A well-written text follows an overall structure, with each sentence following naturally from the ones before and leading into the ones which come afterwards. We call this structure "coherence"; without it, a document becomes a confusing series of non sequiturs. Understanding the principles that make a text coherent is an important goal of natural language processing. These principles can be applied to the design of systems that create new documents, like summaries, or make changes to existing documents.

Coherence is a universal principle of language, but typical approaches to evaluation focus on the application of multidocument summarization. We test the generality of our models by applying them to a new task, chat disentanglement, in which we distinguish independent conversational threads in a crowded chat room. To study this task, we create our own corpus and evaluation metrics, propose a baseline model with basic coherence features, and then test the performance of our own and others' more sophisticated models of local coherence.

We present evidence that despite the significant differences between this task setting and conventional summarization-inspired evaluations, many of these models generalize fairly well, improving over the baseline. Problems with lexicalized models are mostly the fault of insufficient in-domain training data, rather than representing weaknesses in the models themselves. Thus we conclude that many of the same basic principles are used to create coherence throughout English discourse, and that simple local models can be used to describe them. Abstract of "Generalizing Local Coherence Modeling" by Micha Elsner, Ph.D., Brown University, May, 2011.

Generalizing Local Coherence Modeling

by
Micha Elsner
B. S., University of Rochester, 2005
Sc. M. Brown University, 2007

A dissertation submitted in partial fulfillment of the
requirements for the Degree of Doctor of Philosophy in
the Department of Computer Science at Brown University.

Providence, Rhode Island
May, 2011

This dissertation by Micha Elsner is accepted in its present form by
the Department of Computer Science as satisfying the dissertation requirement
for the degree of Doctor of Philosophy.

Date ____________ ______________________________

Eugene Charniak, Director

Recommended to the Graduate Council

Date ____________ ______________________________

Mark Johnson, Reader
Macquarie University, Sydney

Date ____________ ______________________________

Regina Barzilay, Reader
MIT

Approved by the Graduate Council

Date ____________ ______________________________

Peter Weber
Dean of the Graduate School

Acknowledgements

I'd like to thank my advisor Eugene Charniak, whose door was always open, and who gave me room to explore new areas and play long shots. My co-advisor, Mark Johnson, shared his wealth of technical knowledge with me. From my first venture into the area of discourse, I have been indebted to Regina Barzilay, not only for her published research, but for her willingness to discuss the field with me in person.

My career as a computational linguist really began at the University of Rochester. I am grateful to Greg Aist, James Allen, Chris Brown, Dan Gildea, Jeff Runner and especially Mary Swift and Joel Tetreault, for giving a clueless undergraduate the tools and opportunity to do research in this area.

My labmates in the BLLIP research group were the first to hear every idea in this dissertation, and the last to read them over before I submitted them– but I'm just as glad we got to spend time cooking, bike riding, playing games and making conversation. Stu Black, Will Headden, Matt Lease, Rebecca Mason, David McClosky, Deepak Santhanam, Ben Swanson, Jenine Turner and Engin Ural– thanks for everything.

I would also like to thank the members of Machine Learning Reading Group for introducing me to the Church of Bayes and other mysteries, notably Dan Grollman, Sharon Goldwater, Jason Pacheco, Stefan Roth and Frank Wood. I'm grateful to the Cognitive Models Reading Group for letting me pretend to be one of them, especially Dave Buchanan, Liz Chrastil, Adam Darlow, Naomi Feldman, Neal Fox and Phil Fernbach. Outside the department, ten minutes with Shane Bergsma, Jason Eisner, Aria Haghighi, Kristy Hollingshead, Dan Jurafsky, Ani Nenkova, Emily Pitler or Ben Van Durme was usually better than an hour on Google Scholar. And I enjoyed my collaborations with Joe Austerweil, and with Warren Schudy, a great friend in practice as well as in theory.

Li-Juan Cai, Suman Karumuri, Dae-Il Kim, Yuri Malitsky and Matt Wronka made CIT 409 a center of learning, culture and flying nerf that I was proud to call home. Sasha Berkoff, Carleton Coffrin, Aparna Das, Jennie Duggan, Dan L. Klein, Erik Murphy, Allison Smith and Becca Schaffner made it cool to be a geek. Steve Gomez, Brendan Hickey and Yulia Malitskaia didn't have to watch practice talks, but did anyway.

Finally, I must thank my family for bearing with the endless flood of in-jokes, acronyms and other incomprehensible jargon I always seem to end up using to explain just what it is I've spent all this time doing. Thanks to my father, Alan Elsner, for making me a moody intellectual and my mother, Shulamit Elsner, for not asking for such a ridiculous dedication– and спасибо to my brother, Noam Elsner, for being himself.

On the several occasions I had to read someone's dissertation, I always read the acknowledgements section first– it helped to humanize the huge pile of research that inevitably followed. So if you're actually going to read this, thank *you*– and I hope you find what you're looking for in the pages ahead.

The Idea of Order at Key West

She sang beyond the genius of the sea.
The water never formed to mind or voice,
Like a body wholly body, fluttering
Its empty sleeves; and yet its mimic motion
Made constant cry, caused constantly a cry,
That was not ours although we understood,
Inhuman, of the veritable ocean.

The sea was not a mask. No more was she.
The song and water were not medleyed sound
Even if what she sang was what she heard,
Since what she sang was uttered word by word.
It may be that in all her phrases stirred
The grinding water and the gasping wind;
But it was she and not the sea we heard.

For she was the maker of the song she sang.
The ever-hooded, tragic-gestured sea
Was merely a place by which she walked to sing.
Whose spirit is this? we said, because we knew
It was the spirit that we sought and knew
That we should ask this often as she sang.
If it was only the dark voice of the sea
That rose, or even colored by many waves;
If it was only the outer voice of sky
And cloud, of the sunken coral water-walled,
However clear, it would have been deep air,
The heaving speech of air, a summer sound
Repeated in a summer without end
And sound alone. But it was more than that,
More even than her voice, and ours, among
The meaningless plungings of water and the wind,

Theatrical distances, bronze shadows heaped
On high horizons, mountainous atmospheres
Of sky and sea.

It was her voice that made
The sky acutest at its vanishing.
She measured to the hour its solitude.
She was the single artificer of the world
In which she sang. And when she sang, the sea,
Whatever self it had, became the self
That was her song, for she was the maker. Then we,
As we beheld her striding there alone,
Knew that there never was a world for her
Except the one she sang and, singing, made.

Ramon Fernandez, tell me, if you know,
Why, when the singing ended and we turned
Toward the town, tell why the glassy lights,
The lights in the fishing boats at anchor there,
As the night descended, tilting in the air,
Mastered the night and portioned out the sea,
Fixing emblazoned zones and fiery poles,
Arranging, deepening, enchanting night.

Oh! Blessed rage for order, pale Ramon,
The maker's rage to order words of the sea,
Words of the fragrant portals, dimly-starred,
And of ourselves and of our origins,
In ghostlier demarcations, keener sounds.

–Wallace Stevens

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Though many linguistic analyses treat texts at the sentence level, sentences rarely stand on their own in real discourse. The relationships between sentences carry important information: the topic of discussion, the identity of key referents and the rhetorical relationships between propositions are all properties of the text as a whole. To read later sentences without the context provided by earlier ones is a bewildering experience.

