolved by exploiting world

[7]There are an additional 346 anaphoric nominal/proper mentions in the 2nd+ category whose heads only appeared previously as part of a different cluster; we only resolve 1.7% of these extremely tricky cases correctly.

knowledge, so we will need to include features that capture this knowledge in some fashion.

3.5.2 Incorporating Shallow Semantics

As we were able to incorporate syntax with shallow features, so too might we hope to incorporate semantics. However, the semantic information contained even in a coreference corpus of thousands of documents is insufficient to generalize to unseen data,[8] so system designers have turned to external resources such as semantic classes derived from WordNet (Soon et al., 2001), WordNet hypernymy or synonymy (Stoyanov et al., 2010), semantic similarity computed from online resources (Ponzetto and Strube, 2006), named entity type features, gender and number match using the dataset of (Bergsma and Lin, 2006), and features from unsupervised clusters (Hendrickx and Daelemans, 2007). In this section, we consider the following subset of these information sources:

- WordNet hypernymy and synonymy
- Number and gender data for nominals and propers from Bergsma and Lin (2006)
- Named entity types
- Latent clusters computed from English Gigaword (Graff et al., 2007), where a latent cluster label generates each nominal head (excluding pronouns) and a conjunction of its verbal governor and semantic role, if any. We use twenty clusters, which include clusters like *president* and *leader* (things which *announce*).

Together, we call these the SEM features. We show results from this expansion of the feature set in Table 3.5. When using system mentions, the improvements are not statistically significant on every metric, and are quite marginal given that these features add information that is intuitively central to coreference and otherwise unavailable to the system. We explore the reasons behind this in the next section.

3.5.3 Analysis of Semantic Features

The main reason that weak semantic cues are not more effective is the small fraction of positive coreference links present in the training data. From Table 3.4, the number of annotated coreferent spans in the OntoNotes data is about a factor of five smaller than the number of system mentions.[9] This both means that most NPs are not coreferent, and for those that are, choosing the correct links is much more difficult because of the large number of possible antecedents. Even head match, which is generally considered a high-precision

[8] We experimented with bilexical features on head pairs, but they did not give statistically significant improvements over the SURFACE features.

[9] This observation is more general than just our system: the majority of coreference systems, including the winners of the CoNLL shared tasks (Lee et al., 2011; Fernandes et al., 2012), opt for high mention recall and resolve a relatively large number of system mentions.

	MUC	B^3	CEAF_e	Avg.
SURFACE	64.39	66.78	49.00	60.06
SURFACE+SEM	64.70	**67.27**	**49.28**	**60.42**
SURFACE (G)	82.80	74.10	68.33	75.08
SURFACE+SEM (G)	**84.49**	**75.65**	**69.89**	**76.68**

Table 3.5: CoNLL metric scores on the development set for our SEM features when added on top of our SURFACE features. We experiment on both system mentions and gold mentions. Surprisingly, despite the fact that absolute performance numbers are much higher on gold mentions and there is less room for improvement, the semantic features help much more than they do on system mentions.

indicator (Lee et al., 2011), would introduce many spurious coreference arcs if applied too liberally (see Table 3.4).

In light of this fact, a system needs very strong evidence to overcome the default hypothesis that a mention is not coreferent, and a weak indicator will have such a high "false positive" rate that it cannot be relied on (given high weight, this feature would do more harm than good, by introducing many false linkages).

To confirm this intuition, we show in the bottom part of Table 3.5 results when we apply these semantic features on top of our SURFACE system on *gold mentions*, where there are no singletons. In the gold mention setting, we see that the semantic features give a consistent improvement on every metric. Moreover, if we look at a breakdown of errors, the main improvement the semantic features give us is on resolution of anaphoric nominals with no head match: accuracy on the 1601 mentions that fall into this category improves from 28.0% to 37.9%. On predicted mentions, by contrast, this category only improves from 7.9% to 12.2%, a much smaller absolute improvement and one that comes at the expense of performance on most other resolution class. The one class that does not get worse, singleton pronouns, actually improves by a similar 4% margin, indicating that roughly half of the gains we observe are not even necessarily a result of our features doing what they were designed to do.

Our weak cues do yield some small gains, so there is hope that better weak indicators of semantic compatibility could prove more useful. However, while extremely high-precision approaches with carefully engineered features have been shown to be successful (Rahman and Ng, 2011a; Bansal and Klein, 2012; Recasens et al., 2013a), we conclude that capturing semantics in a data-driven, shallow manner remains an uphill battle.

3.6 Final System and Results

While semantic features ended up giving only marginal benefit, we have demonstrated that nevertheless our SURFACE system is a state-of-the-art English coreference system. However,

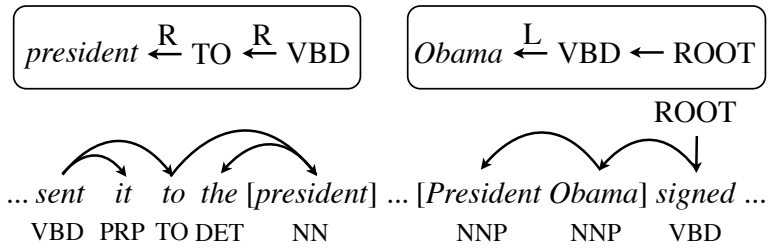


Figure 3.3: Demonstration of the ancestry extraction process. These features capture more sophisticated configurational information than our context word features do: in this example, *president* is in a characteristic indirect object position based on its dependency parents, and *Obama* is the subject of the main verb of the sentence.

there remain a few natural features that we omitted in order to keep the system as simple as possible, since they were orthogonal to the discussion of data-driven versus heuristic-driven features and do not target world knowledge. Before giving final results, we will present a small set of additional features that consider four additional mention properties beyond those in Section 3.4.1:

- Whether two mentions are nested
- Ancestry of each mention head: the dependency parent and grandparent POS tags and arc directions (shown in Figure 3.3)
- The speaker of each mention
- Number and gender of each mention as determined by Bergsma and Lin (2006)

The specific additional features we use are shown in Table 3.6. Note that unlike in Section 3.5, we use the number and gender information only on the antecedent. Due to our conjunction scheme, both this semantic information and the speaker information can apply in a fine-grained way to different pronouns, and can therefore improve pronoun resolution substantially; however, these features generally only improve pronoun resolution.

Full results for our SURFACE and FINAL feature sets are shown in Table 3.7. Again, we compare to Lee et al. (2011) and Björkelund and Farkas (2012).[10] Despite our system's emphasis on one-pass resolution with as simple a feature set as possible, we are able to outperform even these sophisticated systems by a wide margin.

[10]Discrepancies between scores here and those printed in Pradhan et al. (2012) arise from two sources: improvements to the system of Lee et al. (2011) since the first CoNLL shared task, and a fix to the scoring of B^3 in the official scorer since results of the two CoNLL shared tasks were released. Unfortunately, because of this bug in the scoring program, direct comparison to the printed results of the other highest-scoring English systems, Fernandes et al. (2012) and Martschat et al. (2012), is impossible.

Feature name	Count
Features of the SURFACE system	418704
Features on the current mention	
[ANAPHORIC] + [CURRENT ANCESTRY]	46047
Features on the antecedent	
[ANTECEDENT ANCESTRY]	53874
[ANTECEDENT GENDER]	338
[ANTECEDENT NUMBER]	290
Features on the pair	
[HEAD CONTAINED (T/F)]	136
[EXACT STRING CONTAINED (T/F)]	133
[NESTED (T/F)]	355
[DOC TYPE] + [SAME SPEAKER (T/F)]	437
[CURRENT ANCESTRY] + [ANT. ANCESTRY]	2555359

Table 3.6: FINAL feature set; note that this includes the SURFACE feature set. As with the features of the SURFACE system, two conjoined variants of each feature are included: first with the type of the current mention (NOMINAL, PROPER, or the citation form of the pronoun), then with the types of both mentions in the pair. These conjunctions allow antecedent features on gender and number to impact pronoun resolution, and they allow speaker match to capture effects like *I* and *you* being coreferent when the speakers differ.

3.7 Related Work

Many of the individual features we employ in the FINAL feature set have appeared in other coreference systems (Björkelund and Nugues, 2011; Rahman and Ng, 2011b; Fernandes et al., 2012). However, other authors have often emphasized bilexical features on head pairs, whereas our features are heavily monolexical. For feature conjunctions, other authors have exploited three classes (Lee et al., 2011) or automatically learned conjunction schemes (Fernandes et al., 2012; Lassalle and Denis, 2013), but to our knowledge we are the first to do fine-grained modeling of every pronoun. Inclusion of a hierarchy of features with regularization also means that we organically get distinctions among different mention types without having to choose a level of granularity a priori, unlike the distinct classifiers employed by Denis and Baldridge (2008).

In terms of architecture, many coreference systems operate in a pipelined fashion, making partial decisions about coreference or pruning arcs before full resolution. Some systems use separate rule-based and learning-based passes (Chen and Ng, 2012; Fernandes et al., 2012), a series of learning-based passes (Björkelund and Farkas, 2012), or referentiality classifiers that prune the set of mentions before resolution (Rahman and Ng, 2009; Björkelund and Farkas, 2012; Recasens et al., 2013b). By contrast, our system resolves all mentions in one pass and

	MUC			B^3			CEAF_e			Avg.
	Prec.	Rec.	F_1	Prec.	Rec.	F_1	Prec.	Rec.	F_1	F_1
	CoNLL 2011 Development Set									
STANFORD	61.62	59.34	60.46	74.05	58.70	65.48	45.98	48.22	47.07	57.67
IMS	66.67	58.20	62.15	77.60	56.77	65.57	42.92	51.11	46.66	58.13
SURFACE*	68.42	60.80	64.39	76.57	59.21	66.78	45.30	53.36	49.00	60.06
FINAL*	68.97	63.47	**66.10**	76.58	62.06	**68.56**	47.32	53.19	**50.09**	**61.58**
	CoNLL 2011 Test Set									
STANFORD	60.91	62.13	61.51	70.61	57.31	63.27	45.79	44.56	45.17	56.65
IMS	68.15	61.60	64.71	75.97	56.39	64.73	42.30	48.88	45.35	58.26
FINAL*	66.81	66.04	**66.43**	71.07	61.89	**66.16**	47.37	48.22	**47.79**	**60.13**

Table 3.7: CoNLL metric scores for our systems on the CoNLL development and blind test sets, compared to the results of Lee et al. (2011) (STANFORD) and Björkelund and Farkas (2012) (IMS). Starred systems are contributions of this work. Bolded F_1 values represent statistically significant improvements over other systems with $p < 0.05$ using a bootstrap resampling test. Metric values reflect version 5 of the CoNLL scorer.

does not need pruning: the SURFACE system can train in less than two hours without any subsampling of coreference arcs, and rule-based pruning of coreference arcs actually causes our system to perform less well, since our features can learn valuable information from these negative examples.

3.8 Conclusion

We have presented a coreference system that uses a simple, homogeneous set of features in a discriminative learning framework to achieve high performance. Large numbers of lexicalized, data-driven features implicitly model linguistic phenomena such as definiteness and centering, obviating the need for heuristic-driven rules explicitly targeting these same phenomena. Additional semantic features give only slight benefit beyond head match because they do not provide strong enough signals of coreference to improve performance in the system mention setting; modeling semantic similarity still requires complex outside information and deep heuristics.

While there is additional mileage to be gained by further improvements to systems of this form, Chapter 4 will attempt to bring in outside information via joint modeling of coreference, entity linking, and semantic typing. Revamping the model in this way will allow us to make inroads on precisely the cases that Table **??** suggests are most damaging to our system's overall accuracy.

Chapter 4

A Joint Model for Coreference, Semantic Typing, and Entity Linking[1]

4.1 Introduction

How do we characterize the collection of entities present in a document? Two broad threads exist in the literature. The first is coreference resolution (Soon et al., 2001; Ng, 2010; Pradhan et al., 2011), which identifies clusters of mentions in a document referring to the same entity. This process gives us access to useful information about the referents of pronouns and nominal expressions, but because clusters are local to each document, it is often hard to situate document entities in a broader context. A separate line of work has considered the problem of entity linking or "Wikification" (Cucerzan, 2007; Milne and Witten, 2008; Ji and Grishman, 2011), where mentions are linked to entries in a given knowledge base. This is useful for grounding proper entities, but in the absence of coreference gives an incomplete picture of document content itself, in that nominal expressions and pronouns are left unresolved.

In this chapter, we describe a joint model of coreference, entity linking, and semantic typing (named entity recognition) using a structured conditional random field. Variables in the model capture decisions about antecedence, semantic type, and entity links for each mention. Unary factors on these variables incorporate features that are commonly employed when solving each task in isolation. Binary and higher-order factors capture interactions between pairs of tasks. For entity linking and NER, factors capture a mapping between NER's semantic types and Wikipedia's semantics as described by infoboxes, categories, and article text. Coreference interacts with the other tasks in a more complex way, via factors that softly encourage consistency of semantic types and entity links across coreference arcs. Figure 4.1 shows an example of the effects such factors can capture. The non-locality of coreference factors make exact inference intractable, but we find that belief propagation is a suitable approximation technique and performs well.

Our joint modeling of these three tasks is motivated by their heavy interdependencies,

[1]Portions of this chapter appeared in Durrett and Klein (2014) and (Durrett et al., 2013).

Figure 4.1: Coreference can help resolve ambiguous cases of semantic types or entity links: propagating information across coreference arcs can inform us that, in this context, *Dell* is an organization and should therefore link to the article on `Dell` in Wikipedia.

which have been noted in previous work (discussed more in Section 4.7). Entity linking has been employed for coreference resolution (Ponzetto and Strube, 2006; Rahman and Ng, 2011a; Ratinov and Roth, 2012) and coreference for entity linking (Cheng and Roth, 2013) as part of pipelined systems. Past work has shown that tighter integration of coreference and entity linking is promising (Hajishirzi et al., 2013; Zheng et al., 2013); we extend these approaches and model the entire process more holistically. Named entity recognition is improved by simple coreference (Finkel et al., 2005; Ratinov and Roth, 2009) and knowledge from Wikipedia (Kazama and Torisawa, 2007; Ratinov and Roth, 2009; Nothman et al., 2013; Sil and Yates, 2013). A joint model of coreference and NER was been proposed in Haghighi and Klein (2010), but they did not use supervised data for both tasks. Technically, our model is most closely related to that of Singh et al. (2013), who handle coreference, named entity recognition, and relation extraction.[2] Our system is novel in three ways: the choice of tasks to model jointly, the fact that we maintain uncertainty about all decisions throughout inference (rather than using a greedy approach), and the feature sets we deploy for cross-task interactions.

In designing a joint model, we would like to preserve the modularity, efficiency, and structural simplicity of pipelined approaches. Our model's feature-based structure permits improvement of features specific to a particular task or to a pair of tasks. By pruning variable domains with a coarse model and using approximate inference via belief propagation, we maintain efficiency and our model is only a factor of two slower than the union of the individual models. Finally, as a structured CRF, it is conceptually no more complex than

[2]Our model could potentially be extended to handle relation extraction or mention detection, which has also been addressed in past joint modeling efforts (Daumé and Marcu, 2005; Li and Ji, 2014), but that is outside the scope of the current work.

its component models and its behavior can be understood using the same intuition.

We apply our model to two datasets, ACE 2005 and OntoNotes, with different mention standards and layers of annotation. In both settings, our joint model outperforms our independent baseline models. On ACE, we achieve state-of-the-art entity linking results, matching the performance of the system of Fahrni and Strube (2014). On OntoNotes, we match the performance of the best published coreference system (Björkelund and Kuhn, 2014) and outperform two strong NER systems (Ratinov and Roth, 2009; Passos et al., 2014).

4.2 Motivating Examples

We first present two examples to motivate our approach. Figure 4.1 shows an example of a case where coreference is beneficial for named entity recognition and entity linking. *The company* is clearly coreferent to *Dell* by virtue of the lack of other possible antecedents; this in turn indicates that *Dell* refers to the corporation rather than to Michael Dell. This effect can be captured for entity linking by a feature tying the lexical item *company* to the fact that COMPANY is in the Wikipedia infobox for `Dell`,[3] thereby helping the linker make the correct decision. This would also be important for recovering the fact that the mention *the company* links to `Dell`; however, in the version of the task we consider, a mention like *the company* actually links to the Wikipedia article for `Company`.[4]

Figure 4.2 shows a different example, one where the coreference is now ambiguous but entity linking is transparent. In this case, an NER system based on surface statistics alone would likely predict that *Freddie Mac* is a PERSON. However, the Wikipedia article for Freddie Mac is unambiguous, which allows us to fix this error. The pronoun *his* can then be correctly resolved.

These examples justify why these tasks should be handled jointly: there is no obvious pipeline order for a system designer who cares about the performance of the model on all three tasks.