The way that these relationships transmit meaning is called *coherence*; a *coherent* discourse is one which is interpretable as a whole. To create a coherent discourse, the language user must first ensure that the content of the discourse forms a consistent rhetorical structure– a logical argument, a narrative, or some other unifying framework– rather than a succession of non-sequiturs. Then they must make sure the surface form of the text indicates what that structure is, a property which Halliday and Hasan (1976) call *cohesion*. This involves marking new information and avoiding repetitions of old information.

For example, consider the short text below:

> So a famous scientist once gave a lecture on astronomy, and afterwards, an old lady approached him. “What you said is rubbish!” she said. “The world is a flat landmass on the back of a giant turtle.” “Ah,” said the scientist. “But what is the turtle standing on?” “A bigger turtle!” she responded. “And that turtle?” the scientist said. “Memory overflow error,” declared the old woman, and vanished.
> –Jenna Moran, “Hitherby Dragons”[1]

The text forms a coherent discourse because it tells a story in chronological order. The first sentence introduces the participants (“a famous scientist”, “an old lady”), and the rest of the discourse explains what they did. Each sentence in the text refers to one or both of the two, often using a pronominal or shortened form, and then gives additional information about their actions. The text below does not form a coherent discourse, despite using many of the same words and phrases.

> “A bigger turtle!” she responded. Astronomy is the study of celestial objects. “And that turtle?” the scientist said. Afterwards, an old lady approached the famous scientist.

[1] “Anecdotes from the Matrix”, November 11, 2004, `imago.hitherby.com/?p=303`

In this thesis, we create and evaluate models of discourse coherence. Such models should be able to distinguish texts like the first example above from texts like the second.

Models of coherence have primarily been used for text-based generation tasks: ordering units of text for multidocument summarization or inserting new text into an existing article. In general, the corpora used consist of informative writing, and the tasks used for evaluation consider different ways of reordering the same set of textual units. But the theoretical concept of coherence goes beyond both this domain and this task setting– and so should coherence models.

This thesis has the broad aim of improving the state of the art in local coherence models. To do so, we design new models, primarily for news text ordering, but also evaluate how well they generalize to a very different domain and task. The specific task setting we choose is chat disentanglement or "threading": separating a transcript of a multiparty interaction into independent conversations. Such simultaneous conversations occur in internet chat rooms, and on shared voice channels such as push-to-talk radio. In these situations, a single, correctly disentangled, conversational thread will be coherent, since the speakers involved understand the normal rules of discourse, but the transcript as a whole will not be– it consists of interleaved utterances which bear no relationship to one another. Thus, a good model of coherence should be able to disentangle sentences as well as order them.

1.1 Contributions

The questions this thesis investigates are *what kinds of features can contribute to modeling local coherence* and *whether local coherence models are general, or specific to summarization-inspired task settings.*

Our answer to the first question is that information about discourse entities gathered from referring expressions can improve local coherence models substantially. We create models based on noun phrase syntax, pronominal coreference, and entity type. *In combination, these models improve document discrimination performance over our baseline, the Entity Grid, from 78% to 87%, yielding a state-of-the-art result.*

Our answer to the second question has two parts. First, we describe a new task setting in which we can measure coherence models' performance outside the summarization setting. Secondly, we report results in this new setting and discuss the implications for model generalization.

We *describe the task of chat disentanglement in detail, annotate a corpus, propose a set of metrics, and measure interannotator agreement. Our corpus is the first to be made publically available for the task, and the first for which annotator reliability is known.* In addition, we create a baseline model. This model *outperforms a variety of naive heuristics, reducing the gap between human and automatic performance by about 30%.*

We next perform chat disentanglement using models of local coherence. We show that *performance on this task can be improved over our baseline model by using coherence*, with results on phone dialogues with different topics increasing from a baseline of 59% to 69% with coherence modeling. This is the first work to explicitly describe disentanglement as a coherence task, and the first to demonstrate improved performance using entity-based and topical features.

We also demonstrate that several popular models of local coherence work on conversational data, and therefore *they are general models of coherence, not limited to a specific setting*. We are the first to evaluate such models on data other than informative writing. For the models which did not perform well in the new setting, we give *an explanation of weaknesses that contribute to their poor performance*.

Before describing these contributions in more detail, we give an overview of related work. In the remainder of this chapter, we first summarize existing techniques and applications for coherence modeling (section 1.2), then of the chat disentanglement problem and related subfields (section 1.3). Chapter 2 is a detailed explanation of the coherence models on which our own work is based, and the experimental tasks conventionally used to evaluate them.

Chapter 3 introduces our own models of coherence. Work in this chapter is primarily based on Elsner et al. (2007) and Elsner and Charniak (2008a).

Chapter 4 describes our setup for disentanglement experiments, our dataset and our baseline model. Most of this work was introduced in Elsner and Charniak (2008b) and expanded in Elsner and Charniak (2010a); our algorithmic experiments were originally presented in Elsner and Schudy (2009).

Chapter 5 presents our experiments using coherence models for disentanglement and their results. Finally, chapter 6 gives the conclusions of this study, and our suggestions for future work.

1.2 Coherence modeling

In this section, we give an overview of previous work on coherence modeling, describing first the techniques typically used, then the tasks to which the models are applied. This section is intended to give a broad view of the field as a whole, without excessive technical detail. The specific models we apply and extend in this thesis are covered in more depth in chapter 2. Many of the ideas covered here have also been applied in related areas: statistical topic modeling, coreference resolution, dialogue understanding, summarization and text pragmatics. Though we recognize the importance of these fields, our discussion here focuses on projects that explicitly construct and apply models of coherence.

1.2.1 Principles of coherence modeling

Researchers distinguish many aspects of coherence, all of which combine to create intelligible discourse. At a high level, a coherent document discusses a sequence of *topics* in a structured way, with each topic tending to occupy a single segment of the text (Hearst, 1997; Galley et al., 2003). This structure is sometimes referred to as *global*, since it is coarse-grained and spans the entire document. Models of global coherence are often constructed as sequential topic models, by augmenting a word clustering model with some kind of temporal structure. Two early examples are Blei and Moreno (2001) and Barzilay and Lee (2004), both using a language model as the topical component and an HMM for the temporal structure. Later research replaces the language modeling component with modern Bayesian topic models following Blei et al. (2001) and the HMM with a variety of structured Bayesian priors. Among these structures are Bayesian Markov models (Purver et al., 2006; Eisenstein and Barzilay, 2008), trees (Eisenstein, 2009) and the Generalized Mallows Model, a prior over permutations (Chen et al., 2009). We discuss our own contribution to the area in section 3.1.

Inside each segment, adjacent sentences link to one another in a *local* structure, in which they share not just a general topic but some specific logical relationship. Local coherence models fall into several types: *rhetorical* models try to explicitly describe the way propositions link up to form an argument or narrative. *Lexical* models capture the tendency for adjacent sentences to use similar vocabulary, since they express related propositions. *Entity-based* models look at the way entities– objects existing in the world– are mentioned in the discourse.

Although the earliest local coherence models, Marcu (1997) and Mellish et al. (1998), used rhetorical structure, few later models have taken this approach. Both these papers focus on text generation; they take input in a non-linguistic format where the relationships between propositions are made explicit. Systems that operate directly on text, however, have to deduce these relationships on their own, and unfortunately, there is currently no method for reliably inferring rhetorical structure in unrestricted text (Sporleder and Lascarides, 2008; Pitler et al., 2008). For texts that may or may not be coherent in the first place (such as those formed by extracting sentences from a multidocument collection), this problem is likely to be even worse. Pitler and Nenkova (2008) obtain some use from rhetorical structure in predicting readability for human-authored documents where rhetorical relations were marked by hand, suggesting that this type of information would indeed be useful if it were available.

Lexical features, on the other hand, require no annotation at all; they are accessible directly from the text. However, their expressive power is limited; in general, lexical methods aim to measure similarity between pairs of sentences adjacent in a text, with little ability to predict the direction of the relationship or how it relates to a higher-level structure. Basic lexical methods restrict themselves to counting repeated words; a variety of metrics are covered in Lapata and Barzilay (2005). Information retrieval techniques, most notably TF-IDF (Jones, 1972), can be used to approximate the importance of each word. More sophisticated models aim to learn associations between *different* words (for instance, “car” is a good context for “tire”, or “brakes” (Prince, 1981)). Foltz et al. (1998) uses a separate dimensionality reduction phase, LSA (Deerwester et al., 1990) to learn words that are similar to one another. Lapata (2003) is the first to learn word associations for local coherence directly, in a manner that respects ordering (so it is able to learn that “tire” is not as good a context for “car” as vice versa). Soricut and Marcu (2006) uses IBM model 1 (Brown et al., 1993) to learn these word associations.

Work on *lexical chains* searches for larger-scale groups of related words which span multiple sentences; rather than relating each sentence to the next, they build larger structures, “chains” of related words which persist throughout a topical segment. As with other lexical methods, lexical chains require some basic method for finding significant repeated or similar words. Galley et al. (2003) uses TF-IDF; Morris and Hirst (1991) describes a variety of methods based on thesauri.

Most of the models presented in this work are entity-based. The key insight of the entity-based approach is that a coherent discourse is about something– some person, object or abstraction that exists (or could exist) external to the text. We call such an object an *entity*. In order to remain coherent, a text should focus on a small set of important entities, explaining what they are and how they interact. We can track the way an entity is used in a text via its *mentions*, pieces of text which refer to it. For instance, an entity-based description of the example text above would say it is about two entities– a scientist and a woman. The three-word English phrase "the famous scientist" is a mention, which refers to the scientist.

Most work on entity-based coherence takes its inspiration from Centering Theory (Grosz et al., 1995), which predicts both repetitions of important entities in adjacent sentences, and also the syntactic forms and positions of references to those entities. The Centering formalism keeps track of a ranked list of entities in each sentence, from which the most prominent NP of the next sentence is selected. The ranking depends on syntactic role (for instance, the subject of a sentence is more prominent than the object of a preposition), but also on other syntactic cues and on the discourse context. Centering has been applied directly in models of coherence (Karamanis et al., 2004; Karamanis et al., 2009); other models use the basic Centering principles as soft constraints or features in a probabilistic framework such as the Entity Grid (Lapata and Barzilay, 2005), which we discuss in more detail in chapter 2.

Work in the entity grid framework has focused mainly on extensions to freer-word-order languages such as German. Filippova and Strube (2007) find that syntactic role information is not useful in German. This finding is confirmed by Cheung and Penn (2010), who show that for German, position in the sentence is more important than syntactic role, because German uses the topology of the sentence to mark information status.

Filippova and Strube (2007) also attempt to relax the Grid's rigid assumption of independence between different entities. They create an extension of the Grid which uses a measure of word similarity to group different entities together, but find that its contribution to their results disappears when they use coreference resolution, suggesting that most of their 'similar entities' are actually coreferent mentions of the same entity. We also attempt to add word similarity to the entity grid in section 3.4; our results suggest this model is useful for disentanglement, but not ordering.

1.2.2 Applications of coherence modeling

The original motivation for computational coherence modeling was text planning; Marcu (1997) points out that a system attempting to explain a set of facts would require a way of ordering those facts, and that this ordering should result in coherent transitions between adjacent units of text. While Marcu (1997) conceived of text planning as one step in generating an article directly from a database, later research on coherence focused on text-to-text generation, and specifically extractive multidocument summarization. In multidocument summarization, the units to be ordered are sentences chosen to represent the content of a large set of documents. Content selection is generally performed as an independent first pass. Jing and McKeown (1999) finds that human summarizers often reorder sentences to create more coherent output, suggesting that poor ordering can compromise readers' understanding; Barzilay et al. (2002) demonstrates this empirically, as does Lapata (2006). Barzilay et al. (2002) also observes that imposing a coherent ordering on the resulting set of sentences is non-trivial, motivating the use of coherence models to solve the problem.

Later work in the summarization framework introduces better models of coherence. This work originates many of the key ideas in current models of local coherence: Barzilay et al. (2002) use chronological information and topical clustering; Lapata (2003) learns word-to-word associations that typify coherent transitions; Lapata and Barzilay (2005) and Barzilay and Lapata (2005) use a syntactic model of information structure.

Research aimed at summarization has also improved inference– the problem of finding a good ordering given a model. Althaus et al. (2004) demonstrate that, for many models, text ordering is NP-hard by reduction to the Traveling Salesman problem, motivating search-based rather than exact algorithms[2]. The earliest proposed search strategy is the genetic algorithm (Mellish et al., 1998). Soricut and Marcu (2006) use A-star search; a major contribution of their approach is the integration of learning and search in order to optimize specific metrics of output quality. Bollegala et al. (2006) uses a ranking approach, ordering pairs of sentences and then using a greedy procedure to extract a final ordering.

As well as ordering whole documents, coherence models can be used to add new content to an existing document– for instance, the biography of a living actor on Wikipedia must be updated whenever they appear in a new movie. Placing this information in the wrong place makes the article harder to understand, while using a coherence model to guide the placement leads to better performance. Chen et al. (2007) and this thesis originate this task simultaneously; Chen et al. (2007) experiment with paragraph-scale incremental updates to Wikipedia.

In addition to text ordering, local coherence models have also been used to score the fluency of texts written by humans or produced by machine. Evaluation of human-authored text can characterize how difficult it is to read. The first use of coherence modeling in this context is Foltz et al. (1998), who use similarity between adjacent sentences to measure readability of encyclopedia articles and a historical text. Recent work by Pitler and Nenkova (2008) models readers' reported difficulty in reading WSJ news articles. Scores for human-authored text are also useful in an educational context, since they can augment human grading for student essays. Miltsakaki and Kukich (2004), Higgins et al. (2004) and Burstein et al. (2010) model various aspects of coherence and find that they are predictive of the grades eventually assigned.