4.3 Model

Our model is a structured conditional random field (Lafferty et al., 2001). The input (conditioning context) is the text of a document, automatic parses, and a set of pre-extracted mentions (spans of text). Mentions are allowed to overlap or nest: our model makes no structural assumptions here, and in fact we will show results on datasets with two different mention annotation standards (see Section 4.6.1 and Section 4.6.3).

[3] Monospaced fonts indicate titles of Wikipedia articles.

[4] This decision was largely driven by a need to match the ACE linking annotations provided by Bentivogli et al. (2010).

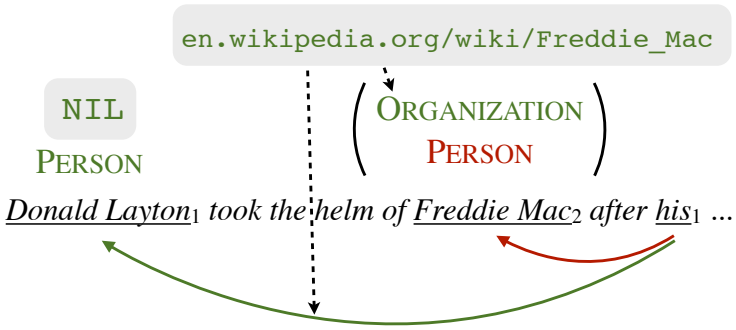


Figure 4.2: Entity links can help resolve ambiguous cases of coreference and entity types. Standard NER and coreference systems might fail to handle *Freddie Mac* correctly, but incorporating semantic information from Wikipedia makes this decision easier.

Figure 4.3 shows the random variables in our model. We are trying to predict three distinct types of annotation, so we naturally have one variable per annotation type per mention (of which there are n):

- Coreference variables $\mathbf{a} = (a_1, \ldots, a_n)$ which indicate antecedents: $a_i \in \{1, \ldots, i-1, \text{NEW}\}$, indicating that the mention refers to some previous mention or that it begins a new cluster. These are identical to the random variables used in Chapter 3.
- Named entity type variables $\mathbf{t} = (t_1, \ldots, t_n)$ which take values in a fixed inventory of semantic types.[5]
- Entity link variables $\mathbf{e} = (e_1, \ldots, e_n)$ which take values in the set of all Wikipedia titles.

In addition we have variables $\mathbf{q} = (q_1, \ldots, q_n)$ which represent queries to Wikipedia. These are explained further in Section 4.3.1; for now, it suffices to remark that they are unobserved during both training and testing.

We place a log-linear probability distribution over these variables as follows:

$$p(\mathbf{a}, \mathbf{t}, \mathbf{e} | x; \theta) \propto \sum_{\mathbf{q}} \exp\left(\theta^\top f(\mathbf{a}, \mathbf{t}, \mathbf{e}, \mathbf{q}, x)\right)$$

where θ is a weight vector, f is a feature function, and x indicates the document text, automatic parses, and mention boundaries.

We represent the features in this model with standard factor graph notation; features over a particular set of output variables (and x) are associated with factors connected to

[5]For the next few sections, we assume a fixed-mention version of the NER task, which looks like multi-way classification of semantic types. In Section 4.6.3, we adapt the model to the standard non-fixed-mention setting for OntoNotes.

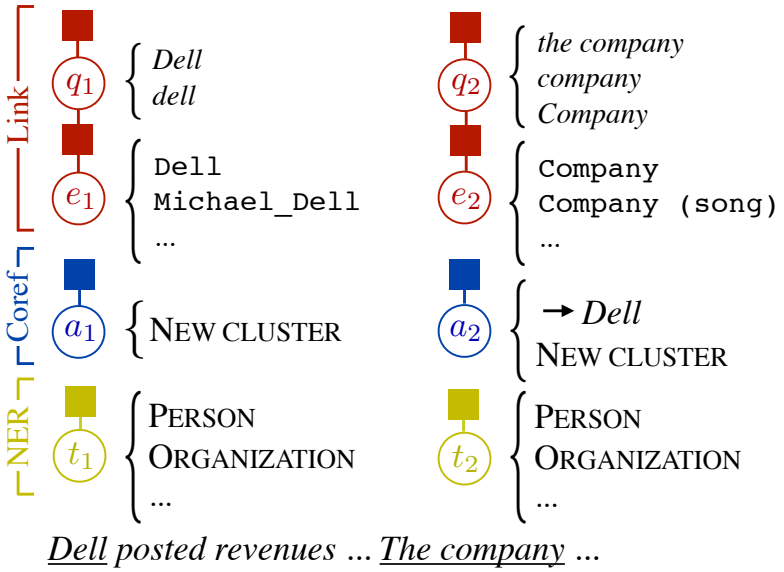


Figure 4.3: Random variables and task-specific factors present in our model. The a_i model coreference antecedents, the t_i model semantic types, the e_i model entity links, and the q_i are latent Wikipedia queries. Factors shown for each task integrate baseline features used when that task is handled in isolation. Factors are described in Section 4.3.1.

those variables. Figure 4.3 shows the task-specific factors in the model, discussed next in Section 4.3.1. Higher-order factors coupling variables between tasks are discussed in Section 4.3.2.

4.3.1 Independent Model

Figure 4.3 shows a version of the model with only task-specific factors. Though this framework is structurally simple, it is nevertheless powerful enough for us to implement high-performing models for each task. State-of-the-art approaches to coreference (see Chapter 3) and entity linking (Ratinov et al., 2011) already have this independent structure and Ratinov and Roth (2009) note that it is a reasonable assumption to make for NER as well.[6] In this section, we describe the features present in the task-specific factors of each type (which also serve as our three separate baseline systems).

[6]Pairwise potentials in sequence-based NER are useful for producing coherent output (e.g. prohibiting configurations like O I-PER), but since we have so far defined the task as operating over fixed mentions, this structural constraint does not come into play for our system.

Coreference

Our modeling of the coreference output space (as antecedents chosen for each mention) follows the setup we established in Chapter 3, namely the mention-ranking approach to coreference (Denis and Baldridge, 2008). Our feature set is the same as presented in Chapter 3, targeting surface properties of mentions: for each mention, we examine the first word, head word, last word, context words, the mention's length, and whether the mention is nominal, proper or pronominal. Anaphoricity features examine each of these properties in turn; coreference features conjoin various properties between mention pairs and also use properties of the mention pair itself, such as the distance between the mentions and whether their heads match. Note that this baseline does not rely on having access to named entity chunks.

Named Entity Recognition

Our NER model places a distribution over possible semantic types for each mention, which corresponds to a fixed span of the input text. We define the features of a span to be the concatenation of standard NER surface features associated with each token in that chunk. We use surface token features similar to those from previous work (Zhang and Johnson, 2003; Ratinov and Roth, 2009; Passos et al., 2014): for tokens at offsets of $\{-2, -1, 0, 1, 2\}$ from the current token, we fire features based on 1) word identity, 2) POS tag, 3) word class (based on capitalization, presence of numbers, suffixes, etc.), 4) word shape (based on the pattern of uppercase and lowercase letters, digits, and punctuation), 5) Brown cluster prefixes of length 4, 6, 10, 20 using the clusters from Koo et al. (2008), and 6) common bigrams of word shape and word identity.

Entity Linking

Our entity linking system diverges more substantially from past work than the coreference or NER systems. Most entity linking systems operate in two distinct phases (Cucerzan, 2007; Milne and Witten, 2008; Dredze et al., 2010; Ratinov et al., 2011). First, in the candidate generation phase, a system generates a ranked set of possible candidates for a given mention by querying Wikipedia. The standard approach for doing so is to collect all hyperlinks in Wikipedia and associate each hyperlinked span of text (e.g. *Michael Jordan*) with a distribution over titles of Wikipedia articles it is observed to link to (`Michael_Jordan`, `Michael_I._Jordan`, etc.). Second, in the disambiguation phase, a learned model selects the correct candidate from the set of possibilities.

As noted by Hachey et al. (2013) and Guo et al. (2013), candidate generation is often overlooked and yet accounts for large gaps in performance between different systems. It is not always clear how to best turn the text of a mention into a query for our set of hyperlinks. For example, the phrase *Chief Executive Michael Dell* has never been hyperlinked on Wikipedia. If we query the substring *Michael Dell*, the highest-ranked title is correct; however, querying the substring *Dell* returns the article on the company.

Our model for entity linking therefore includes both predictions of final Wikipedia titles e_i as well as latent query variables q_i that model the choice of query. Given a mention, possible queries are all prefixes of the mention containing the head with optional truecasing or lemmatization applied. Unary factors on the q_i model the appropriateness of a query based on surface text of the mention, investigating the following properties: whether the mention is proper or nominal, whether the query employed truecasing or lemmatization, the query's length, the POS tag sequence within the query and the tag immediately preceding it, and whether the query is the longest query to yield a nonempty set of candidates for the mention. This part of the model can learn, for example, that queries based on lemmatized proper names are bad, whereas queries based on lemmatized common nouns are good.

Our set of candidates links for a mention is the set of all titles produced by some query. The binary factors connecting q_i and e_i then decide which title a given query should yield. These include: the rank of the article title among all possible titles returned by that query (sorted by relative frequency count), whether the title is a close string match of the query, and whether the title matches the query up to a parenthetical (e.g. *Paul Allen* and `Paul_Allen_(editor)`).

We could also at this point add factors between pairs of variables (e_i, e_j) to capture coherence between choices of linked entities. Integration with the rest of the model, learning, and inference would remain unchanged. However, while such features have been employed in past entity linking systems (Ratinov et al., 2011; Hoffart et al., 2011), Ratinov et al. found them to be of limited utility, so we omit them from the present work.

4.3.2 Cross-task Interaction Factors

We now add factors that tie the predictions of multiple output variables in a feature-based way. Figure 4.4 shows the general structure of these factors. Each couples variables from one pair of tasks.

Entity Linking and NER

We want to exploit the semantic information in Wikipedia for better semantic typing of mentions. We also want to use semantic types to disambiguate tricky Wikipedia links. We use three sources of semantics from Wikipedia (Kazama and Torisawa, 2007; Nothman et al., 2013):

- Categories (e.g. `American financiers`); used by Ponzetto and Strube (2006), Kazama and Torisawa (2007), and Ratinov and Roth (2012)
- Infobox type (e.g. `Person`, `Company`)
- Copula in the first sentence (*is a British politician*); used for coreference previously in Haghighi and Klein (2009)

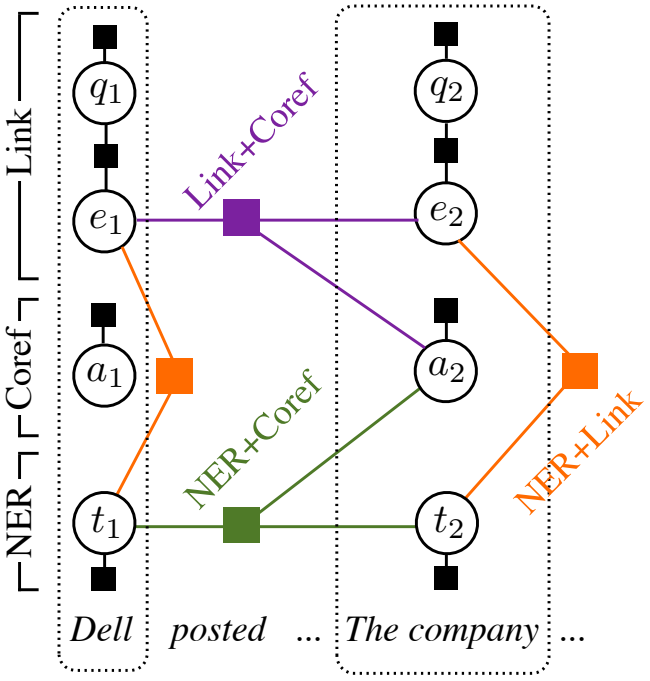


Figure 4.4: Factors that tie predictions between variables across tasks. Joint NER and entity linking factors (Section 4.3.2) tie semantic information from Wikipedia articles to semantic type predictions. Joint coreference and NER factors (Section 4.3.2) couple type decisions between mentions, encouraging consistent type assignments within an entity. Joint coreference and entity linking factors (Section 4.3.2) encourage relatedness between articles linked from coreferent mentions.

We fire features that conjoin the information from the selected Wikipedia article with the selected NER type. Because these types of information from Wikipedia are of a moderate granularity, we should be able to learn a mapping between them and NER types and exploit Wikipedia as a soft gazetteer.

Coreference and NER

Coreference can improve NER by ensuring consistent semantic type predictions across coreferent mentions; likewise, NER can help coreference by encouraging the system to link up mentions of the same type. Our factor structure is as follows:

$$\log F_{i-j}(a_i, t_i, t_j) = \begin{cases} 0 & \text{if } a_i \neq j \\ f(i, j, t_i, t_j) & \text{otherwise} \end{cases}$$

That is, the features between the type variables for mentions i and j does not come into play unless i and j are coreferent. Note that there are quadratically many such factors in

the graph (before pruning; see Section 4.5), one for each ordered pair of mentions (j, i) with $j < i$. When scoring a particular configuration of variables, only a small subset of the factors is active, but during inference when we marginalize over all settings of variables, each of the factors comes into play for some configuration. This model structure allows us to maintain uncertainty about coreference decisions but still propagate information along coreference arcs in a soft way.

Given this factor definition, we define features that should fire over *coreferent* pairs of entity types. Our features target:

- The pair of semantic types for the current and antecedent mention
- The semantic type of the current mention and the head of the antecedent mention, and the type of the antecedent and head of the current

We found such monolexical features to improve over just type pairs and while not suffering from the sparsity problems of bilexical features.

Coreference and Entity Linking

As we said in Section 4.2, coreferent mentions can actually have different entity links (e.g. `Dell` and `Company`), so encouraging equality alone is less effective for entity linking than it is for NER. Our factors have the same structure as those for coreference-NER, but features now target overall semantic relatedness of Wikipedia articles using the structure of Wikipedia by computing whether the articles have the same title, share any out links, or link to each other. More complex relatedness schemes such as those described in Ratinov et al. (2011) can be implemented in this framework. Nevertheless, these basic features still promise to help identify related articles as well as name variations by exploiting the abundance of entity mentions on Wikipedia.

4.4 Learning

Our training data consists of d documents, where a given document consists of a tuple $(x, C^*, \mathbf{t}^*, \mathbf{e}^*)$. Gold-standard labels for types ($\mathbf{t}^*$) and entity links ($\mathbf{e}^*$) are provided directly, while supervision for coreference is provided in the form of a clustering C^*. Regardless, we can simply marginalize over the uncertainty about $\mathbf{a}^*$ and form the conditional log-likelihood of the training labels as follows:

$$\mathcal{L}(\theta) = \sum_{i=1}^{d} \log \sum_{\mathbf{a}^* \in \mathcal{A}(C_i^*)} p(\mathbf{a}^*, \mathbf{t}_\mathbf{i}^*, \mathbf{e}_\mathbf{i}^* | x; \theta)$$

where $\mathcal{A}(C^*)$ is the set of antecedent structures consistent with the gold annotation: the first mention in a cluster must pick the NEW label and subsequent mentions must pick an

antecedent from the set of those preceding them in the cluster. This is the same marginalization over latent structure that we employed in Chapter 3 and which has been employed in prior work as well (Fernandes et al., 2012).

We adapt this objective to exploit parameterized loss functions for each task by modifying the distribution as follows:

$$p'(\mathbf{a}, \mathbf{t}, \mathbf{e}|x; \theta) \propto p(\mathbf{a}, \mathbf{t}, \mathbf{e}, x) \exp\left[\alpha_c \ell_c(\mathbf{a}, C^*) + \alpha_t \ell_t(\mathbf{t}, \mathbf{t}^*) + \alpha_e \ell_e(\mathbf{e}, \mathbf{e}^*)\right]$$

where ℓ_c, ℓ_t, and ℓ_e are task-specific loss functions with weight parameters α. This technique, softmax-margin, allows us to shape the distribution learned by the model and encourage the model to move probability mass away from outputs that are bad according to our loss functions (Gimpel and Smith, 2010). As in Chapter 3, we take $\alpha_c = 1$ and use ℓ_c as defined there, penalizing the model by $\alpha_{\text{FA}} = 0.1$ for linking up a mention that should have been nonanaphoric, by $\alpha_{\text{FN}} = 3$ for calling nonanaphoric a mention that should have an antecedent, and by $\alpha_{\text{WL}} = 1$ for picking the wrong antecedent for an anaphoric mention. ℓ_t and ℓ_e are simply Hamming distance, with $\alpha_t = 3$ and $\alpha_e = 0$ for all experiments. We found that the outcome of learning was not particularly sensitive to these parameters.[7]

We optimize our objective using AdaGrad (Duchi et al., 2011) with L_1 regularization and $\lambda = 0.001$. Our final objective is

$$\mathcal{L}(\theta) = \sum_{i=1}^{d} \log \sum_{\mathbf{a}^* \in \mathcal{A}(C_i^*)} p'(\mathbf{a}^*, \mathbf{t}_\mathbf{i}^*, \mathbf{e}_\mathbf{i}^*|x; \theta) + \lambda \|\theta\|_1$$

This objective is nonconvex, but in practice we have found that it is very stable. One reason is that for any mention that has fewer than two antecedents in its cluster, all elements of $\mathcal{A}(C^*)$ only contain one possibility for that mention, and even for mentions with ambiguity, the parameters that the model ends up learning tend to place almost all of the probability mass consistently on one antecedent.

4.5 Inference

For both learning and decoding, inference consists of computing marginals for individual variables or for sets of variables adjacent to a factor. Exact inference is intractabile due to our factor graph's loopiness; however, we can still perform efficient inference using belief propagation, which has been successfully employed for other NLP tasks (Smith and Eisner, 2008; Burkett and Klein, 2012). Marginals typically converge in 3-5 iterations of belief propagation; we use 5 iterations for all experiments.

However, belief propagation would still be quite computationally expensive if run on the full factor graph as described in Section 5.2. In particular, the factors in Section 4.3.2

[7] These parameters allow us to trade off contributions to the objective from the different tasks, addressing Singh et al. (2013)'s objection to single objectives for joint models.

and Section 4.3.2 are costly to sum over due to their ternary structure and the fact that there are quadratically many of them in the number of mentions. The solution to this is to prune the domains of the coreference variables using a coarse model consisting of the coreference factors trained in isolation. Given marginals $p_0(a_i|x)$, we prune values a_i such that $\log p_0(a_i|x) < \log p_0(a_i^*|x) - k$ for a threshold parameter k, which we set to 5 for our experiments; this is sufficient to prune over 90% of possible coreference arcs while leaving at least one possible gold link for 98% of mentions.[8] With this optimization, our full joint model could be trained for 20 iterations on the ACE 2005 corpus in around an hour.

We use minimum Bayes risk (MBR) decoding, where we compute marginals for each variable under the full model and independently return the most likely setting of each variable. Note that for coreference, this implies that we produce the MBR antecedent structure rather than the MBR clustering; the latter is much more computationally difficult to find and would be largely the same, since the posterior distributions of the a_i are quite peaked.

4.6 Experiments

We present results on two corpora. First, we use the ACE 2005 corpus (NIST, 2005): this corpus annotates mentions complete with coreference, semantic types (per mention), and entity links (also per mention) later added by Bentivogli et al. (2010). We evaluate on gold mentions in this setting for comparability with prior work on entity linking; we lift this restriction in Section 4.6.3.

Second, we evaluate on the OntoNotes 5 corpus (Hovy et al., 2006) as used in the CoNLL 2012 coreference shared task (Pradhan et al., 2012). This corpus does not contain gold-standard entity links, so we cannot evaluate this portion of our model, though the model still exploits the information from Wikipedia to make coreference and named entity decisions. We will compare to prior coreference and named entity work in the system mentions setting.

4.6.1 ACE Evaluation

We tokenize and sentence-split the ACE dataset using the tools bundled with Reconcile (Stoyanov et al., 2010) and parse it using the Berkeley Parser (Petrov et al., 2006). We use the train/test split from Stoyanov et al. (2009), Haghighi and Klein (2010), and Bansal and Klein (2012).

Table 4.1 shows our results. Coreference results are reported using MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), and CEAF_e (Luo, 2005), as well as their average, the CoNLL metric, all computed from the reference implementation of the CoNLL scorer (Pradhan et al., 2014). We see that the joint model improves all three tasks compared to the individual task models in the baseline.

[8] In addition to inferential benefits, pruning an arc allows us to prune entire joint coreference factors and avoid instantiating their associated features, which reduces the memory footprint and time needed to build a factor graph.

	Dev						Test					
	MUC	B^3	CEAF_e	Avg.	NER	Link	MUC	B^3	CEAF_e	Avg.	NER	Link
INDEP.	77.95	74.81	71.84	74.87	83.04	73.07	81.03	74.89	72.56	76.16	82.35	74.71
JOINT	**79.41**	**75.56**	**73.34**	**76.10**	**85.94**	**75.69**	81.41	74.70	72.93	76.35	**85.60**	**76.78**
Δ	+1.46	+0.75	+1.50	+1.23	+2.90	+2.62	+0.42	−0.19	+0.37	+0.19	+3.25	+2.07

Table 4.1: Results on the ACE 2005 dev and test sets for the INDEP. (task-specific factors only) and JOINT models. Coreference metrics are computed using their reference implementations (Pradhan et al., 2014). We report accuracy on NER because the set of mentions is fixed and all mentions have named entity types. Coreference and NER are compared to prior work in a more standard setting in Section 4.6.3. Finally, we also report accuracy of our entity linker (including links to NIL); entity linking is analyzed more thoroughly in Table 4.2. Bolded values represent statistically significant improvements with $p < 0.05$ according to a bootstrap resampling test.

	Non-NILS			NILS			
	Prec.	Rec.	F_1	Prec.	Rec.	F_1	Accuracy
FAHRNI	81.15	78.10	79.60	41.25	61.10	49.25	76.87
INDEP.	80.26	76.30	78.23	33.39	54.47	41.40	74.71
JOINT	83.26	77.67	80.37	35.19	65.42	45.77	76.78
Δ over INDEP.	+3.00	+1.37	+2.14	+1.80	+10.95	+3.37	+2.07

Table 4.2: Detailed entity linking results on the ACE 2005 test set. We evaluate both our INDEP. (task-specific factors only) and JOINT models and compare to the results of the FAHRNI model, a state-of-the-art entity linking system. We compare overall accuracy as well as performance at predicting NILS (mentions not in the knowledge base) and non-NILS. The JOINT model roughly matches the performance of FAHRNI and gives strong gains over the INDEP. system.

More in-depth entity linking results are shown in Table 4.2. We both evaluate on overall accuracy (how many mentions are correctly linked) as well as two more specific criteria: precision/recall/F_1 of non-NIL[9] predictions, and precision/recall/F_1 of NIL predictions. This latter measure may be important if a system designer is trying to identify new entities in a document. We compare to the results of the best model from Fahrni and Strube (2014), which is a sophisticated discriminative model incorporating a latent model of mention scope.[10]

Our performance is similar to that of Fahrni and Strube (2014), though the results are not exactly comparable for two reasons. First, our models are trained on different datasets: Fahrni and Strube (2014) train on Wikipedia data whereas we train on the ACE training set.

[9] NIL is a placeholder for mentions which do not link to an article in Wikipedia.

[10] On the TAC datasets, this FAHRNI model substantially outperforms Ratinov et al. (2011) and has comparable performance to Cheng and Roth (2013), hence it is quite competitive.

	Coref	NER	Link
INDEP.	74.87	83.04	73.07
INDEP+LINKNER		+1.85	+2.41
INDEP+COREFNER	+0.56	+1.15	
INDEP+COREFLINK	+0.48		−0.16
JOINT−LINKNER	+0.79	+1.28	−0.06
JOINT−COREFNER	+0.56	+1.94	+2.07
JOINT−COREFLINK	+0.85	+2.68	+2.57
JOINT	+1.23	+2.90	+2.62
JOINT/LATENTLINK	+1.26	+3.47	−18.8

Table 4.3: Results of model ablations on the ACE development set. We hold out each type of factor in turn from the JOINT model and add each in turn over the INDEP. model. We evaluate the coreference performance using the CoNLL metric, NER accuracy, and entity linking accuracy.

Second, they make use of the annotated head spans in ACE whereas we only use detected heads based on automatic parses. Note that this information is particularly beneficial for locating the right query because "heads" may be multi-word expressions such as *West Bank* as part of the phrase *southern West Bank*.

4.6.2 Model Ablations

To evaluate the importance of the different parts of the model, we perform a series of ablations on the model interaction factors. Table 4.3 shows the results of adding each interaction factor in turn to the baseline and removing each of the three interaction factors from the full joint model (see Figure 4.4).

Link–NER interactions. These joint factors are the strongest out of any considered here and give large improvements to entity linking and NER. Their utility is unsurprising: effectively, they give NER access to a gazetteer that it did not have in the baseline model. Moreover, our relatively rich featurization of the semantic information on Wikipedia allows the model to make effective use of it.

Coref–NER interactions. These are moderately beneficial to both coreference and NER. Having reliable semantic types allows the coreference system to be bolder about linking up mention pairs that do not exhibit direct head matches. Part of this is due to our use of monolexical features, which are fine-grained but still effectively learnable.

Coref–Link interactions. These are the least useful of any of the major factors, providing only a small benefit to coreference. This is likely a result of the ACE entity linking annotation

standard: a mention like *the company* is not linked to the specific company it refers to, but instead the Wikipedia article `Company`. Determining the relatedness of `Company` to an article like `Dell` is surprisingly difficult: many related articles share almost no out-links and may not explicitly link to one another. Further feature engineering could likely improve the utility of these factors.

The last line of Table 4.3 shows the results of an experiment where the entity links were not observed during training, i.e. they were left latent. Unsurprisingly, the system is not good at entity linking; however, the model is still able to do as well or even slightly better on coreference and named entity recognition. A possible explanation for this is that even the wrong Wikipedia link can in many cases provide correct semantic information: for example, not knowing which *Donald Layton* is being referred to is irrelevant for the question of determining that he is a PERSON and may also have little impact on coreference performance. This result indicates that the joint modeling approach is not necessarily dependent on having all tasks annotated. The model can make use of cross-task information even when that information comes via latent variables.

4.6.3 OntoNotes Evaluation

The second part of our evaluation uses the datasets from the CoNLL 2012 Shared Task (Pradhan et al., 2012), specifically the coreference and NER annotations. All experiments use the standard automatic parses from the shared task and mentions detected according to the method in Chapter 3.

Evaluating on OntoNotes carries with it a few complications. First, gold-standard entity linking annotations are not available; we can handle this by leaving the e_i variables in our model latent. Second, and more seriously, NER chunks are no longer the same as coreference mentions, so our assumption of fixed NER spans no longer holds.

Divergent Coreference and NER

Our model can be adapted to handle NER chunks that diverge from mentions for the other two tasks, as shown in Figure 4.5. We have kept the coreference and entity linking portions of our model the same, now defined over system predicted mentions. However, we have replaced mention-synchronous type variables with standard token-synchronous BIO-valued variables. The unary NER features developed in Section 4.3.1 are now applied in the standard way, namely they are conjoined with the BIO labels at each token position. Binary factors between adjacent NER nodes enforce appropriate structural constraints and fire indicator features on transitions. In order to maintain tractability in the face of a larger number of variables and factors in the NER portion of our model, we prune the NER variables' domains using the NER model trained in isolation, similar to the procedure that we described for pruning coreference arcs in Section 4.5.

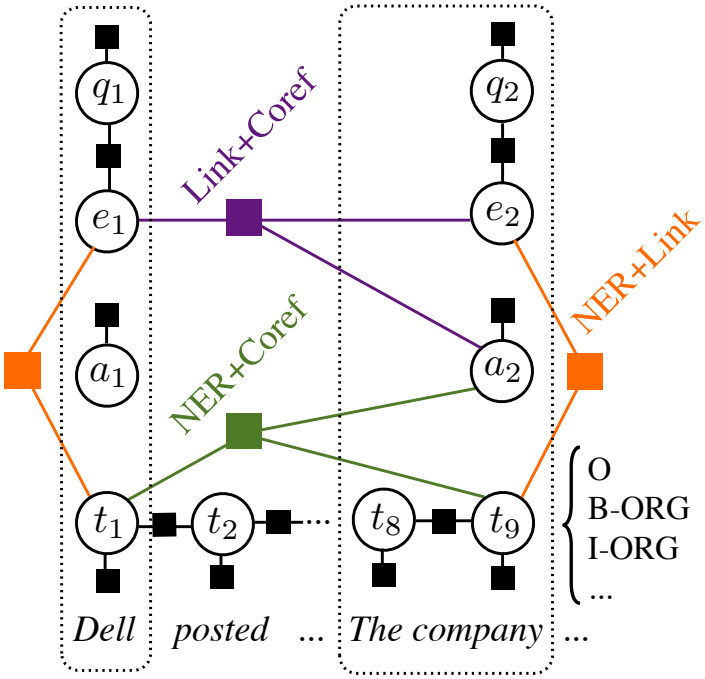


Figure 4.5: Modified factor graph for OntoNotes-style annotations, where NER chunks can now diverge from mentions for the other two tasks. NER is now modeled with token-synchronous random variables taking values in a BIO tagset. Factors coupling NER and the other tasks now interact with the NER chain via the NER nodes associated with the heads of mentions.

Cross-task factors that previously would have fired features based on the NE type for a whole mention now instead consult the NE type of that mention's head.[11] In Figure 4.5, this can be seen with factors involving e_2 and a_2 touching t_9 (*company*), the head of the second mention. Since the chain structure enforces consistency between adjacent labels, features that strongly prefer a particular label on one node of a mention will implicitly affect other nodes in that mention and beyond.

Training and inference proceed as before, with a slight modification: instead of computing the MBR setting of every variable in isolation, we instead compute the MBR sequence of labeled NER chunks to avoid the problem of producing inconsistent tag sequences, e.g. O I-PER or B-PER I-ORG.

[11] The NER-coreference portion of the model now resembles the skip-chain CRF from Finkel et al. (2005), though with soft coreference.

	MUC			B^3			CEAF_e			Avg.
	Prec.	Rec.	F_1	Prec.	Rec.	F_1	Prec.	Rec.	F_1	F_1
BERKELEY	72.85	65.87	69.18	63.55	52.47	57.48	54.31	54.36	54.34	60.33
FERNANDES	–	–	70.51	–	–	57.58	–	–	53.86	60.65
BJORKELUND	74.30	67.46	70.72	62.71	54.96	58.58	59.40	52.27	**55.61**	61.63
INDEP.	72.25	69.30	70.75	60.92	55.73	58.21	55.33	54.14	54.73	61.23
JOINT	72.61	69.91	**71.24**	61.18	56.43	**58.71**	56.17	54.23	55.18	**61.71**

Table 4.4: CoNLL metric scores for our systems on the CoNLL 2012 blind test set, compared to our system from Chapter 3 (BERKELEY), Fernandes et al. (2012) (the winner of the CoNLL shared task), and Björkelund and Kuhn (2014) (the best reported results on the dataset to date). INDEP. and JOINT are the contributions of this chapter; JOINT improves substantially over INDEP. (these improvements are statistically significant with $p < 0.05$ according to a bootstrap resampling test) and achieves state-of-the-art results.

	Prec.	Rec.	F_1
ILLINOIS	82.00	**84.95**	83.45
PASSOS	–	–	82.24
INDEP.	83.79	81.53	82.64
JOINT	**85.22**	82.89	**84.04**
Δ over INDEP.	+1.43	+1.36	+1.40

Table 4.5: Results for NER tagging on the OntoNotes 5.0 / CoNLL 2011 test set. We compare our systems to the Illinois system (Ratinov and Roth, 2009) and the system of Passos et al. (2014). Our model outperforms both other systems in terms of F_1, and once again joint modeling gives substantial improvements over our baseline system.

Results

Table 4.4 shows coreference results from our INDEP. and JOINT models compared to three strong systems: our model from Chapter 3, Fernandes et al. (2012) (the winner of the CoNLL shared task), and Björkelund and Kuhn (2014) (the best reported results on the dataset). Our JOINT method outperforms all three as well as the INDEP. system.[12]

Next, we report results on named entity recognition. We use the same OntoNotes splits as for the coreference data; however, the New Testament (NT) portion of the CoNLL 2012 test set does not have gold-standard named entity annotations, so we omit it from our evaluation. This leaves us with exactly the CoNLL 2011 test set. We compare to two existing baselines from the literature: the Illinois NER system of Ratinov and Roth (2009) and the results of

[12]The systems of Chang et al. (2013) and Webster and Curran (2014) perform similarly to the FERNANDES system; changes in the reference implementation of the metrics make exact comparison to printed numbers difficult.

	Hard case accuracy
INDEP.	48.2
JOINT	57.3

Table 4.6: Comparison of our system variants on "hard cases" of coreference that typically require semantic or world knowledge to address. These hard cases are defined as anaphoric nominal or proper mentions for which no antecedent in the gold coreference cluster has the same head as them. We see a 9% absolute improvement from the joint model, a promising improvement suggesting that integration of knowledge sources is a way to make progress on these cases.

Passos et al. (2014). Table 4.5 shows that we outperform both prior systems in terms of F_1, though the ILLINOIS system features higher recall while our system features higher precision.