Finally, coherence modeling is an implicit ingredient in many attempts to understand the hidden structure of text. In the course of modeling, researchers often posit relationships between parts of the discourse that are not immediately visible at the surface– for instance, they may break up a long text into segments (Galley et al., 2003; Malioutov and Barzilay, 2006), or link noun phrases that refer to the same real-world entity (Soon et al., 2001). Noticing that a word or phrase fails to fit into a topical segment can aid in detection of malapropisms (Hirst and St-Onge, 1998) or figurative language (Sporleder and Li, 2009). Such hidden variables are often interesting in their own right.

A detailed discussion of either text segmentation or coreference resolution is beyond the scope of this thesis, but we do apply some ideas from these fields in our own work. Text segmentation and similar problems are discussed below, in the section on chat. The relationship between coreference resolution and coherence is discussed in more detail in section 3.3.

1.3 Chat disentanglement

Chat disentanglement (also called "thread detection" (Shen et al., 2006), "thread extraction" (Adams and Martell, 2008), and "thread/conversation management" (Traum, 2004)) is the clustering task of dividing a chat transcript into a set of distinct conversations. As previously stated, a major goal of this thesis is to evaluate local coherence models on the task of disentanglement. From the standpoint of coherence modeling, this serves as a challenging test of domain and task generality. As discussed in the previous section, nearly

[2] One strength of the Mallows model approach of Chen et al. (2009) is that it has an efficient exact procedure for ordering.

all previous work explicitly using coherence models targets newspaper text or other informative writing such as essays, and typical applications are based either on text generation or evaluation.

In this section, we cover previous work on the disentanglement task. One might suspect that, since coherence researchers have not previously worked on chat data, the disentanglement community would have independently invented their own analogues to core coherence modeling concepts. However, they have applied only fairly simplistic lexical and rhetorical models to the task.

Early work on disentanglement was hampered by a lack of standard datasets and metrics, which made model comparison difficult, and by researchers from different communities failing to recognize that they were working on a common problem. This is partly because disentanglement has a variety of potential applications. Researchers in human computer interaction use it to construct better interfaces for online chat and push-to-talk systems. For instance, Smith et al. (2000) constructs a threaded interface for typed chat, in which disentanglement is done manually. Aoki et al. (2003) constructs an automatic system for speech, in which users hear relevant utterances as louder and irrelevant ones as quieter.

On the other hand, researchers from information retrieval and text mining approach disentanglement with the aim of searching through pre-recorded chat transcripts in order to extract information such as question-answer pairs or summaries. This work (beginning with Shen et al. (2006)) expands on approaches from email and newsgroup clustering, which tend to employ content matching approaches, since these work well for longer texts. Email is easier than newsgroups; Yeh and Harnly (2006) find that heuristic information from message headers can be useful, as can content-based matching such as detecting quotes from earlier messages. Wang et al. (2008) analyzes a student discussion group using TF-IDF; their technique is applied to disentanglement by Adams and Martell (2008). Wang and Oard (2009) adapts a message expansion technique from information retrieval to disentanglement.

Finally, researchers on social interactions work on the similar, but not identical task of discovering groups of chatters who talk to one another. Discovering these social groups within a particular chat room is helpful for surveillance (for instance, finding friends of a suspicious individual) or social science research. Camtepe et al. (2005) and Acar et al. (2005) work within this framework. Social network discovery differs from disentanglement because many speakers participate in more than one conversation, speaking to different groups of people at the same time or in short succession.

Because of this fragmentation among fields, early results tend to be difficult to compare to one another. Aoki et al. (2006) construct an annotated speech corpus, but they give no results for model performance, only user satisfaction with their conversational system. Camtepe et al. (2005) and Acar et al. (2005) do give performance results, but only on synthetic data. Adams and Martell (2008) and Shen et al. (2006) publish results on human-annotated data, but without releasing their datasets. This thesis introduces an annotated corpus and a standard set of metrics, and measures interannotator agreement, in chapter 4.

Though their authors had different objectives in mind, most of these disentanglement systems use similar sources of information. The key non-linguistic features, the timing and speaker of each utterance, are essentially universal, though used in different ways. Aoki et al. (2003) and Camtepe et al. (2005) detect the arrival times of messages, and use them to construct an affinity graph between participants by detecting turn-taking behavior among pairs of speakers. (Turn-taking is typified by short pauses between utterances; speakers aim neither to interrupt nor leave long gaps.)

Most researchers add linguistic information in the form of unigram word overlap, a simple measure of lexical coherence. Even getting this to work is not completely straightforward; Acar et al. (2005) deals with all word repetitions on an equal basis, and so degrades quickly in the presence of “noise words” (their term for words which are shared across conversations) to almost complete failure when only $1/2$ of the words are shared. Shen et al. (2006), Adams and Martell (2008) and Wang and Oard (2009) use a more robust representation for lexical features: TF–IDF weighted unigrams, as often used in information extraction. Though this works much better, it is still unsophisticated from a coherence standpoint, using no syntax, topics or relationships between different lexical items.

The most successful extensions to this feature set use hand-written feature detectors for rhetorical markers. Shen et al. (2006) classifies sentences as declarative, interrogative, imperative or conditional, and also counts the number of pronouns, so that they can learn frequencies for common conversational patterns such as question-and-answer. Our own work uses some similar features, along with a list of cue words that mark discourse functions such as greetings (section 4.2). Cue models are common in text segmentation (Hirschberg and Litman, 1993), including models that automatically discover cue phrases from data (Beeferman et al., 1999). However, models that learn cues automatically have not yet been applied to disentanglement.

Attempts to add more complex lexical features have had little success. Adams and Martell (2008) investigate WordNet hypernyms (Miller et al., 1993) as a measure of semantic relatedness. It is unclear from their results whether these features are effective or not. Adams (2008) uses features from an LDA topic model to augment their sentence similarity system. Again, they fail to show an improvement. Since lexical relatedness and topicality are both standard approaches to coherence, their failure to work in this domain is puzzling. We provide our own investigation of this question in chapter 5, where we conclude that topic features are indeed useful for disentanglement, but that the underlying topic model works poorly in the IRC chat domain.

Beyond work that deals directly with chat, there is a large body of work on dialogues and meetings. Meetings do not typically allow multiple simultaneous conversations, so they do not require disentanglement, but they do contain digressions or “subordinate conversations”, in which the speaker addresses someone specific, who is then expected to answer. Some meeting analysis systems attempt to discover where these digressions begin, who is involved in them, and who has the floor when they end; features used in this work are relevant to disentanglement.

The task of automatically determining the intended recipient of an utterance in a meeting is called Addressee Identification. It requires detecting digressions and identifying their participants. Several studies attempt this task. Jovanovic et al. (2006), (2004) perform addressee identification using a complex feature set including linguistic cues like pronouns and discourse markers, temporal information, and gaze direction. They also find that addressee identity can be annotated with high reliability ($\kappa = .7$ for one set and .8 for another). Traum (2004) discusses the necessity for addressee identification and disentanglement in the design of a system for military dialogues involving virtual agents. Subsequent work (Traum et al., 2004) develops a rule-based system with high accuracy on addressee identification.

Another meeting-related task is floor tracking, which attempts to determine which speaker has the floor after each utterance. This task involves modeling the coordination strategies which speakers use to acquire or give up the floor, and so provides a good model of an ongoing conversation. A detailed analysis is given in Chen et al. (2006); Chen (2008) also gives a model for detecting floor shifts. Hawes et al. (2008) use a CRF model to predict the next speaker in Supreme Court oral argument transcripts.