Analysis

One goal of modeling coreference jointly with the other two tasks was to see if we could make progress on cases that require world knowledge or knowledge of semantics, the "uphill battles" in Chapter 3. We can revisit this by analyzing the performance of our JOINT model on cases that seem especially likely to require this information. Recall that we made by far the most errors on anaphoric nominal and proper mentions for which their head did not previously appear in the document (see Table 3.4).

Table 4.6 reports accuracy rates on anaphoric nominal or proper mentions which do not have a head match with any of their antecedents.[13] These are typically the most difficult cases to recall, since in the absence of strong cues like head match, a system will typically make the conservative prediction that a mention is non-anaphoric. We see that the JOINT system achieves a 9% absolute improvement over the INDEP. system, a much larger improvement on this type of mention than would be suggested by the macro improvements in Table 4.4. These results suggest that integrating knowledge is useful for improving results on these cases of coreference.

4.7 Related Work

There are two closely related threads of prior work: those that address the tasks we consider in a different way and those that propose joint models for other related sets of tasks. In the first category, Hajishirzi et al. (2013) integrate entity linking into a sieve-based coreference system (Raghunathan et al., 2010), the aim being to propagate link decisions throughout coreference chains, block coreference links between different entities, and use semantic information to make additional coreference links. Zheng et al. (2013) build coreference clusters

[13]Note that this is a slightly different and more relaxed criterion than that in Table 3.4.

greedily left-to-right and maintain entity link information for each cluster, namely a list of possible targets in the knowledge base as well as a current best link target that is used to extract features (though that might not be the target that is chosen by the end of inference). Cheng and Roth (2013) use coreference as a preprocessing step for entity linking and then solve an ILP to determine the optimal entity link assignments for each mention based on surface properties of that mention, other mentions in its cluster, and other mentions that it is related to. Compared to these systems, our approach maintains greater uncertainty about all random variables throughout inference and uses features to capture cross-task interactions as opposed to rules or hard constraints, which can be less effective for incorporating semantic knowledge (Lee et al., 2011).

The joint model most closely related to ours is that of Singh et al. (2013), modeling coreference, named entity recognition, and relation extraction. Their techniques differ from ours in a few notable ways: they choose a different objective function than we do and also opt to freeze the values of certain variables during the belief propagation process rather than pruning with a coarse pass. Sil and Yates (2013) jointly model NER and entity linking in such a way that they maintain uncertainty over mention boundaries, allowing information from Wikipedia to inform segmentation choices. We could strengthen our model by integrating this capability; however, the primary cause of errors for mention detection on OntoNotes is parsing ambiguities rather than named entity ambiguities, so we would be unlikely to see improvements in the experiments presented here. Beyond maintaining uncertainty over mention boundaries, we might also consider maintaining uncertainty over the entire parse structure, as in Finkel and Manning (2009), who consider parsing and named entity recognition together with a PCFG.

4.8 Additional Entity Properties

We have seen that propagating coarse semantic type information around can be useful for coreference. This suggests that we should explore additional semantic properties as features in this framework as well. Specifically, we both look at ϕ-features (number, gender, and animacy) as well as latent semantic types induced from automatic clustering.

Since we are not necessarily interested in modeling the values of these semantic properties jointly, we could potentially design a pipelined model for this task (Luo et al., 2004; Rahman and Ng, 2009). However, inference in this model still requires reasoning about all possible partitions of mentions, which is computationally infeasible without resorting to severe approximations like a left-to-right inference method (Rahman and Ng, 2009). Instead, we will simply adapt our joint model of coreference and semantic typing to this purpose. As with semantic types, we introduce one auxiliary variable per mention corresponding to each semantic property of interest. Then, we will require each anaphoric mention to agree with its antecedent on the value of each of these properties.

Our Transitive model which implements this scheme is shown in Figure 4.6. Each mention i has been augmented with a single property node $p_i \in \{1, ..., k\}$. The unary P_i

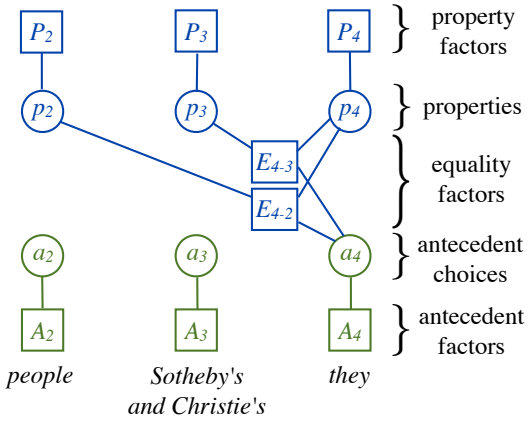


Figure 4.6: The factor graph for our TRANSITIVE coreference model. Each node a_i now has a property p_i, which is informed by its own unary factor P_i. In our example, a_4 strongly indicates that mentions 2 and 4 are coreferent; the factor E_{4-2} then enforces equality between p_2 and p_4, while the factor E_{4-3} has no effect.

factors encode prior knowledge about the setting of each p_i; these factors may be hard (I will not refer to a plural entity), soft (such as a distribution over latent cluster types), or practically uniform (e.g. the last name Smith does not specify a particular gender).

To enforce agreement of a particular property, we diverge from Section 4.3.2: rather than enforcing constraints softly with features, we instead require a mention to have the same property value as its antecedent, but introduce an additional set of variables and factors that allow property values to deviate from the prior in a regular way. That is, for mentions i and j, if $a_i = j$, we want to ensure that p_i and p_j agree. We can achieve this with the following set of structural equality factors:

$$E_{i-j}(a_i, p_i, p_j) = 1 - \mathbb{I}[a_i = j \wedge p_i \neq p_j]$$

In words, this factor is zero if both $a_i = j$ and p_i disagrees with p_j. These equality factors essentially provide a mechanism by which these priors P_i can influence the coreference decisions: if, for example, the factors P_i and P_j disagree very strongly, choosing $a_i \neq j$ will be preferred in order to avoid forcing one of p_i or p_j to take an undesirable value. Moreover, note that although a_i only indicates a single antecedent, the transitive nature of the F factors forces p_i to agree with the p nodes of all other mentions likely to be in the same entity.

Taking this process one step further, we can introduce softness into the properties and learn a regular model of property mutation on a per-mention basis using an additional set of binary factors. This new scheme is shown in Figure 4.7. As before, we have a set of properties p_i and agreement factors E_{ij}. On top of that, we introduce the notion of raw property values $r_i \in \{1, ..., k\}$ together with priors in the form of the R_i factors. The r_i and p_i could in principle have different domains, but for this work we take them to have the same domain. The P_i factors now have a new structure: they now represent a featurized projection

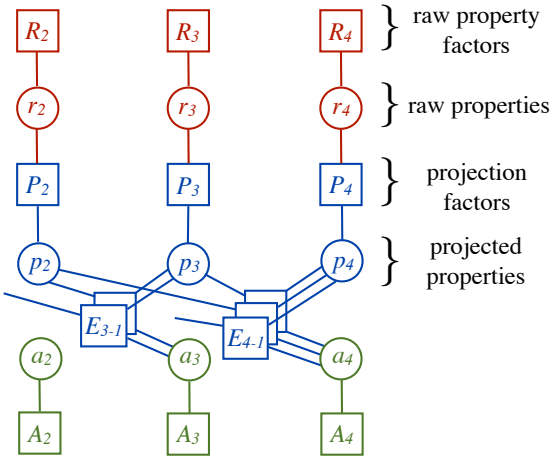


Figure 4.7: The complete factor graph for our TRANSITIVE coreference model. Compared to Figure 4.6, the R_i contain the raw cluster posteriors, and the P_i factors now project raw cluster values r_i into a set of "coreference-adapted" clusters p_i that are used as before. This projection allows mentions with different but compatible raw property values to coexist in the same coreference chain.

of the r_i onto the p_i, which can now be thought of as "coreference-adapted" properties. The P_i factors are defined by $P_i(p_i, r_i) \propto \exp(\mathbf{w}^T \mathbf{f}_P(p_i, r_i))$, where $\mathbf{f}_P$ is a feature vector over the projection of r_i onto p_i. While there are many possible choices of $\mathbf{f}_P$, we choose it to be an indicator of the values of p_i and r_i, so that we learn a fully-parameterized projection matrix.[14] The R_i are constant factors, and may come from an upstream model or some other source depending on the property being modeled.[15]

Our description thus far has assumed that we are modeling only one type of property. In fact, we can use multiple properties for each mention by duplicating the r and p nodes and the R, P, and E factors across each desired property. We index each of these by $l \in \{1, \ldots, m\}$ for each of m properties.

The final log-linear model is given by the following formula:

$$p(a|x) \propto \sum_{p,r} \left[\left(\prod_{i,j,l} E_{l,i-j}(a_i, p_{li}, p_{lj}) R_{li}(r_{li}) \right) \exp\left(\mathbf{w}^T \sum_i \left(\mathbf{f}_A(i, a_i, x) + \sum_l \mathbf{f}_P(p_{li}, r_{li}) \right) \right) \right]$$

[14] Initialized to zero (or small values), this matrix actually causes the transitive machinery to have no effect, since all posteriors over the p_i are flat and completely uninformative. Therefore, we regularize the weights of the indicators of $p_i = r_i$ towards 1 and all other features towards 0 to give each raw cluster a preference for a distinct projected cluster.

[15] In practice, this scheme has much the same effect as that presented in Section 4.3.2. However, theoretically there are some distinctions. Consider, for example, the merger of two clusters $\{m_{1,1}, m_{1,2}, \ldots, m_{1,n}, m_{2,1}, m_{2,2}, \ldots, m_{2,k}\}$ with consistently different property values between the $m_{1,i}$ and the $m_{2,i}$. In the model of Section 4.3.2, we only pay a "switching cost" along the edge $(m_{1,n}, m_{2,1})$, but in the present model, we pay a cost for either n or k mentions to mutate their property values accordingly.

where i and j range over mentions, l ranges over each of m properties, and the outer sum indicates marginalization over all p and r variables.

Learning and inference in this model proceed as described in Section 5.3 and Section 4.5; this model behaves analogously to the joint model.

4.8.1 Systems

We can compare the incorporation of features this way to both a basic pairwise system Besides our INDEP[16] and TRANSITIVE systems, we evaluate a strictly pairwise system that incorporates property information by way of indicator features on the current mention's most likely property value and the proposed antecedent's most likely property value. We call this system PAIRPROPERTY; it is simply the INDEP system with an expanded feature set.

Furthermore, we compare against a LEFTTORIGHT entity-level system like that of Rahman and Ng (2009).[17] Decoding now operates in a sequential fashion, with INDEP features computed as before and entity features computed for each mention based on the coreference decisions made thus far. Following Rahman and Ng (2009), features for each property indicate whether the current mention agrees with no mentions in the antecedent cluster, at least one mention, over half of the mentions, or all of the mentions; antecedent clusters of size 1 or 2 fire special-cased features. These additional features beyond those in Rahman and Ng (2009) were helpful, but more involved conjunction schemes and fine-grained features were not. During training, entity features of both the gold and the prediction are computed using the Viterbi clustering of preceding mentions under the current model parameters.[18]

All systems are run in a two-pass manner: first, the INDEP model is run, then antecedent choices are pruned, then our second-round model is trained from scratch on the pruned data.[19]

4.8.2 Noisy Oracle Features

We first evaluate our model's ability to exploit synthetic entity-level properties. For this experiment, mention properties are derived from corrupted oracle information about the true underlying coreference cluster. Each coreference cluster is assumed to have one underlying value for each of m coreference properties, each taking values over a domain D. Mentions then

[16] In this section, we use a slightly reduced version of the INDEP feature set which achieves slightly lower results than the version we use in the rest of this chapter. Our versions of the metrics are also slightly different.

[17] Unfortunately, their publicly-available system is closed-source and performs poorly on the CoNLL shared task dataset, so direct comparison is difficult.

[18] Using gold entities for training as in Rahman and Ng (2009) resulted in a lower-performing system.

[19] We even do this for the INDEP model, since we found that performance of the pruned and retrained model was generally higher.

Noisy Oracle				
	MUC	B^3	CEAF_e	Avg.
Indep	61.96	70.66	47.30	59.97
PairProperty	66.31	72.68	49.08	62.69
LeftToRight	66.49	73.14	49.46	63.03
Transitive	**67.37**	**74.05**	**49.68**	**63.70**

Table 4.7: CoNLL metric scores on the CoNLL 2011 development set for our four different systems incorporating noisy oracle data. This information helps substantially in all cases. Both entity-level models outperform the PairProperty model, but we observe that the Transitive model is more effective than the LeftToRight model at using this information.

sample distributions over D from a Dirichlet distribution peaked around the true underlying value.[20] These posteriors are taken as the R_i for the Transitive model.

We choose this setup to reflect two important properties of entity-level information: first, that it may come from a variety of disparate sources, and second, that it may be based on the determinations of upstream models which produce posteriors naturally. A strength of our model is that it can accept such posteriors as input, naturally making use of this information in a model-based way.

Table 4.7 shows development results averaged across ten train-test splits with $m = 3$ properties, each taking one of $|D| = 5$ values. We emphasize that these parameter settings give fairly weak oracle information: a document may have hundreds of clusters, so even in the absence of noise these oracle properties do not have high discriminating power. Still, we see that all models are able to benefit from incorporating this information; however, our Transitive model outperforms both the PairProperty model and the LeftToRight model. There are a few reasons for this: first, our model is able to directly use soft posteriors, so it is able to exploit the fact that more peaked samples from the Dirichlet are more likely to be correct. Moreover, our model can propagate information backwards in a document as well as forwards, so the effects of noise can be more easily mitigated. By contrast, in the LeftToRight model, if the first or second mention in a cluster has the wrong property value, features indicating high levels of property agreement will not fire on the next few mentions in those clusters.

[20]Specifically, the distribution used is a Dirichlet with $\alpha = 3.5$ for the true underlying cluster and $\alpha = 1$ for other values, chosen so that 25% of samples from the distribution did not have the correct mode. Though these parameters affect the quality of the oracle information, varying them did not change the relative performance of the different models.

Phi Features				
	MUC	B^3	CEAF_e	Avg.
Indep	61.96	70.66	47.30	59.97
LeftToRight	61.34	70.41	**47.64**	59.80
Transitive	**62.66**	**70.92**	46.88	**60.16**
Phi Features (ablated Basic)				
Indep-Phi	59.45	69.21	46.02	58.23
PairProperty	61.88	70.66	47.14	59.90
LeftToRight	61.42	70.53	**47.49**	59.81
Transitive	**62.23**	**70.78**	46.74	**59.92**

Table 4.8: CoNLL metric scores on the CoNLL 2011 development set for our systems incorporating phi features. Our standard Indep system already includes phi features, so no results are reported for PairProperty. Here, our Transitive system does not give substantial improvement on the averaged metric. Over a baseline which does not include phi features, all systems are able to incorporate them comparably.

4.8.3 Phi Features

As we have seen, our Transitive model can exploit high-quality entity-level features. How does it perform using real features that have been proposed for entity-level coreference?

Here, we use hard phi feature determinations extracted from the system of Lee et al. (2011). Named-entity type and animacy are both computed based on the output of a named-entity tagger, while number and gender use the dataset of Bergsma and Lin (2006). Once this information is determined, the PairProperty and LeftToRight systems can compute features over it directly. In the Transitive model, each of the R_i factors places $\frac{3}{4}$ of its mass on the determined label and distributes the remainder uniformly among the possible options.

Table 4.8 shows results when adding entity-level phi features on top of our Indep pairwise system (which already contains pairwise features) and on top of an ablated Indep system without pairwise phi features. Our entity-level systems successfully captures phi features when they are not present in the baseline, but there is only slight benefit over pairwise incorporation, a result which has been noted previously (Luo et al., 2004).

4.8.4 Clustering Features

Finally, we consider mention properties derived from unsupervised clusterings, which should capture semantic properties of nominals.

We consider clusterings that take as input pairs (n, r) of a noun head n and a string r which contains the semantic role of n (or some approximation thereof) conjoined with its governor. Two different algorithms are used to cluster these pairs: a NaiveBayes model, where c generates n and r, and a Conditional model, where c is generated conditioned on

Clusters				
	MUC	B^3	$CEAF_e$	Avg.
Indep	61.96	70.66	47.30	59.97
PairProperty	62.88	70.71	**47.45**	60.35
LeftToRight	61.98	70.19	45.77	59.31
Transitive	**63.34**	**70.89**	46.88	**60.37**

Table 4.9: CoNLL metric scores on the CoNLL 2011 development set for our systems incorporating four types of clustering features. These features are equally effectively incorporated by our PairProperty system and our Transitive system, showing that the extra machinery of the Transitive system gives little benefit in this case.

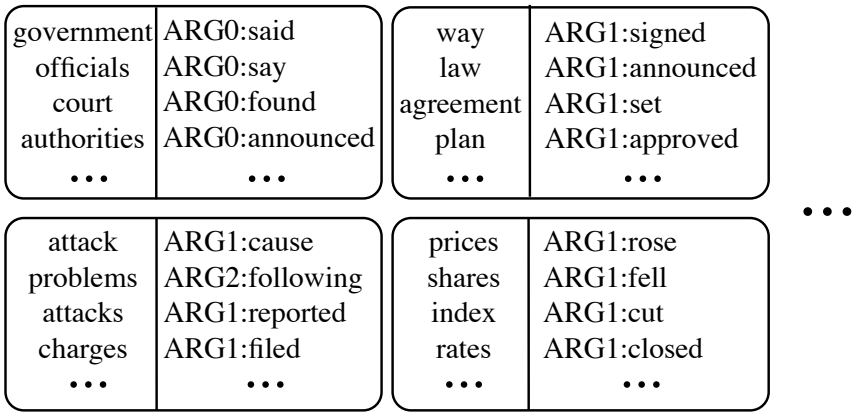


Figure 4.8: Examples of clusters produced by the NaiveBayes model on SRL-tagged data with pronouns discarded.

r and then n is generated from c. Parameters for each can be learned with the expectation maximization (EM) algorithm (Dempster et al., 1977), with symmetry broken by a small amount of random noise at initialization.

Similar models have been used to learn subcategorization information (Rooth et al., 1999) or properties of verb argument slots (Yao et al., 2011). We choose this kind of clustering for its relative simplicity and because it allows pronouns to have more informed properties (from their verbal context) than would be possible using a model that makes type-level decisions about nominals only. Though these specific cluster features are novel to coreference, previous work has used similar types of fine-grained semantic class information (Hendrickx and Daelemans, 2007; Ng, 2007; Rahman and Ng, 2010). Other approaches incorporate information from other sources (Ponzetto and Strube, 2006) or compute heuristic scores for real-valued features based on a large corpus or the web (Dagan and Itai, 1990; Yang et al., 2005; Bansal and Klein, 2012).

All of our experiments use a set of four cluster-based properties, each with twenty clus-

ters: dependency-parse-derived NaiveBayes clusters, semantic-role-derived Conditional clusters, SRL-derived NaiveBayes clusters generating a NoVerb token when r cannot be determined, and SRL-derived NaiveBayes clusters with all pronoun tuples discarded. Examples of the latter clusters are shown in Figure 4.8. Each clustering is learned for 30 iterations of EM over English Gigaword (Graff et al., 2007), parsed with the Berkeley Parser (Petrov et al., 2006) and with SRL determined by Senna (Collobert et al., 2011).

Table 4.9 shows results of modeling these four cluster properties. As in the case of oracle features, the PairProperty and LeftToRight systems use the modes of the cluster posteriors, and the Transitive system uses the posteriors directly as the R_i. We see comparable performance from incorporating features in both an entity-level framework and a pairwise framework, though the Transitive system appears to be more effective than the LeftToRight system. All together, we do not see a great benefit from using our Transitive model to capture this information, and gains are substantially less than those from using NER types in Section 4.6.3.

4.9 Conclusion

We return to our initial motivation for joint modeling, namely that the three tasks we address have the potential to influence one another. Table 4.3 shows that failing to exploit any of the pairwise interactions between the tasks causes lower performance on at least one of them. Therefore, any pipelined system would necessarily underperform a joint model on whatever task came first in the pipeline, which is undesirable given the importance of these tasks. The trend towards broader and deeper NLP pipelines will only exacerbate this problem and make it more difficult to find a suitable pipeline ordering. In addition to showing that joint modeling is high-performing, we have also shown that it can be implemented with relatively low overhead, requiring no fundamentally new learning or inference techniques, and that it is extensible, due to its modular structure and natural partitioning of features. Taken together, these aspects make a compelling case that joint models can provide a way to integrate deeper levels of processing, particularly for semantic layers of annotation, and that this modeling power does not need to come at the expense of computational efficiency, structural simplicity, or modularity.

Chapter 5

Compressive Summarization with Pronoun Coreference Constraints[1]

5.1 Introduction

In Chapters 3 and 4, we explored automatic approaches to coreference resolution and entity linking. In this chapter, we shift our focus slightly: rather than trying to solve those problems as completely as possible, we instead investigate how to apply our JOINT system from Chapter 4 to a downstream task, namely single-document summarization.

While multi-document summarization is well-studied in the NLP literature (Carbonell and Goldstein, 1998; Gillick and Favre, 2009; Lin and Bilmes, 2011; Nenkova and McKeown, 2011), single-document summarization (McKeown et al., 1995; Marcu, 1998; Mani, 2001; Hirao et al., 2013) has received less attention in recent years and is generally viewed as more difficult. Content selection is tricky without redundancy across multiple input documents as a guide and simple positional information is often hard to beat (Penn and Zhu, 2008). We tackle the single-document problem by training an expressive summarization model on a large naturally occurring corpus—the New York Times Annotated Corpus (Sandhaus, 2008) which contains around 100,000 news articles with abstractive summaries—learning to select important content with lexical features. This corpus has been explored in related contexts (Dunietz and Gillick, 2014; Hong and Nenkova, 2014), but to our knowledge it has not been directly used for single-document summarization.

To increase the expressive capacity of our model we allow more aggressive compression of individual sentences by combining two different formalisms—one syntactic and the other discursive. Additionally, we incorporate constraints from anaphora resolution using our system from Chapter 4 and give our system the ability rewrite pronominal mentions, further increasing expressivity. Our constraints from coreference ensure that critical pronoun references are clear in the final summary, even in the presence of two mechanisms for sentence compression. Despite the complexity of these additional constraints, we demonstrate an

[1] An early version of this chapter appeared in Durrett et al. (2016).

efficient inference procedure using an ILP-based approach. By training our full system end-to-end on a large-scale dataset, we are able to learn a high-capacity structured model of the summarization process, contrasting with past approaches to the single-document task which have typically been heuristic in nature (Daumé and Marcu, 2002; Hirao et al., 2013).

We focus our evaluation on the New York Times Annotated corpus (Sandhaus, 2008). According to ROUGE, our system outperforms a document prefix baseline, a bigram coverage baseline adapted from a strong multi-document system (Gillick and Favre, 2009), and a discourse-informed method from prior work (Yoshida et al., 2014). Imposing discursive and referential constraints improves human judgments of linguistic clarity and referential structure—outperforming the method of Yoshida et al. (2014) and approaching the clarity of a sentence-extractive baseline—and still achieves substantially higher ROUGE score than either method. These results indicate that our model has the expressive capacity to extract important content, but is sufficiently constrained to ensure fluency is not sacrificed as a result.

Past work has explored various kinds of structure for summarization. Some work has focused on improving content selection using discourse structure (Louis et al., 2010; Hirao et al., 2013), topical structure (Barzilay and Lee, 2004), or related techniques (Mithun and Kosseim, 2011). Other work has used structure primarily to reorder summaries and ensure coherence (Barzilay et al., 2001; Barzilay and Lapata, 2008; Louis and Nenkova, 2012; Christensen et al., 2013) or to represent content for sentence fusion or abstraction (Thadani and McKeown, 2013; Pighin et al., 2014). Similar to these approaches, we appeal to structures from upstream NLP tasks (syntactic parsing, RST parsing, and coreference) to restrict our model's capacity to generate. However, we go further by optimizing for ROUGE subject to these constraints with end-to-end learning.

5.2 Model

Our model is shown in Figure 5.1. Broadly, our ILP takes a set of textual units $\mathbf{u} = (u_1, \ldots, u_n)$ from a document and finds the highest-scoring extractive summary by optimizing over variables $\mathbf{x}^{\text{UNIT}} = x_1^{\text{UNIT}}, \ldots, x_n^{\text{UNIT}}$, which are binary indicators of whether each unit is included. Textual units are contiguous parts of sentences that serve as the fundamental units of extraction in our model. For a sentence-extractive model, these would be entire sentences, but for our compressive models we will have more fine-grained units, as shown in Figure 5.2 and described in Section 5.2.1. Textual units are scored according to features $\mathbf{f}$ and model parameters $\mathbf{w}$ learned on training data. Finally, the extraction process is subject to a length constraint of k words. This approach is similar in spirit to ILP formulations of multi-document summarization systems, though in those systems content is typically modeled in terms of bigrams (Gillick and Favre, 2009; Berg-Kirkpatrick et al., 2011; Hong and Nenkova, 2014; Li et al., 2015). For our model, type-level n-gram scoring only arises when we compute our loss function in max-margin training (see Section 5.3).

$$\max_{\mathbf{x}^{\text{UNIT}},\mathbf{x}^{\text{REF}}} \left[\underbrace{\sum_i \left[x_i^{\text{UNIT}}(\mathbf{w}^\top \mathbf{f}(u_i))\right]}_{\textbf{Extraction Score}} + \underbrace{\sum_{(i,j)} \left[x_{ij}^{\text{REF}}(\mathbf{w}^\top \mathbf{f}(r_{ij}))\right]}_{\textbf{Anaphora Score}} \right]$$

subject to

Grammaticality Constraints (Section 2.1)

$\forall i,k \quad x_i^{\text{UNIT}} \leq x_k^{\text{UNIT}}$ if u_i requires u_k

Length Constraint

$$\sum_i x_i^{\text{UNIT}}|u_i| + \overbrace{\sum_{(i,j)} x_{ij}^{\text{REF}}(|r_{ij}|-1)}^{\textbf{Length adjustment for explicit mention}} \leq k$$

Anaphora Constraints (Section 2.2)

$\forall j \quad x_{ij}^{\text{REF}} = 1$ iff no prior included textual unit mentions the entity that r_{ij} refers to

$\forall i,k \quad x_i^{\text{UNIT}} \leq x_k^{\text{UNIT}}$ if u_i requires u_k on the basis of pronoun anaphora

Figure 5.1: ILP formulation of our single-document summarization model. The basic model extracts a set of textual units with binary variables $\mathbf{x}^{\text{UNIT}}$ subject to a length constraint. These textual units $\mathbf{u}$ are scored with weights $\mathbf{w}$ and features $\mathbf{f}$. Next, we add constraints derived from both syntactic parses and Rhetorical Structure Theory (RST) to enforce grammaticality. Finally, we add anaphora constraints derived from coreference in order to improve summary coherence. We introduce additional binary variables $\mathbf{x}^{\text{REF}}$ that control whether each pronoun is replaced with its antecedent using a candidate replacement r_{ij}. These are also scored in the objective and are incorporated into the length constraint.

In Section 5.2.1, we discuss grammaticality constraints, which take the form of introducing dependencies between textual units, as shown in Figure 5.2. If one textual unit *requires* another, it cannot be included unless its prerequisite is. We will show that different sets of requirements can capture both syntactic and discourse-based compression schemes.

Furthermore, we introduce anaphora constraints (Section 5.2.2) via a new set of variables that capture the process of rewriting pronouns to make them explicit mentions. That is, $x_{ij}^{\text{REF}} = 1$ if we should rewrite the jth pronoun in the ith unit with its antecedent. These pronoun rewrites are scored in the objective and introduced into the length constraint to make sure they do not cause our summary to be too long. Finally, constraints on these variables control when they are used and also require the model to include antecedents of pronouns when the model is not confident enough to rewrite them.

5.2.1 Grammaticality Constraints

Following work on isolated sentence compression (McDonald, 2006; Clarke and Lapata, 2008) and compressive summarization (Lin, 2003; Martins and Smith, 2009; Berg-Kirkpatrick et al., 2011; Woodsend and Lapata, 2012; Almeida and Martins, 2013), we wish to be able to

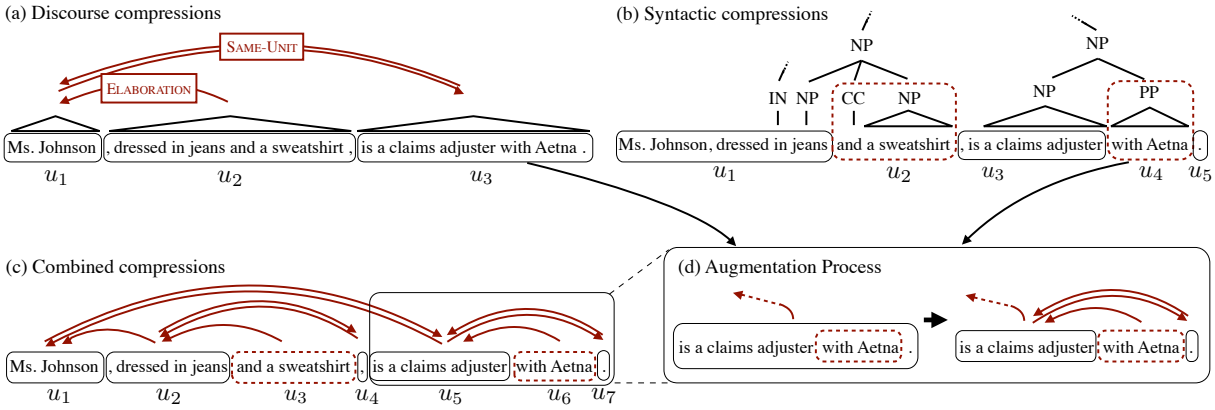


Figure 5.2: Compression constraints on an example sentence. (a) RST-based compression structure like that in Hirao et al. (2013), where we can delete the ELABORATION clause. (b) Two syntactic compression options from Berg-Kirkpatrick et al. (2011), namely deletion of a coordinate and deletion of a PP modifier. (c) Textual units and requirement relations (arrows) after merging all of the available compressions. (d) Process of augmenting a textual unit with syntactic compressions.

compress sentences so we can pack more information into a summary. During training, our model learns how to take advantage of available compression options and select content to match human generated summaries as closely possible.[2] We explore two ways of deriving units for compression: the RST-based compressions of Hirao et al. (2013) and the syntactic compressions of Berg-Kirkpatrick et al. (2011).

RST compressions Figure 5.2a shows how to derive compressions from Rhetorical Structure Theory (Mann and Thompson, 1988; Carlson et al., 2001). We show a sentence broken into elementary discourse units (EDUs) with RST relations between them. Units marked as SAME-UNIT must both be kept or both be deleted, but other nodes in the tree structure can be deleted as long as we do not delete the parent of an included node. For example, we can delete the ELABORATION clause, but we can delete neither the first nor last EDU. Arrows depict the constraints this gives rise to in the ILP (see Figure 5.1): u_2 requires u_1, and u_1 and u_3 mutually require each other. This is a more constrained form of compression than was used in past work (Hirao et al., 2013), but we find that it improves human judgments of fluency (Section 5.4.3).

[2]The features in our model are actually rich enough to learn a sophisticated compression model, but the data we have (abstractive summaries) does not directly provide examples of correct compressions; past work has gotten around this with multi-task learning (Almeida and Martins, 2013), but we simply treat grammaticality as a constraint from upstream models.

Syntactic compressions Figure 5.2b shows two examples of compressions arising from syntactic patterns (Berg-Kirkpatrick et al., 2011): deletion of the second part of a coordinated NP and deletion of a PP modifier to an NP. These patterns were curated to leave sentences as grammatical after being compressed, though perhaps with damaged semantic content.

Combined compressions Figure 5.2c shows the textual units and requirement relations yielded by combining these two types of compression. On this example, the two schemes capture orthogonal compressions, and more generally we find that they stack to give better results for our final system (see Section 5.4.3). To actually synthesize textual units and the constraints between them, we start from the set of RST textual units and introduce syntactic compressions as new children when they don't cross existing brackets; because syntactic compressions are typically narrower in scope, they are usually completely contained in EDUs. Figure 5.2d shows an example of this process: the possible deletion of *with Aetna* is grafted onto the textual unit and appropriate requirement relations are introduced. The net effect is that the textual unit is wholly included, partially included (*with Aetna* removed), or not at all.