Finally, disentanglement is somewhat similar to the classic *cocktail party* problem in auditory processing, the task of attending to a specific speaker in a noisy room (Haykin and Chen, 2005). Utterance content has some influence on what the listener perceives, but only for extremely salient cues such as the listener's name (Moray, 1959), so cocktail party research does not typically use lexical models.

Chapter 2

Previous Models and Experimental Tasks

In this chapter, we take a closer look at previous work on models of local coherence and the methods normally used to evaluate them. Since we have already given a summary of the field as a whole, our focus here is on the specific systems we implement, evaluate and extend in the remainder of the thesis. For the most part, the work we describe here is not our own. One exception, the task of sentence insertion, is a minor contribution (simultaneously with Chen et al. (2007)), which we incorporate here because it fits naturally into our section on ordering evaluations.

The chapter has four sections. The first (section 2.1) covers models, the second (section 2.2) evaluation methods, the third (section 2.3) statistical significance testing, and the fourth (section 2.4) corpora of informative writing. Empirical results are deferred until the next chapter (3), where we also evaluate our own proposed models. Corpora of chat data are discussed in chapter 4.

2.1 Previous models

2.1.1 Entity grids

The entity grid (Lapata and Barzilay, 2005; Barzilay and Lapata, 2005) is a simple model of entity-based coherence. It is intended to represent the entity-to-entity transitions described by Centering Theory (Grosz et al., 1995). Centering Theory is a description of the interaction between discourse pragmatics and syntax– in particular, the assignment of important entities to certain emphatic positions in the sentence (the "centers"). In coherent discourse, the theory predicts that one position in each new sentence will select its contents from among the centers of the previous sentence. The entity grid, however, does not commit itself to fine-grained rankings of the centers or to a pre-specified transition rule. Instead, it provides a representation which can be constructed from relatively light-weight syntactic analyses, and in which the kinds of rules specified by the theory will be efficiently learnable.

0 [The commercial pilot]$_{\mathbf{O}}$, [sole occupant of [the airplane]$_{\mathbf{X}}$]$_{\mathbf{X}}$, was not injured .
1 [The airplane]$_{\mathbf{O}}$ was owned and operated by [a private owner]$_{\mathbf{X}}$.
2 [Visual meteorological conditions]$_{\mathbf{S}}$ prevailed for [the personal cross country flight for which [a VFR flight plan]$_{\mathbf{O}}$ was filed]$_{\mathbf{X}}$.
3 [The flight]$_{\mathbf{S}}$ originated at [Nuevo Laredo , Mexico]$_{\mathbf{X}}$, at [approximately 1300]$_{\mathbf{X}}$.

Figure 2.1: A section of a document, with syntactic roles of noun phrases marked.

	0	1	2	3
PLAN	-	-	O	-
AIRPLANE	X	O	-	-
CONDITION	-	-	S	-
FLIGHT	-	-	X	S
PILOT	O	-	-	-
LAREDO	-	-	-	X
OWNER	-	X	-	-
OCCUPANT	X	-	-	-

Figure 2.2: The entity grid for figure 2.1. The numeric token “1300” is removed in preprocessing.

The grid represents a document as a matrix with a column for each entity, and a row for each sentence. The entry $r_{i,j}$ describes the syntactic role of entity j in sentence i: these roles are subject (**S**), object (**O**), or some other role (**X**)[1]. In addition there is a special marker (**-**) for entities which do not appear at all in a given sentence. Each entity appears only once in a given row of the grid; if an entity appears multiple times in the same sentence, its grid symbol describes the most important of its syntactic roles: subject if possible, then object, or finally other. An example text is figure 2.1, whose grid is figure 2.2.

This discussion of the grid glosses over the important question of which textual units are to be considered “entities”, and how the different mentions of an entity are to be linked. Although a perfect solution to the problem would use coreference resolution to link arbitrary mentions to the same entity, while discarding noun phrases which do not correspond to an entity, this approach is not generally the one adopted. The problem with it is that coreference resolution is far from perfect, and tends to work even more poorly on disordered or incoherent documents, introducing more errors than it fixes (Barzilay and Lapata, 2005). Instead, implementations of the grid tend to treat all noun phrases as mentions and perform heuristic coreference resolution by linking mentions which share a head noun. Detailed discussions of this heuristic are given in Poesio et al. (2005) and Elsner and Charniak (2010b).

Given a grid representation of a document, we must build some kind of model that distinguishes coherent from incoherent text. The two main approaches are generative (Lapata and Barzilay, 2005) and discriminative (Barzilay and Lapata, 2005). Since this thesis uses only generative grid systems, this is the approach we choose to present[2].

[1] Roles are determined heuristically using trees produced by the parser of (Charniak and Johnson, 2005). Following previous work, we slightly conflate thematic and syntactic roles, marking the subject of a passive verb as **O**.

[2] To give a quick overview of the alternative, the discriminative approach decomposes the grid into a set of discrete events, then uses the frequency of each event type as a feature for making decisions in a particular evaluation setting. An important difference between the discriminative and generative grids is this dependence on the evaluation task; while generative grids are trained from raw data, discriminative systems require a task-specific contrast set.

The generative grid breaks down the grid into a series of independent transition events for which probabilities can be estimated. First, the probability of a document is defined as $P(D) = P(S_i..S_n)$, the joint probability of all the sentences. Sentences are generated in order conditioned on all previous sentences:

$$P(D) = \prod_i P(S_i|S_{0..(i-1)}). \tag{2.1}$$

We make a Markov assumption of order h (usually, $h = 2$) to shorten the history. We represent the truncated history as $\vec{S}^h_{i-1} = S_{(i-h)}..S_{(i-1)}$.

Each sentence S_i contains a set of entities, E_i, and their corresponding syntactic roles R_i (**SOX-**); the remaining words are ignored. Thus, generating sentence i is equivalent to predicting $r_{i,j}$ for each entity e_j. We assume each entity e_j appears in sentences and takes on syntactic roles independent of all the other entities, so the conditioning information for $r_{i,j}$ is the history of the specific entity e_j, which we denote $\vec{r}^h_{(i-1),j} = r_{(i-h),j}..r_{(i-1),j}$ (plus potentially some extra, entity-specific information). In particular, the probability of S_i is:

$$P(S_i|S^h_{i-1}) = \prod_j P(r_{i,j}|\vec{r}^h_{(i-1),j}) \tag{2.2}$$

Notice that the set of potential entities j ranges over all entities in the document, including those which do not appear in sentence i. For instance, in figure 2.2, the probability of S_3 with horizon 1 is the product of $P(\mathbf{S}|\mathbf{X})$ (for `FLIGHT`), $P(\mathbf{X}|-)$ (for `LAREDO`), and likewise for each other entity, $P(-|\mathbf{O}), P(-|\mathbf{S}), P(-|-)^4$.