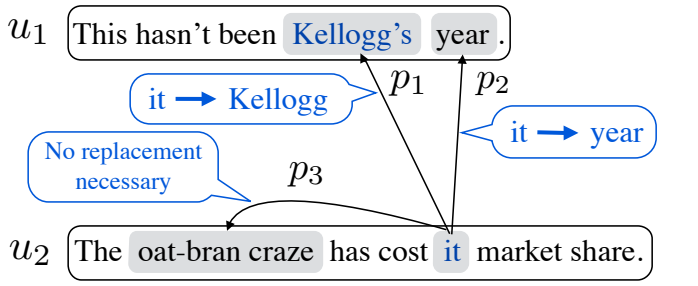
Formally, we define an RST tree as $T_{\text{rst}} = (S_{\text{rst}}, \pi_{\text{rst}})$ where S_{rst} is a set of EDU spans (i, j) and $\pi : S \rightarrow 2^S$ is a mapping from each EDU span to EDU spans it depends on. Syntactic compressions can be expressed in a similar way with trees T_{syn}. These compressions are typically smaller-scale than EDU-based compressions, so we use the following modification scheme. Denote by $T_{\text{syn}(kl)}$ a nontrivial (supports some compression) subtree of T_{syn} that is completely contained in an EDU (i, j). We build the following combined compression tree, which we refer to as the *augmentation* of T_{rst} with $T_{\text{syn}(kl)}$:

$$T_{\text{comb}} = (S \cup S_{\text{syn}}(kl) \cup \{(i,k),(l,j)\}, \pi_{\text{rst}} \cup \pi_{\text{syn}(kl)} \cup \{(i,k) \rightarrow (l,j), (l,j) \rightarrow (i,k), (k,l) \rightarrow (i,k)\})$$

That is, we maintain the existing tree structure except for the EDU (i, j), which is broken into three parts: the outer two depend on each other (*is a claims adjuster* and . from Figure 5.2d) and the inner one depends on the others and preserves the tree structure from T_{syn}. We augment T_{rst} with all maximal subtrees of T_{syn}, i.e. all trees that are not contained in other trees that are used in the augmentation process.

This is broadly similar to the combined compression scheme in Kikuchi et al. (2014) but we use a different set of constraints that more strictly enforce grammaticality.[3]

[3]We also differ from past work in that we do not use cross-sentential RST constraints (Hirao et al., 2013; Yoshida et al., 2014). We experimented with these and found no improvement from using them, possibly because we have a feature-based model rather than a heuristic content selection procedure, and possibly because automatic discourse parsers are less good at recovering cross-sentence relations.



Replacement (2.2.1): If $\max(p_1, p_2, p_3) > \alpha$:

Allow pronoun replacement with the predicted antecedent and add the following constraint:

$$x_{2,1}^{\text{REF}} = 1 \text{ iff } x_1^{\text{UNIT}} = 0 \text{ and } x_2^{\text{UNIT}} = 1$$

Replace the first pronoun in the second textual unit

Otherwise (i.e. if no replacement is possible):

Antecedent inclusion (2.2.2): If $p_1 + p_2 > \beta$

Add the following constraint: $x_2^{\text{UNIT}} \leq x_1^{\text{UNIT}}$

Figure 5.3: Modifications to the ILP to capture pronoun coherence. *It*, which refers to *Kellogg*, has several possible antecedents from the standpoint of an automatic coreference system, such as that in Chapter 4. If the coreference system is confident about its selection (above a threshold α on the posterior probability), we allow for the model to explicitly replace the pronoun if its antecedent would be deleted (Section 5.2.2). Otherwise, we merely constrain one or more probable antecedents to be included (Section 5.2.2); even if the coreference system is incorrect, a human can often correctly interpret the pronoun with this additional context.

5.2.2 Anaphora Constraints

What kind of cross-sentential coherence do we need to ensure for the kinds of summaries our system produces? Many notions of coherence are useful, including centering theory (Grosz et al., 1995) and lexical cohesion (Nishikawa et al., 2014), but one of the most pressing phenomena to deal with is pronoun anaphora (Clarke and Lapata, 2010). Cases of pronouns being "orphaned" during extraction (their antecedents are deleted) are relatively common: they occur in roughly 60% of examples produced by our summarizer when no anaphora constraints are enforced. This kind of error is particularly concerning for summary

interpretation and impedes the ability of summaries to convey information effectively (Grice, 1975). Our solution is to explicitly impose constraints on the model based on pronoun anaphora resolution.[4]

Figure 5.3 shows an example of a problem case. If we extract only the second textual unit shown, the pronoun *it* will lose its antecedent, which in this case is *Kellogg*. We explore two types of constraints for dealing with this: rewriting the pronoun explicitly, or constraining the summary to include the pronoun's antecedent.

Pronoun Replacement

One way of dealing with these pronoun reference issues is to explicitly replace the pronoun with what it refers to. This replacement allows us to maintain maximal extraction flexibility, since we can make an isolated textual unit meaningful even if it contains a pronoun. Figure 5.3 shows how this process works. We run the JOINT model from Chapter 4 and compute posteriors over possible links for the pronoun. If the coreference system is sufficiently confident in its prediction (i.e. $\max_i p_i > \alpha$ for a specified threshold $\alpha > \frac{1}{2}$), we allow ourselves to replace the pronoun with the first mention of the entity corresponding to the pronoun's most likely antecedent. In Figure 5.3, if the system correctly determines that *Kellogg* is the correct antecedent with high probability, we enable the first replacement shown there, which is used if u_2 is included the summary without u_1.[5]

As shown in the ILP in Figure 5.1, we instantiate corresponding pronoun replacement variables $\mathbf{x}^{\text{REF}}$ where $x_{ij}^{\text{REF}} = 1$ implies that the jth pronoun in the ith sentence should be replaced in the summary. We use a candidate pronoun replacement if and only if the pronoun's corresponding (predicted) entity hasn't been mentioned previously in the summary.[6] Because we are generally replacing pronouns with longer mentions, we also need to modify the length constraint to take this into account. Finally, we incorporate features on pronoun replacements in the objective, which helps the model learn to prefer pronoun replacements that help it to more closely match the human summaries.

Pronoun Antecedent Constraints

Explicitly replacing pronouns is risky: if the coreference system makes an incorrect prediction, the intended meaning of the summary may be damaged. Fortunately, our coreference model's posterior probabilities have been shown to be well-calibrated (Nguyen and O'Connor, 2015), meaning that cases where it is likely to make errors are signaled by flatter posterior

[4]We focus on pronoun coreference because it is the most pressing manifestation of this problem and because existing coreference systems perform well on pronouns compared to harder instances of coreference, as established in Chapter 3.

[5]If the proposed replacement is a proper mention, we replace the pronoun just with the subset of the mention that constitutes a named entity (rather than the whole noun phrase). We control for possessive pronouns by deleting or adding *'s* as appropriate.

[6]Such a previous mention may be a pronoun; however, note that that pronoun would then be targeted for replacement unless its antecedent were included somehow.

distributions. In this case, we enable a more conservative set of constraints that include additional content in the summary to make the pronoun reference clear without explicitly replacing it. This is done by requiring the inclusion of any textual unit which contains possible pronoun references whose posteriors sum to at least a threshold parameter β. Figure 5.3 shows that this constraint can force the inclusion of u_1 to provide additional context. Although this could still lead to unclear pronouns if text is stitched together in an ambiguous or even misleading way, in practice we observe that the textual units we force to be added almost always occur very recently before the pronoun, giving enough additional context for a human reader to figure out the pronoun's antecedent unambiguously.

5.2.3 Features

The features in our model (see Figure 5.1) consist of a set of surface indicators capturing mostly lexical and configurational information. Their primary role is to identify important document content. The first three types of features fire over textual units, the last over pronoun replacements.

Lexical These include indicator features on non-stopwords in the textual unit that appear at least five times in the training set and analogous POS features. We also use lexical features on the first, last, preceding, and following words for each textual unit. Finally, we conjoin each of these features with an indicator of bucketed position in the document (the index of the sentence containing the textual unit).

Structural These features include various conjunctions of the position of the textual unit in the document, its length, the length of its corresponding sentence, the index of the paragraph it occurs in, and whether it starts a new paragraph (all values are bucketed).

Centrality These features capture rough information about the centrality of content: they consist of bucketed word counts conjoined with bucketed sentence index in the document. We also fire features on the number of times of each entity mentioned in the sentence is mentioned in the rest of the document (according to a coreference system), the number of entities mentioned in the sentence, and surface properties of mentions including type and length

Pronoun replacement These target properties of the pronoun replacement such as its length, its sentence distance from the current mention, its type (nominal or proper), and the identity of the pronoun being replaced.

5.3 Learning

We learn weights $\mathbf{w}$ for our model by training on a large corpus of documents $\mathbf{u}$ paired with reference summaries $\mathbf{y}$. We formulate our learning problem as a standard instance of structured SVM (see Smith (2011) for an introduction). Because we want to optimize explicitly for ROUGE-1,[7] we define a ROUGE-based loss function that accommodates the nature of our supervision, which is in terms of abstractive summaries $\mathbf{y}$ that in general cannot be produced by our model. Specifically, we take:

$$\ell(\mathbf{x}^{\text{NGRAM}}, \mathbf{y}) = \max_{\mathbf{x}^*} \text{ROUGE-1}(\mathbf{x}^*, \mathbf{y}) - \text{ROUGE-1}(\mathbf{x}^{\text{NGRAM}}, \mathbf{y})$$

i.e. the gap between the hypothesis's ROUGE score and the oracle ROUGE score achievable under the model (including constraints). Here $\mathbf{x}^{\text{NGRAM}}$ are indicator variables that track, for each n-gram type in the reference summary, whether that n-gram is present in the system summary. These are the sufficient statistics for computing ROUGE.

We train the model via stochastic subgradient descent on the primal form of the structured SVM objective (Ratliff et al., 2007; Kummerfeld et al., 2015). In order to compute the subgradient for a given training example, we need to find the most violated constraint on the given instance through a loss-augmented decode, which for a linear model takes the form $\arg\max_{\mathbf{x}} \mathbf{w}^\top \mathbf{f}(\mathbf{x}) + \ell(\mathbf{x}, \mathbf{y})$. To do this decode at training time in the context of our model, we use an extended version of our ILP in Figure 5.1 that is augmented to explicitly track type-level n-grams:

$$\max_{\mathbf{x}^{\text{UNIT}}, \mathbf{x}^{\text{REF}}, \mathbf{x}^{\text{NGRAM}}} \left[\sum_i \left[x_i^{\text{UNIT}} (\mathbf{w}^\top \mathbf{f}(u_i)) \right] + \sum_{(i,j)} \left[x_{ij}^{\text{REF}} (\mathbf{w}^\top \mathbf{f}(r_{ij})) \right] - \ell(\mathbf{x}^{\text{NGRAM}}, \mathbf{y}) \right]$$

subject to all constraints from Figure 5.1, and

$x_i^{\text{NGRAM}} = 1$ iff an included textual unit or replacement contains the ith reference n-gram

These kinds of variables and constraints are common in multi-document summarization systems that score bigrams (Gillick and Favre, 2009 *inter alia*). Note that since ROUGE is only computed over non-stopword n-grams and pronoun replacements only replace pronouns, pronoun replacement can never remove an n-gram that would otherwise be included.

For all experiments, we optimize our objective using AdaGrad (Duchi et al., 2011) with ℓ_1 regularization ($\lambda = 10^{-8}$, chosen by grid search), with a step size of 0.1 and a minibatch size of 1. We train for 10 iterations on the training data, at which point held-out model

[7] We found that optimizing for ROUGE-1 actually resulted in a model with better performance on both ROUGE-1 and ROUGE-2. We hypothesize that this is because framing our optimization in terms of ROUGE-2 would lead to a less nuanced set of constraints: bigram matches are relatively rare when the reference is a short, abstractive summary, so a loss function based on ROUGE-2 will express a flatter preference structure among possible outputs.

NYT50 article:

Federal officials reported yesterday that students in 4th, 8th and 12th grades had scored modestly higher on an American history test than five years earlier, although more than half of high school seniors still showed poor command of basic facts like the effect of the cotton gin on the slave economy or the causes of the Korean War. Federal officials said they considered the results encouraging because at each level tested, student performance had improved since the last time the exam was administered, in 2001. "In U.S. history there were higher scores in 2006 for all three grades," said Mark Schneider, commissioner of the National Center for Education Statistics, which administers the test, at a Boston news conference that the Education Department carried by Webcast. **The results were less encouraging on a national civics test, on which only fourth graders made any progress.** The best results in the history test were also in fourth grade, where 70 percent of students attained the basic level of achievement or better. **The test results in the two subjects are likely to be closely studied, because Congress is considering the renewal of President Bush's signature education law, the No Child Left Behind Act.** A number of studies have shown that because No Child Left Behind requires states…

Summary:

National Center for Education Statistics reports students in 4th, 8th and 12th grades scored modestly higher on American history test than five years earlier. Says more than half of high school seniors still show poor command of basic facts. Only 4th graders made any progress in civics test. New exam results are another ingredient in debate over renewing Pres Bush's signature No Child Left Behind Act.

Filtered article:

Long before President Bush's proposal to rethink Social Security became part of the national conversation, Westchester County came up with its own dialogue to bring issues of aging to the forefront. Before the White House Conference on Aging scheduled in October, **the county's Office for the Aging a year ago started Speak-Up,** which stands for Student Participants Embrace Aging Issues of Key Concern, to reach students in the county's 13 colleges and universities. Through a variety of events **to bring together the elderly and college students,** organizers said they hoped to have by this spring a series of recommendations that could be given to Washington…

Summary:

Article on Speak-Up, program begun by Westchester County Office for the Aging to bring together elderly and college students.

Figure 5.4: Examples of an article kept in the NYT50 dataset (top) and an article removed because the summary is too short. The top summary has a rich structure to it, corresponding to various parts of the document (bolded) and including some text that is essentially a direct extraction.

performance no longer improves. Finally, we set the anaphora thresholds $\alpha = 0.8$ and $\beta = 0.6$ (see Section 5.2.2). The values of these and other hyperparameters were determined on a held-out development set from our New York Times training data. All ILPs are solved using GLPK version 4.55.

5.4 Experiments

We primarily evaluate our model on a roughly 3000-document evaluation set from the New York Times Annotated Corpus (Sandhaus, 2008). We also investigate its performance on the RST Discourse Treebank (Carlson et al., 2001), but because this dataset is only 30 documents it provides much less robust estimates of performance.[8] Throughout this section, when we decode a document, we set the word budget for our summarizer to be the same as the number of words in the corresponding reference summary, following previous work (Hirao et al., 2013; Yoshida et al., 2014).

[8]Tasks like DUC and TAC have focused on multi-document summarization since around 2003, hence the lack of more standard datasets for single-document summarization.

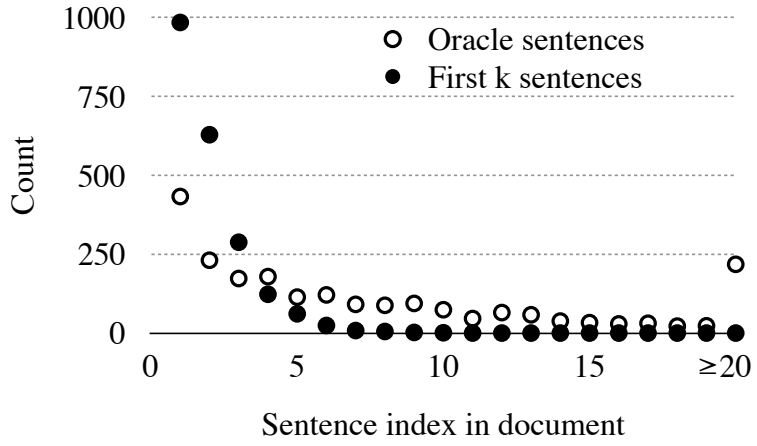


Figure 5.5: Counts on a 1000-document sample of how frequently both a document prefix baseline and a ROUGE oracle summary contain sentences at various indices in the document. There is a long tail of useful sentences later in the document, as seen by the fact that the oracle sentence counts drop off relatively slowly. Smart selection of content therefore has room to improve over taking a prefix of the document.

5.4.1 Preprocessing

We preprocess all data using the Berkeley Parser (Petrov et al., 2006), specifically the GPU-accelerated version of the parser from (Hall et al., 2014), and our entity resolution system from Chapter 4. For RST discourse analysis, we segment text into EDUs using a semi-Markov CRF trained on the RST treebank with features on boundaries similar to those of Hernault et al. (2010), plus novel features on spans including span length and span identity for short spans.

To follow the conditions of Yoshida et al. (2014) as closely as possible, we also build a discourse parser in the style of Hirao et al. (2013), since their parser is not publicly available. Specifically, we use the first-order projective parsing model of McDonald et al. (2005) and features from Soricut and Marcu (2003), Hernault et al. (2010), and Joty et al. (2013). When using the same head annotation scheme as Yoshida et al. (2014), we outperform their discourse dependency parser on unlabeled dependency accuracy, getting 56% as opposed to 53%.

5.4.2 New York Times Corpus

We now provide some details about the New York Times Annotated corpus. This dataset contains 110,540 articles with abstractive summaries; we split these into 100,834 training and 9706 test examples, based on date of publication (test is all articles published on January

1, 2007 or later). Examples of two documents from this dataset are shown in Figure 5.4. The bottom example demonstrates that some summaries are extremely short and formulaic (especially those for obituaries and editorials). To counter this, we filter the raw dataset by removing all documents with summaries that are shorter than 50 words. One benefit of filtering is that the length distribution of our resulting dataset is more in line with standard summarization evaluations like DUC; it also ensures a sufficient number of tokens in the budget to produce nontrivial summaries. The filtered test set, which we call NYT50, includes 3,452 test examples out of the original 9,706.