Learning in the generative framework consists of estimating the conditional density $P(r_{i,j}|\vec{r}^h_{(i-1),j})$. In practice, most implementations follow Barzilay and Lapata (2005) in also conditioning on a measure of *salience*, which ideally should measure the importance of a particular entity to the document. The usual way of representing this is to condition on the number of times the entity occurs throughout the document (capped at 4 occurrences, so that entities that occur more frequently are assigned salience 4). Unfortunately, this way of representing salience renders the model probabilistically deficient, since the model conditions on a particular noun occurring e.g. 2 times, but assigns nonzero probabilities to documents where it occurs 3 times. In our experiments, this does not seem to cause problems in practice.

Even with the addition of a salience term, the number of different conditioning contexts is fairly small ($4^h * 4$ for a maximum salience value of 4; this produces 64 contexts when $h = 2$). Thus the maximum likelihood estimator is quite reliable even without sophisticated smoothing or large amounts of data.

2.1.2 IBM-1

The goal of models with word association parameters is a simple one: to find pairs of words v, w for which v in the current sentence predicts w in the next. As stated in the introduction, it is possible to do this using generic word-similarity measurements, either derived from a resource like WordNet or from dimensionality reduction on a matrix of word coocurrences, but these methods may not produce associations suitable for coherence modeling– in particular, measurements of similarity are symmetric, while word associations are not.

NULL	philippine	rulings	decisions
it	manila	rulings	choices
i	philippine	justices	decision
he	ferdinand	court	consequences
they	filipinos	courts	basis
you	fidel	decisions	courts

Table 2.1: Top five translations of four words, as learned by IBM-1 on NANC data.

Lapata (2003) gives the first model for learning word associations for coherence from data. Her model gives the probability of a sentence W with words w_i following a sentence V with words V_j as:

$$P(W) = \prod_i \prod_j p(w_i|v_j) \tag{2.3}$$

This model explains each word w as produced by a predecessor word v, although which word in the previous sentence should serve as predecessor for each w is unknown. Lapata describes the nested products as the result of assuming the words w_i are generated independently, though in fact, since each one appears in j different terms, this assumption is not valid under the model.

Soricut and Marcu (2006), inspired by a personal communication from Kevin Knight, use IBM model 1 (Brown et al., 1993) to learn word-to-word associations. IBM uses hidden variables to obtain a true generative model with the same lexical parameterization as Lapata (2003). In a direct comparison, they obtain superior results, though since they use additional features and a better search algorithm, it is unclear whether this particular aspect of their model is responsible.

IBM solves the problem that we do not know the correct predecessor word v_j for each w_i by introducing a hidden alignment variable $a_i \in [0..j]$ which indicates precisely which previous word is responsible for each current word. (The dummy index 0 indicates a "null" word assumed to be available as a predecessor in every sentence.) The prior distribution over a_i is assumed to be uniform, so that $P(a_i = j) = \frac{1}{|V|+1}$. This yields the following probability for a sentence (marginalizing out the a_i):

$$P(W) = \prod_i \frac{1}{|V|+1} \sum_j p(w_i|v_{a_i}) \tag{2.4}$$

Because IBM-1 has hidden variables, its parameters cannot be learned by direct estimation; conventionally, EM or variational EM are used instead.

Our implementation of IBM-1 produces and conditions on nouns and verbs only; other parts of speech are discarded. It is most effective when trained on large datasets, so we use a distributed E-step in our EM calculations.

For illustration, we show some "translation" parameters learned by our system (Table 2.1). The NULL word tends to produce pronouns, since they appear in most sentences, regardless of the topic of the preceding sentence. In general, there is a bias toward repetition– nearly all words are well-translated as themselves. The parameters are indeed asymmetric: while the word "rulings" is often followed by "decisions", "decisions" is not often followed by "rulings".

2.1.3 Model combination with discriminative mixtures

Presented with a variety of different models for local coherence, all considering different sources of information, an obvious question is whether they can be combined to obtain improved results. In some cases, the best strategy for model combination is full-scale joint inference; we consider such a strategy for a specific case in section 3.1. Creating a joint model requires in-depth analysis of the component models and tends to complicate learning and inference.

At the opposite extreme, one can simply assume the models are independent and multiply the probabilities. For models that are indeed (nearly) independent, this is typically a good strategy, but it can be problematic if for models which are closely correlated because they consider the same evidence (for instance, the entity grid and IBM-1 both predict that a noun appearing in sentence i increases the chance of its recurring in $i + 1$). This leads to the same piece of evidence being "double-counted" because of its influence on both models. Another problem occurs when the models make correct decisions, but with the wrong confidences. Because different models approximate reality in different ways, the range of probabilities they produce are often incomparable[3], and the product tends to agree with more confident rather than more reliable models.

The log-linear mixture framework for coherence models was introduced by Soricut and Marcu (2006). It retains the ease of use of the product approach, but without requiring independence assumptions. Each model P_i is assigned a weight λ_i, and the combined score $P(d)$ is proportional to:

$$\sum_i \lambda_i log(P_i(d))$$

This corresponds to a product of exponentially weighted probabilities. Unfortunately, it cannot be used as a generative model, because the normalization constant (which sums over all documents) is intractable. However, the weights λ can be learned discriminatively, by maximizing the probability of d relative to a task-specific contrast set. For ordering experiments, the contrast set is a single random permutation of d; we explain the training regime for disentanglement below, in section 5.4.1.

2.2 Ordering tasks

While the generic framework of sentence ordering is motivated by summarization and other text generation tasks, researchers quickly developed an arsenal of surrogate tasks and metrics with which to evaluate their coherence models. There are several reasons for this development. Direct evaluation of summaries by humans is expensive, and mechanical evaluations are not entirely reliable (Sjöbergh, 2007; Donaway et al., 2000); moreover, any attempt to evaluate different text ordering strategies via the summaries they produce will be dependent on the underlying sentence selection mechanism. Summarization also introduces new complexities for an idealized coherence model, since the sentences extracted probably do not actually form a really coherent document in *any* order, even if some orderings are significantly better than others.

[3] For instance, the IBM model assumes all words w_i in a sentence are independent of one another, which is demonstrably false; thus, its probabilities for generated sentences are typically underestimates.

Karamanis et al. (2009) (partly summarizing older contributions (Karamanis et al., 2004)) describes single-document ordering tasks as a proxy. In these tasks, the set of sentences consists of a single document, and a good model is assumed to prefer their original order to any reordering. Here we rely on the assumption that human-authored documents are coherent, and that permutations of their sentences are less so; some specific tasks and evaluation metrics make further assumptions, which we discuss below. Lapata (2006) verifies these assumptions for the case of short documents and random orderings; she shows that in this case, orderings that diverge more from the originals are assigned lower coherence by human judges. While longer texts do sometimes allow the movement of sections or paragraphs as units, the assumption is still true on the whole. To demonstrate the unlikeliness that a natural text could have two very different coherent orderings, we exhibit some purposefully written texts with this property in appendix B.

In general, ordering tasks lie on a continuum between the trivial and the intractable. Tasks that are too easy fail to distinguish between good and bad models (since even simple models quickly approach ceiling performance). For tasks that are computationally difficult, the results of any experiment depend heavily on what approximation or heuristic search is used to obtain the results (the number of *search errors*). This can render comparisons of results ineffective (since it is difficult to distinguish model performance from search performance), and slow down the pace of research (since several days may be necessary to evaluate a given model).