Interestingly, this dataset is one where the classic document prefix baseline can be substantially outperformed, unlike in some other summarization settings (Penn and Zhu, 2008). We show this fact explicitly in Section 5.4.3, but Figure 5.5 provides additional analysis in this regard. We compute oracle ROUGE-1 sentence-extractive summaries on a 1000-document subset of the training set and look at where the extracted sentences lie in the document. While they certainly skew earlier in the document, they do not all fall within the document prefix summary. One reason for this is that many of the articles are longer-form pieces that begin with a relatively content-free lede of several sentences, which should be identifiable with lexicosyntactic indicators as are used in our discriminative model.

5.4.3 New York Times Results

We evaluate our system along two axes: first, on content selection, using ROUGE[9] (Lin and Hovy, 2003), and second, on clarity of language and referential structure, using annotators from Amazon Mechanical Turk. We follow the method of Gillick and Liu (2010) for this evaluation and ask Turkers to rate a summary on how grammatical it is using a 10-point Likert scale. Furthermore, we ask how many unclear pronouns references there were in the text. The Turkers do not see the original document or the reference summary, and rate each summary in isolation. Gillick and Liu (2010) showed that for linguistic quality judgments (as opposed to content judgments), Turkers reproduced the ranking of systems according to expert judgments.

To speed up preprocessing and training time on this corpus, we further restrict our training set to only contain documents with fewer than 100 EDUs. All told, the final system takes roughly 20 hours to make 10 passes through the subsampled training data (22,000 documents) on a single core of an Amazon EC2 r3.4xlarge instance.

Table 5.1 shows the results on the NYT50 corpus. We compare several variants of our system and baselines. For baselines, we use two variants of first k: one which must stop on a sentence boundary (which gives better linguistic quality) and one which always consumes k tokens (which gives better ROUGE). We also use a heuristic sentence-extractive baseline that maximizes the document counts (term frequency) of bigrams covered by the summary, similar

[9]We use the ROUGE 1.5.5 script with the following command line arguments: `-n 2 -x -m -s`. All given results are macro-averaged recall values over the test set.

	R-1 ↑	R-2 ↑	CG ↑	UP ↓
Baselines				
First sentences	28.6	17.3	8.21	0.28
First k words	35.7	21.6	–	–
Bigram Frequency	25.1	9.8	–	–
Past work				
Tree Knapsack	34.7	19.6	7.20	0.42
This work				
Sentence extraction	38.8	23.5	7.93	0.32
EDU extraction	41.9	25.3	6.38	0.65
Full	**42.2**	**25.9**	*†7.52	*0.36
Ablations from Full				
No Anaphoricity	42.5	26.3	7.46	0.44
No Syntactic Compr	41.1	25.0	–	–
No Discourse Compr	40.5	24.7	–	–

Table 5.1: Results on the NYT50 test set (documents with summaries of at least 50 tokens) from the New York Times Annotated Corpus (Sandhaus, 2008). We report ROUGE-1 (R-1), ROUGE-2 (R-2), clarity/grammaticality (CG), and number of unclear pronouns (UP) (lower is better). On content selection, our system substantially outperforms all baselines, our implementation of the tree knapsack system (Yoshida et al., 2014), and learned extractive systems with less compression, even an EDU-extractive system that sacrifices grammaticality. On clarity metrics, our final system performs nearly as well as sentence-extractive systems. The symbols * and † indicate statistically significant gains compared to No Anaphoricity and Tree Knapsack (respectively) with $p < 0.05$ according to a bootstrap resampling test. We also see that removing either syntactic or EDU-based compressions decreases ROUGE.

in spirit to the multi-document method of Gillick and Favre (2009).[10] We also compare to our implementation of the Tree Knapsack method of Yoshida et al. (2014), which matches their results very closely on the RST Discourse Treebank when discourse trees are controlled for. Finally, we compare several variants of our system: purely extractive systems operating over sentences and EDUs respectively, our full system, and ablations removing either the anaphoricity component or parts of the compression module.

In terms of content selection, we see that all of the systems that incorporate end-to-end learning (under "This work") substantially outperform our various heuristic baselines. Our full system using the full compression scheme is substantially better on ROUGE than ablations where the syntactic or discourse compressions are removed. These improvements

[10]Other heuristic multi-document approaches could be compared to, e.g. He et al. (2012), but a simple term frequency method suffices to illustrate how these approaches can underperform in the single-document setting.

	ROUGE-1	ROUGE-2
First k words	23.5	8.3
Tree Knapsack	25.1	8.7
Full	26.3	8.0

Table 5.2: Results for RST Discourse Treebank (Carlson et al., 2001). Differences between our system and the Tree Knapsack system of Yoshida et al. (2014) are not statistically significant, reflecting the high variance in this small (20 document) test set.

reflect the fact that more compression options give the system more flexibility to include key content words. Removing the anaphora resolution constraints actually causes ROUGE to increase slightly (as a result of granting the model flexibility), but has a negative impact on the linguistic quality metrics.

On our linguistic quality metrics, it is no surprise that the sentence prefix baseline performs the best. Our sentence-extractive system also does well on these metrics. Compared to the EDU-extractive system with no constraints, our constrained compression method improves substantially on both linguistic quality and reduces the number of unclear pronouns, and adding the pronoun anaphora constraints gives further improvement. Our final system is approaches the sentence-extractive baseline, particularly on unclear pronouns, and achieves substantially higher ROUGE score.

5.4.4 RST Treebank

We also evaluate on the RST Discourse Treebank, of which 30 documents have abstractive summaries. Following Hirao et al. (2013), we use the gold EDU segmentation from the RST corpus but automatic RST trees. We break this into a 10-document development set and a 20-document test set. Table 5.2 shows the results on the RST corpus. Our system is roughly comparable to Tree Knapsack here, and we note that none of the differences in the table are statistically significant. We also observed significant variation between multiple runs on this corpus, with scores changing by 1-2 ROUGE points for slightly different system variants.[11]

5.5 Conclusion

We presented a single-document summarization system trained end-to-end on a large corpus. We integrate a compression model that enforces grammaticality as well as pronoun anaphoricity constraints that enforce coherence. Our system improves substantially over baseline systems on ROUGE while still maintaining good linguistic quality. Moreover, we

[11]The system of Yoshida et al. (2014) is unavailable, so we use a reimplementation. Our results differ from theirs due to having slightly different discourse trees, which cause large changes in metrics due to high variance on the test set.

see that even an imperfect entity resolution system can help us improve the quality of our summarizer, reducing the number of unclear pronoun references in our summaries. Given that we face uphill battles in making coreference resolution systems better (Chapter 3), it is encouraging to see that these systems can still be gainfully used to improve downstream tasks like summarization.

Chapter 6

Conclusion

In this thesis, we developed a system for entity analysis: Chapter 3 focused on coreference resolution and Chapter 4 built on top of that a larger model that jointly handles coreference resolution, entity linking, and semantic typing. For coreference resolution, we showed that a simple learning-based system relying on a uniform set of surface features can capture diverse phenomena from pronoun agreement to centering The main drawback of this system is its lack of world knowledge and ability to deal with deeper semantic distinctions important for coreference, which we addressed in Chapter 4 by joining our coreference model with models for entity linking and semantic typing. These import some of the world knowledge we need, and we see improvements in all three tasks from the deeper integration of these models.

Then, in Chapter 5, we showed how the entity analysis tools we developed can be applied to single-document summarization. We present a learning-based extractive and compressive summarization model with inference formulated as an ILP. Additional constraints in this ILP allow us to impose a simple model of how pronouns should behave, namely that we should not use a pronoun to refer to something that has not yet been introduced into the discourse. Our entity analysis model is used to predict coreference information, and we allow for more or less bold handling of pronouns based on the posterior probabilities of these predictions. The effectiveness of this approach shows that even imperfect NLP tools have a role to play in improving systems for applications; moreover, we see that coreference resolution plays a critical role in discourse understanding.

One natural direction for future work is to extend joint modeling approach from Chapter 4 to use more sophisticated grounding. We've shown that symbolic representations of world knowledge from Wikipedia are useful for NLP tasks like coreference, but we currently only use a small amount of the knowledge that's out there and we do so in an isolated way. For example, we can connect *France* with *country*, but we don't make more complex logical inferences like a *country* is the same thing as a *nation* which also has a government, which might be referred to with the country name in certain contexts (*France opposed the UN resolution*). This requires combining information from multiple knowledge sources, which can lead to erroneous inferences due to context-dependent relations. To this end, we need better contextual representations of knowledge and ways of reinforcing our inferences, like

double-checking them against knowledge mined from unlabeled corpora. Another way to support these kinds of deductions is through explicit grounding in an environment, such as for a robotics application where the robot is following detailed natural language instructions. All of these methods provide avenues for better holistic processing of language, more thorough use of knowledge, and deeper integration of the core tools in the NLP stack.

Bibliography

Almeida, Miguel and Andre Martins. 2013. "Fast and Robust Compressive Summarization with Dual Decomposition and Multi-Task Learning". In: *Proceedings of the Association for Computational Linguistics (ACL)*.

Bagga, Amit and Breck Baldwin. 1998. "Algorithms for Scoring Coreference Chains". In: *Proceedings of the Linguistic Coreference Workshop at the Conference on Language Resources and Evaluation (LREC)*.

Bansal, Mohit and Dan Klein. 2012. "Coreference Semantics from Web Features". In: *Proceedings of the Association for Computational Linguistics (ACL)*.

Barzilay, Regina and Mirella Lapata. 2008. "Modeling Local Coherence: An Entity-based Approach". In: *Computational Linguistics* 34.1, pp. 1–34. ISSN: 0891-2017.

Barzilay, Regina and Lillian Lee. 2004. "Catching the Drift: Probabilistic Content Models, with Applications to Generation and Summarization". In: *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Barzilay, Regina, Noemie Elhadad, and Kathleen R. McKeown. 2001. "Sentence Ordering in Multidocument Summarization". In: *Proceedings of the International Conference on Human Language Technology Research*.

Bengtson, Eric and Dan Roth. 2008. "Understanding the Value of Features for Coreference Resolution". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Bentivogli, Luisa et al. 2010. "Extending English ACE 2005 Corpus Annotation with Ground-truth Links to Wikipedia". In: *Proceedings of the Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources*.

Berg-Kirkpatrick, Taylor, Dan Gillick, and Dan Klein. 2011. "Jointly Learning to Extract and Compress". In: *Proceedings of the Association for Computational Linguistics (ACL)*.

Bergsma, Shane and Dekang Lin. 2006. "Bootstrapping Path-Based Pronoun Resolution". In: *Proceedings of the Conference on Computational Linguistics and the Association for Computational Linguistics (ACL)*.

Björkelund, Anders and Richárd Farkas. 2012. "Data-driven Multilingual Coreference Resolution using Resolver Stacking". In: *Proceedings of the Joint Conference on Empirical Methods in Natural Language Proceedings and Conference on Computational Natural Language Learning (EMNLP-CoNLL) - Shared Task*.

Björkelund, Anders and Jonas Kuhn. 2014. "Learning Structured Perceptrons for Coreference Resolution with Latent Antecedents and Non-local Features". In: *Proceedings of the Association for Computational Linguistics (ACL)*.

Björkelund, Anders and Pierre Nugues. 2011. "Exploring Lexicalized Features for Coreference Resolution". In: *Proceedings of the Conference on Computational Natural Language Learning (CoNLL): Shared Task*.

Burkett, David and Dan Klein. 2012. "Fast Inference in Phrase Extraction Models with Belief Propagation". In: *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Cai, Jie and Michael Strube. 2010. "Evaluation Metrics for End-to-End Coreference Resolution Systems". In: *Proceedings of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*.

Carbonell, Jaime and Jade Goldstein. 1998. "The Use of MMR, Diversity-based Reranking for Reordering Documents and Producing Summaries". In: *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Carlson, Lynn, Daniel Marcu, and Mary Ellen Okurowski. 2001. "Building a Discourse-tagged Corpus in the Framework of Rhetorical Structure Theory". In: *Proceedings of the Second SIGDIAL Workshop on Discourse and Dialogue*.

Chang, Kai-Wei, Rajhans Samdani, and Dan Roth. 2013. "A Constrained Latent Variable Model for Coreference Resolution". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Charniak, Eugene and Mark Johnson. 2005. "Coarse-to-Fine n-Best Parsing and MaxEnt Discriminative Reranking". In: *Proceedings of the Association for Computational Linguistics (ACL)*.

Chen, Chen and Vincent Ng. 2012. "Combining the Best of Two Worlds: A Hybrid Approach to Multilingual Coreference Resolution". In: *Proceedings of the Joint Conference on Empirical Methods in Natural Language Proceedings and Conference on Computational Natural Language Learning (EMNLP-CoNLL) - Shared Task*.

Cheng, Xiao and Dan Roth. 2013. "Relational Inference for Wikification". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Christensen, Janara et al. 2013. "Towards Coherent Multi-Document Summarization". In: *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Clarke, James and Mirella Lapata. 2008. "Global Inference for Sentence Compression an Integer Linear Programming Approach". In: *Journal of Artificial Intelligence Research* 31.1, pp. 399–429. ISSN: 1076-9757.

— 2010. "Discourse Constraints for Document Compression". In: *Computational Linguistics* 36.3, pp. 411–441. ISSN: 0891-2017.

Collobert, Ronan et al. 2011. "Natural Language Processing (Almost) from Scratch". In: *Journal of Machine Learning Research* 12, pp. 2493–2537.

Cucerzan, Silviu. 2007. "Large-Scale Named Entity Disambiguation Based on Wikipedia Data". In: *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

Dagan, Ido and Alon Itai. 1990. "Automatic Processing of Large Corpora for the Resolution of Anaphora References". In: *Proceedings of the Conference on Computational Linguistics (COLING) - Volume 3*.

Daumé III, Hal and Daniel Marcu. 2002. "A Noisy-Channel Model for Document Compression". In: *Proceedings of the Association for Computational Linguistics (ACL)*.

— 2005. "A Large-scale Exploration of Effective Global Features for a Joint Entity Detection and Tracking Model". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Dempster, Arthur P., Nan M. Laird, and Donald B. Rubin. 1977. "Maximum Likelihood from Incomplete Data via the EM Algorithm". In: *Journal of the Royal Statistical Society, Series B* 39.1, pp. 1–38.

Denis, Pascal and Jason Baldridge. 2008. "Specialized Models and Ranking for Coreference Resolution". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Dredze, Mark et al. 2010. "Entity Disambiguation for Knowledge Base Population". In: *Proceedings of the International Conference on Computational Linguistics (COLING)*.

Duchi, John, Elad Hazan, and Yoram Singer. 2011. "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization". In: *Journal of Machine Learning Research* 12, pp. 2121–2159.

Dunietz, Jesse and Daniel Gillick. 2014. "A New Entity Salience Task with Millions of Training Examples". In: *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)*.

Durrett, Greg and Dan Klein. 2013. "Easy Victories and Uphill Battles in Coreference Resolution". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

— 2014. "A Joint Model for Entity Analysis: Coreference, Typing, and Linking". In: *Transactions of the Association for Computational Linguistics (TACL)*.

Durrett, Greg, David Hall, and Dan Klein. 2013. "Decentralized Entity-Level Modeling for Coreference Resolution". In: *Proceedings of the Association for Computational Linguistics (ACL)*.

Durrett, Greg, Taylor Berg-Kirkpatrick, and Dan Klein. 2016. "Learning-Based Single-Document Summarization with Compression and Anaphoricity Constraints". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Fahrni, Angela and Michael Strube. 2014. "A Latent Variable Model for Discourse-aware Concept and Entity Disambiguation". In: *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)*.

Fernandes, Eraldo Rezende, Cícero Nogueira dos Santos, and Ruy Luiz Milidiú. 2012. "Latent Structure Perceptron with Feature Induction for Unrestricted Coreference Resolution". In: *Proceedings of the Joint Conference on Empirical Methods in Natural Language*

Proceedings and Conference on Computational Natural Language Learning (EMNLP-CoNLL) - Shared Task.

Finkel, Jenny Rose and Christopher D. Manning. 2009. "Joint Parsing and Named Entity Recognition". In: *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Finkel, Jenny Rose, Trond Grenager, and Christopher Manning. 2005. "Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling". In: *Proceedings of the Association for Computational Linguistics (ACL)*.

Gillick, Dan and Benoit Favre. 2009. "A Scalable Global Model for Summarization". In: *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing*.

Gillick, Dan and Yang Liu. 2010. "Non-Expert Evaluation of Summarization Systems is Risky". In: *Proceedings of the NAACL Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*.

Gimpel, Kevin and Noah A. Smith. 2010. "Softmax-Margin CRFs: Training Log-Linear Models with Cost Functions". In: *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Graff, David et al. 2007. *English Gigaword Third Edition*. Linguistic Data Consortium, Catalog Number LDC2007T07.

Grice, H.P. 1975. "Logic and Conversation". In: *Syntax and Semantics 3: Speech Acts*, pp. 41–58.

Grosz, Barbara J., Scott Weinstein, and Aravind K. Joshi. 1995. "Centering: A Framework for Modeling the Local Coherence of Discourse". In: *Computational Linguistics* 21.2, pp. 203–225. ISSN: 0891-2017.

Guo, Yuhang et al. 2013. "Improving Candidate Generation for Entity Linking". In: *Natural Language Processing and Information Systems*.

Hachey, Ben et al. 2013. "Evaluating Entity Linking with Wikipedia". In: *Artificial Intelligence* 194, pp. 130–150.

Haghighi, Aria and Dan Klein. 2009. "Simple Coreference Resolution with Rich Syntactic and Semantic Features". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

— 2010. "Coreference Resolution in a Modular, Entity-Centered Model". In: *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Hajishirzi, Hannaneh et al. 2013. "Joint Coreference Resolution and Named-Entity Linking with Multi-Pass Sieves". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Hall, David et al. 2014. "Sparser, Better, Faster GPU Parsing". In: *Proceedings of the Association for Computational Linguistics (ACL)*.

He, Zhanying et al. 2012. "Document Summarization Based on Data Reconstruction". In: *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*.

Hendrickx, Iris and Walter Daelemans. 2007. "Adding Semantic Information: Unsupervised Clusters for Coreference Resolution". In: *Workshop notes on Machine Learning for Natural Language Processing*.

Hernault, Hugo et al. 2010. "HILDA: A discourse parser using support vector machine classification". In: *Dialogue and Discourse* 1 (3), pp. 1–33.

Hirao, Tsutomu et al. 2013. "Single-Document Summarization as a Tree Knapsack Problem". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Hobbs, Jerry R. 1977. "Resolving Pronoun References". In: *Lingua*.

— 1979. "Coherence and Coreference". In: *Cognitive Science* 3.1, pp. 67–90. ISSN: 1551-6709. DOI: 10.1207/s15516709cog0301_4. URL: http://dx.doi.org/10.1207/s15516709cog0301_4.

Hoffart, Johannes et al. 2011. "Robust Disambiguation of Named Entities in Text". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Hong, Kai and Ani Nenkova. 2014. "Improving the Estimation of Word Importance for News Multi-Document Summarization". In: *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)*.

Hovy, Eduard et al. 2006. "OntoNotes: The 90% Solution". In: *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL): Short Papers*.

Ji, Heng and Ralph Grishman. 2011. "Knowledge Base Population: Successful Approaches and Challenges". In: *Proceedings of the Association for Computational Linguistics (ACL)*.

Joty, Shafiq et al. 2013. "Combining Intra- and Multi-sentential Rhetorical Parsing for Document-level Discourse Analysis". In: *Proceedings of the Association for Computational Linguistics (ACL)*.

Kazama, Jun'ichi and Kentaro Torisawa. 2007. "Exploiting Wikipedia as External Knowledge for Named Entity Recognition". In: *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

Kikuchi, Yuta et al. 2014. "Single Document Summarization based on Nested Tree Structure". In: *Proceedings of the Association for Computational Linguistics (ACL)*.

Koo, Terry, Xavier Carreras, and Michael Collins. 2008. "Simple Semi-supervised Dependency Parsing". In: *Proceedings of the Association for Computational Linguistics (ACL)*.

Krishnan, Vijay and Christopher D. Manning. 2006. "An Effective Two-Stage Model for Exploiting Non-Local Dependencies in Named Entity Recognition". In: *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Kummerfeld, Jonathan K., Taylor Berg-Kirkpatrick, and Dan Klein. 2015. "An Empirical Analysis of Optimization for Max-Margin NLP". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Lafferty, John D., Andrew McCallum, and Fernando C. N. Pereira. 2001. "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data". In: *Proceedings of the International Conference on Machine Learning (ICML)*.

Lassalle, Emmanuel and Pascal Denis. 2013. "Improving Pairwise Coreference Models Through Feature Space Hierarchy Learning". In: *Proceedings of the Association for Computational Linguistics (ACL)*.

Lee, Heeyoung et al. 2011. "Stanford's Multi-Pass Sieve Coreference Resolution System at the CoNLL-2011 Shared Task". In: *Proceedings of the Conference on Computational Natural Language Learning (CoNLL): Shared Task*.

Lee, Heeyoung et al. 2012. "Joint Entity and Event Coreference Resolution Across Documents". In: *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

Levesque, Hector J., Ernest Davis, and Leora Morgenstern. 2012. "The Winograd Schema Challenge". In: *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*.

Li, Chen, Yang Liu, and Lin Zhao. 2015. "Using External Resources and Joint Learning for Bigram Weighting in ILP-Based Multi-Document Summarization". In: *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Li, Qi and Heng Ji. 2014. "Incremental Joint Extraction of Entity Mentions and Relations". In: *Proceedings of the Association for Computational Linguistics (ACL)*.

Lin, Chin-Yew. 2003. "Improving Summarization Performance by Sentence Compression: A Pilot Study". In: *Proceedings of the International Workshop on Information Retrieval with Asian Languages*.

Lin, Chin-Yew and Eduard Hovy. 2003. "Automatic Evaluation of Summaries Using N-gram Co-Occurrence Statistics". In: *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Lin, Hui and Jeff Bilmes. 2011. "A Class of Submodular Functions for Document Summarization". In: *Proceedings of the Association for Computational Linguistics (ACL)*.

Liu, Fei et al. 2015. "Toward Abstractive Summarization Using Semantic Representations". In: *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Louis, Annie and Ani Nenkova. 2012. "A Coherence Model Based on Syntactic Patterns". In: *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

Louis, Annie, Aravind Joshi, and Ani Nenkova. 2010. "Discourse Indicators for Content Selection in Summarization". In: *Proceedings of the SIGDIAL 2010 Conference*.

Luo, Xiaoqiang. 2005. "On Coreference Resolution Performance Metrics". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Vancouver, British Columbia, Canada.

Luo, Xiaoqiang et al. 2004. "A Mention-Synchronous Coreference Resolution Algorithm Based on the Bell Tree". In: *Proceedings of the Association for Computational Linguistics (ACL)*.

Mani, Inderjeet. 2001. *AutomaticSummarization*. John Benjamins Publishing.

Mann, William C. and Sandra A. Thompson. 1988. "Rhetorical Structure Theory: Toward a Functional Theory of Text Organization". In: *Text* 8.3, pp. 243–281.

Marcu, Daniel. 1998. "Improving Summarization Through Rhetorical Parsing Tuning". In: *Proceedings of the Workshop on Very Large Corpora*.

Martins, Andre and Noah A. Smith. 2009. "Summarization with a Joint Model for Sentence Extraction and Compression". In: *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing*.

Martschat, Sebastian et al. 2012. "A Multigraph Model for Coreference Resolution". In: *Proceedings of the Joint Conference on Empirical Methods in Natural Language Proceedings and Conference on Computational Natural Language Learning - Shared Task*.

McDonald, Ryan. 2006. "Discriminative Sentence Compression With Soft Syntactic Evidence". In: *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)*.

McDonald, Ryan, Koby Crammer, and Fernando Pereira. 2005. "Online Large-margin Training of Dependency Parsers". In: *Proceedings of the Association for Computational Linguistics (ACL)*.

McKeown, Kathleen, Jacques Robin, and Karen Kukich. 1995. "Generating Concise Natural Language Summaries". In: *Information Processing and Management* 31.5, pp. 703–733. ISSN: 0306-4573.

Milne, David and Ian H. Witten. 2008. "Learning to Link with Wikipedia". In: *Proceedings of the Conference on Information and Knowledge Management*.

Mithun, Shamima and Leila Kosseim. 2011. "Discourse Structures to Reduce Discourse Incoherence in Blog Summarization". In: *Proceedings of Recent Advances in Natural Language Processing*.

Nenkova, Ani and Kathleen McKeown. 2011. "Automatic Summarization". In: *Foundations and Trends in Information Retrieval* 5.2?3, pp. 103–233. ISSN: 1554-0669. DOI: 10.1561/1500000015. URL: http://dx.doi.org/10.1561/1500000015.

Ng, Vincent. 2007. "Semantic Class Induction and Coreference Resolution". In: *Proceedings of the Association for Computational Linguistics (ACL)*.

— 2010. "Supervised Noun Phrase Coreference Research: The First Fifteen Years". In: *Proceedings of the Association for Computational Linguistics (ACL)*.

Ng, Vincent and Claire Cardie. 2002. "Improving Machine Learning Approaches to Coreference Resolution". In: *Proceedings of the Association for Computational Linguistics (ACL)*. Philadelphia, Pennsylvania, pp. 104–111.

Nguyen, Khanh and Brendan O'Connor. 2015. "Posterior Calibration and Exploratory Analysis for Natural Language Processing Models". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Nishikawa, Hitoshi et al. 2014. "Learning to Generate Coherent Summary with Discriminative Hidden Semi-Markov Model". In: *Proceedings of the International Conference on Computational Linguistics (COLING)*.

NIST. 2005. "The ACE 2005 Evaluation Plan". In: *NIST*.

Nothman, Joel et al. 2013. "Learning Multilingual Named Entity Recognition from Wikipedia". In: *Artificial Intelligence* 194, pp. 151–175. ISSN: 0004-3702.

Passos, Alexandre, Vineet Kumar, and Andrew McCallum. 2014. "Lexicon Infused Phrase Embeddings for Named Entity Resolution". In: *Proceedings of the Conference on Computational Natural Language Learning*.

Penn, Gerald and Xiaodan Zhu. 2008. "A Critical Reassessment of Evaluation Baselines for Speech Summarization". In: *Proceedings of the Association for Computational Linguistics (ACL)*.

Petrov, Slav et al. 2006. "Learning Accurate, Compact, and Interpretable Tree Annotation". In: *Proceedings of the Conference on Computational Linguistics and the Association for Computational Linguistics (ACL-COLING)*.

Pighin, Daniele et al. 2014. "Modelling Events through Memory-based, Open-IE Patterns for Abstractive Summarization". In: *Proceedings of the Association for Computational Linguistics (ACL)*.

Ponzetto, Simone Paolo and Michael Strube. 2006. "Exploiting Semantic Role Labeling, WordNet and Wikipedia for Coreference Resolution". In: *Proceedings of the North American Chapter of the Association of Computational Linguistics*.

Pradhan, Sameer et al. 2011. "CoNLL-2011 Shared Task: Modeling Unrestricted Coreference in OntoNotes". In: *Proceedings of the Conference on Computational Natural Language Learning (CoNLL): Shared Task*.

Pradhan, Sameer et al. 2012. "CoNLL-2012 Shared Task: Modeling Multilingual Unrestricted Coreference in OntoNotes". In: *Joint Conference on EMNLP and CoNLL - Shared Task*.

Pradhan, Sameer et al. 2014. "Scoring Coreference Partitions of Predicted Mentions: A Reference Implementation". In: *Proceedings of the Association for Computational Linguistics (ACL)*.

Raghunathan, Karthik et al. 2010. "A Multi-Pass Sieve for Coreference Resolution". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Rahman, Altaf and Vincent Ng. 2009. "Supervised Models for Coreference Resolution". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

— 2010. "Inducing Fine-Grained Semantic Classes via Hierarchical and Collective Classification". In: *Proceedings of the International Conference on Computational Linguistics (COLING)*.

— 2011. a. "Coreference Resolution with World Knowledge". In: *Proceedings of the Association for Computational Linguistics (ACL)*.

— 2011. b. "Narrowing the Modeling Gap: A Cluster-Ranking Approach to Coreference Resolution". In: *Journal of Artificial Intelligence Research* 40.1, pp. 469–521. ISSN: 1076-9757.

Ratinov, Lev and Dan Roth. 2009. "Design Challenges and Misconceptions in Named Entity Recognition". In: *Proceedings of the Conference on Computational Natural Language Learning*.

Ratinov, Lev and Dan Roth. 2012. "Learning-based Multi-sieve Co-reference Resolution with Knowledge". In: *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

Ratinov, Lev et al. 2011. "Local and Global Algorithms for Disambiguation to Wikipedia". In: *Proceedings of the Association for Computational Linguistics (ACL)*.

Ratliff, Nathan J., Andrew Bagnell, and Martin Zinkevich. 2007. "(Online) Subgradient Methods for Structured Prediction". In: *Proceedings of the International Conference on Artificial Intelligence and Statistics*.

Recasens, Marta, Matthew Can, and Daniel Jurafsky. 2013a. "Same Referent, Different Words: Unsupervised Mining of Opaque Coreferent Mentions". In: *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL): Human Language Technologies*.

Recasens, Marta, Marie-Catherine de Marneffe, and Christopher Potts. 2013b. "The Life and Death of Discourse Entities: Identifying Singleton Mentions". In: *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL): Human Language Technologies*.

Rooth, Mats et al. 1999. "Inducing a Semantically Annotated Lexicon via EM-Based Clustering". In: *Proceedings of the Association for Computational Linguistics (ACL)*.

Rush, Alexander M., Sumit Chopra, and Jason Weston. 2015. "A Neural Attention Model for Abstractive Sentence Summarization". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Sandhaus, Evan. 2008. "The New York Times Annotated Corpus". In: *Linguistic Data Consortium*.

Sil, Avirup and Alexander Yates. 2013. "Re-ranking for Joint Named-Entity Recognition and Linking". In: *Proceedings of the International Conference on Information and Knowledge Management*.

Singh, Sameer et al. 2011. "Large-scale Cross-document Coreference Using Distributed Inference and Hierarchical Models". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Singh, Sameer et al. 2013. "Joint Inference of Entities, Relations, and Coreference". In: *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction*.

Smith, David A. and Jason Eisner. 2008. "Dependency Parsing by Belief Propagation". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Smith, Noah A. 2011. *Linguistic Structure Prediction*. 1st. Morgan & Claypool Publishers.

Soon, Wee Meng, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. "A Machine Learning Approach to Coreference Resolution of Noun Phrases". In: *Computational Linguistics* 27.4, pp. 521–544.

Soricut, Radu and Daniel Marcu. 2003. "Sentence Level Discourse Parsing Using Syntactic and Lexical Information". In: *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Stoyanov, Veselin et al. 2009. "Conundrums in Noun Phrase Coreference Resolution: Making Sense of the State-of-the-Art". In: *Proceedings of the Association for Computational Linguistics (ACL)*.

Stoyanov, Veselin et al. 2010. "Coreference Resolution with Reconcile". In: *Proceedings of the Association for Computational Linguistics (ACL): Short Papers*.

Thadani, Kapil and Kathleen McKeown. 2013. "Supervised Sentence Fusion with Single-Stage Inference". In: *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP)*.

Tjong Kim Sang, Erik F. and Fien De Meulder. 2003. "Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition". In: *Proceedings of the Conference on Natural Language Learning (CoNLL)*.

Versley, Yannick et al. 2008. "BART: A Modular Toolkit for Coreference Resolution". In: *Proceedings of the Association for Computational Linguistics (ACL): Demo Session*.

Vilain, Marc et al. 1995. "A Model-Theoretic Coreference Scoring Scheme". In: *Proceedings of the Conference on Message Understanding*.

Webster, Kellie and James R. Curran. 2014. "Limited Memory Incremental Coreference Resolution". In: *Proceedings of the Conference on Computational Linguistics (COLING)*.

Winograd, Terry. 1972. *Understanding Natural Language*. Orlando, FL, USA: Academic Press, Inc.

Woodsend, Kristian and Mirella Lapata. 2012. "Multiple Aspect Summarization Using Integer Linear Programming". In: *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

Yang, Xiaofeng, Jian Su, and Chew Lim Tan. 2005. "Improving Pronoun Resolution Using Statistics-Based Semantic Compatibility Information". In: *Proceedings of the Association for Computational Linguistics (ACL)*.

Yao, Limin et al. 2011. "Structured Relation Discovery Using Generative Models". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Yoshida, Yasuhisa et al. 2014. "Dependency-based Discourse Parser for Single-Document Summarization". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Zhang, Tong and David Johnson. 2003. "A Robust Risk Minimization Based Named Entity Recognition System". In: *Proceedings of the Conference on Natural Language Learning (CoNLL)*.

Zheng, Jiaping et al. 2013. "Dynamic Knowledge-Base Alignment for Coreference Resolution". In: *Proceedings of the Conference on Computational Natural Language Learning (CoNLL)*.