In the discussion below, we present ordering tasks arranged from computationally hardest to easiest, including our own contribution, *insertion*, which was simultaneously proposed by (Chen et al., 2007).

2.2.1 Ranking

The ranking task is among the earliest considered, and was introduced by Karamanis et al. (2004). In this task, we use the model to evaluate all permutations of the sentences in a document, and report the percentage judged *better*, *worse* and *the same as* the original order. Making the assumption that the original order is the most coherent, we would like to find all other orders *worse*, or at least *the same*.

The ranking task in its exact form is tractable only for very short documents, since the number of permutations grows as $n!$. For longer documents, it is necessary to resort to Monte Carlo approximations, which we discuss below under the name *discrimination*.

2.2.2 Sentence Ordering

In the sentence ordering task, (Lapata, 2003; Barzilay and Lee, 2004; Barzilay and Lapata, 2005; Soricut and Marcu, 2006), we view a document as an unordered bag of sentences and try to find the ordering of the sentences which maximizes coherence according to our model. Unfortunately, finding the optimal ordering according to a probabilistic model with local features is NP-complete and non-approximable (Althaus et al., 2004), by reduction to the Traveling Salesman Problem: take each sentence as a city and $dist(S_1, S_2) \equiv P(S_1 \text{ follows } S_2)$; this produces a TSP with asymmetric distances.

For a Markov model which factors neatly into $P(S_1 \text{ follows } S_2)$, it is possible to find a good relaxation to use as a heuristic for A^* search (Soricut and Marcu, 2006). For other classes of model, however, local search techniques such as simulated annealing are necessary.

To evaluate the quality of the orderings proposed by a system, it is common to use *Kendall's* τ, a measurement of the number of pairwise swaps needed to transform a proposed ordering into the original document, normalized to lie between -1 (reverse order) and 1 (original order).

Lapata (2006) shows that, for short documents, τ corresponds well with human judgements of coherence and reading times. A problem with τ is that it cannot distinguish between proposed orderings of a document which disrupt local relationships at random, and orderings in which paragraph-like units move as a whole.

For this reason, Bollegala et al. (2006) introduces the Average Continuity (AC), which measures how many continuous long substrings of sentences are placed together in the selected ordering. AC is similar to the BLEU metric for machine translation. It defines the precision of substrings of length n as:

$$P_n = \frac{m}{N - n + 1} \tag{2.5}$$

N is the length of the document and m is the number of continuous subsequences of length n. The AC is defined as the average of $log(P_i)$ for i between 2 and 4.

2.2.3 Insertion

Insertion is motivated by a slightly different text generation setting, the problem of updating an existing document with new information. Chen et al. (2007) work directly on this task, predicting the target paragraphs of edits to Wikipedia articles from their content. Like other ordering tasks, however, insertion can be abstracted into an experimental evaluation for coherence models, which is useful even in domains without text update information.

In our version of insertion, given a document, we remove each sentence in turn, then find the point of insertion which yields the highest coherence score. The model is scored correct if it reinstates the sentence in its original position. Our reported score is the average fraction of sentences correct per document (averaged over documents, not sentences, so that longer documents do not disproportionally influence the results).

Besides any practical applications, insertion has two properties that make it well-suited to evaluation of models. First, a whole document can be evaluated exactly in n^2 time, which, though it can be significant, is still much faster than ordering and does not involve a potentially error-prone search. Second, it grows more difficult as documents lengthen, unlike ranking (and its approximate version, discrimination), which grow easier. For a document with n sentences, each sentence insertion is an n-way decision, and so a random baseline is expected to get one sentence correct and all the rest wrong, regardless of document length. Thus, random performance scales as $1/n$, going to zero as length increases.

2.2.4 Discrimination

The discriminative test is introduced in Barzilay and Lapata (2005). In this task, the experimenter generates random permutations of a test document and measures how often the score of a permutation is higher than that of the original document.

For small numbers of random permutations, discrimination is computationally easy. For large numbers (approaching $n!$ for a document of length n), it approximates ranking. In our experiments, we follow Barzilay and Lapata (2005) and use 20 permutations per document.

Unlike ordering, which tries to measure how *close* the model's preferred orderings are to the original, this measurement assesses how *many* orderings the model prefers. Whether the orders preferred by the model bear any resemblance to the original is immaterial as long as they are few. For instance, a model defined in terms of symmetric affinities between sentences (such as cosine distances) cannot distinguish a document from its reverse. Such a model may perform well on discrimination while obtaining very low τ scores.

Discrimination becomes easier for longer documents, since the average random permutation grows less similar to the original document.

2.3 Significance testing

In this thesis, we perform significance tests of ordering systems using the Wilcoxon rank-sum test, also known as the Mann-Whitney U test[4]. This is a test which tests, for paired observations x, y, whether the median of x is identical to the median of y; if this hypothesis may be rejected, we call our result statistically significant. We choose the rank-sum test because it does not require a parametric assumption about the distribution of x or y, unlike the more popular T-test, which requires a Gaussian distribution; the test requires only the assumption that, under the null hypothesis of an equal median, the distributions of scores for both models are equal. A few earlier experiments were tested with the Wilcoxon signed-rank test instead; this test requires the additional assumption that the distributions are symmetric about their medians. Where this test is used, we point it out in a footnote.

In order to avoid making distributional assumptions, the rank-sum test uses order statistics, not moments, so it does not exactly correspond to the scores we report. We show mean scores in our tables, but our tests measure median scores. This means that the test is unable to detect effects which skew the mean more than the median. In other words, if system A scores 8,9,10,11,12 and system B scores 8,9,10,19,20, system B will have higher scores in our results, but they will not be judged significant because they do not change the median[5].

To perform the test, we must have a large number of x, y pairs which are i.i.d. given the model we are testing. We assume independence among the scores for separate documents. Thus, to test the significance of a discrimination experiment, we use each model to compute the number of correct discriminations over 20 permutations for each document. This gives us a set of independent scores, each one a number between 0 and 20.

For insertion, we follow the same procedure, except that instead of a score between 0 and 20, we have a real-numbered score between 0 and 1, corresponding to the proportion of perfect insertions in each document.

Significance tests in the thesis are reported at two p-values, p=.001 (indicated in tables by ‡) and p=.05

[4] We use the built-in implementation in MATLAB.

[5] This means our significance test for discrimination experiments does not correspond directly to either discrimination accuracy or F-score, but we mark significance of discrimination decisions next to the accuracy figure throughout the thesis.

(indicated by †). Where † appears, the test is *not* significant at p=.001. The exact comparison being drawn is described in the table caption.

For disentanglement, we are unable to report significance, because our metrics for that task (discussed in section 4.2.2) are averaged over utterances within a document, rather than documents. Utterances within the same document are not independent, although for utterances far enough apart, we expect them to behave roughly as if they are, which is why we believe our scores are meaningful. However, this deviation from independence is enough to render the p-values computed by our tests invalid, and we do not know of an alternate procedure for non-i.i.d. data.

2.4 Corpora

2.4.1 Airplane

Early experiments on coherence used the AIRPLANE corpus, a collection of documents describing airplane crashes taken from the database of the National Transportation Safety Board (Barzilay and Lee, 2004; Barzilay and Lapata, 2005; Soricut and Marcu, 2006). The corpus is divided into 100 training and 100 test documents. The AIRPLANE documents have some advantages for coherence research: they are short (11.5 sentences on average) and quite formulaic, which makes it easy to find lexical and structural patterns. On the other hand, they have some notable oddities. 46 of the training documents begin with a standard preamble: “This is preliminary information, subject to change, and may contain errors. Any errors in this report will be corrected when the final report has been completed,” which gives models with enough features to recognize these particular sentences a sizeable advantage. Other documents, however, begin abruptly with no introductory material whatsoever, and sometimes without even providing references for their definite noun phrases; one document begins: “At V1, the DC-10-30’s number 1 engine, a General Electric CF6-50C2, experienced a casing breach when the 2nd-stage low pressure turbine (LPT) anti-rotation nozzle locks failed.” Even humans might have trouble identifying this sentence as the beginning of a document.

2.4.2 WSJ

For later experiments, we use the Penn Treebank (Marcus et al., 1993), which contains typical newspaper text. It contains longer, more complex articles than AIRPLANE, averaging 22 sentences, as opposed to 11.5 for AIRPLANE. Individual sentences are also longer and more syntactically complicated.

For sentence ordering models, we use sections 2-13 of the Penn Treebank for training and development, and sections 14-24 (1004 documents total) for testing.

For coreference, we use an annotatation of sections 0 and 1 done by Ge et al. (1998), which marks all personal pronouns as anaphoric or non-anaphoric and indicates all the NPs in the same document which are coreferent with them[6]. Section 0 is used for development and section 1 for testing. The test set has 1119 personal pronouns, of which 246 are non-anaphoric.

[6] This dataset is distributed along with the software for (Charniak and Elsner, 2009).

For all experiments on WSJ, we use the gold parse trees. If additional unlabeled data is needed, we also train on news text from the North American News Corpus (NANC), parsed as part of McClosky et al. (2006), which is drawn from the LA Times.

Chapter 3

Models of entity-based coherence

This chapter presents our work on directly modeling entity-based coherence. The major contribution of this chapter is a *state-of-the-art result on ordering* WSJ, obtained in subsection 3.4.1.

The standard starting point for entity-based coherence is the entity grid, which we explained in detail in the previous chapter. In section 3.1, we point out some weaknesses of the entity grid model, namely its inability to describe global topical structure or lexical co-occurrences. Using a Bayesian framework, we jointly model these aspects of coherence along with entities. Although learning and inference for the model are not scalable, on short documents, it performs well on several ordering tasks.

In the remainder of the chapter, we introduce several models which look more at *what kind* of entity a given mention refers to. In the standard entity grid, all entities are essentially created equal; there are no particular differences between “Barack Obama” and “ten miles”, unless they happen to be mentioned different numbers of times, and this problem is exacerbated by the use of same-head coreference, which lumps together different “miles” expressions to create a pseudo-entity that appears much more salient than it really is. This simplicity is a strength of the model, since it requires little in the way of support technology and transfers well between domains, but also a weakness, in that it ignores information that can be profitably exploited.

We incorporate some of this information using ideas from coreference resolution. In a sense, coreference solves a problem opposite from the entity grid– the grid model takes coreference as given and evaluates the structure of the document, while coreference takes the document as given and evaluates the relationships between the NPs. Therefore, it makes sense to use some of the same features for both. Section 3.2 describes a coreference-inspired model of NP structure for coherence, which distinguishes between mentions that introduce a new entity and those that refer to an established one. In section 3.3, we propose the use of a generative pronoun coreference resolver as a coherence model. Section 3.4 describes two extensions to the entity grid, one which uses entity-specific features, and another which uses a topic model.

3.1 Topic-Based Model

The entity grid is a purely local model, and moreover it relies on a particular set of local features which capture the way adjacent sentences tend to share lexical choices. Its lack of any global structure makes it impossible for the model to recover at a paragraph boundary, or to accurately guess which sentence should begin a document. Its lack of lexicalization, meanwhile, renders it incapable of learning dependences between pairs of words: for instance, that a sentence discussing a crash is often followed by a casualty report.

We remedy both these problems by extending our model of document generation. Like Barzilay and Lee (2004), we learn an HMM in which each sentence has a hidden topic q_i, which is chosen conditioned on the previous state q_{i-1}. The emission model of each state is an instance of the relaxed entity grid model as described in Elsner et al. (2007)[1], but in addition to conditioning on the role and history, we also condition on the state.

A full explanation of the mathematics of the model is given in appendix A. However, the basic idea of the model is simple: each sentence has a hidden topic, from which all words in the sentence which do not refer to entities are produced. Words that do refer to entities are produced from a topic-specific entity grid– a model which has high probability of mentioning both entities specific to the topic at hand, and recently mentioned entities (and an even higher probability of mentioning entities which are both topically appropriate and currently under discussion). These topic-specific grids are tied together via a common prior which prefers to mention recently mentioned entities, regardless of topic.

To represent the HMM itself, we adapt the non-parametric HMM of Beal et al. (2001). This is a Bayesian alternative to the conventional HMM model learned using EM, chosen mostly for convenience. Our variant of it, unlike Beal et al. (2001), has no parameter γ to control self-transitions; our emission model is complex enough to make it unnecessary.

The actual number of states found by the model depends mostly on the backoff constants, the θs (and, for Pitman-Yor processes, $discount$s) chosen for the emission models (the entity grid, non-entity word model and new noun model), and is relatively insensitive to particular choices of prior for the other hyperparameters. As the backoff constants decrease, the emission models become more dependent on the state variable q, which leads to more states (and eventually to memorization of the training data). If instead the backoff rate increases, the emission models all become close to the general distribution and the model prefers relatively few states. We train with interpolations which generally result in around 40 states.

Once the interpolation constants are set, the model can be trained by Gibbs sampling. We also do inference over the remaining hyperparameters of the model by Metropolis sampling from uninformative priors. Convergence is generally very rapid; we obtain good results after about 10 iterations. Unlike Barzilay and

[1] We designed the relaxed entity grid as an early attempt to create correlations between the columns of the standard grid model, which we denoted "naive"– by treating all entities as independent, the standard grid is capable of generating sentences with no entities at all, and other syntactic anomalies. Unfortunately, our method for solving this problem proved to be ineffective– but even more unfortunately, it worked *better* on the AIRPLANE dataset, because it was capable of recognizing and correctly placing the preamble sentence "This is preliminary information..." which begins 40% of the AIRPLANE sentences. When the paper was published, we thought the improved results were a consequence of the statistics, but further analysis and tests on WSJ convinced us that we had been mistaken. Though the relaxed grid proved to be a mistake, however, the comparison of the entity grid with the HMM is still an interesting one, which we feel reveals important subtleties in the differences between local and global models, and in the different sensitivities of the tasks used to evaluate them.

	τ	Discr. (%)
(Barzilay and Lapata, 2005)	-	90
(Barzilay and Lee, 2004)	.44	74
(Soricut and Marcu, 2006)	**.50**	-
Topic-based	**.50**	**94**

Table 3.1: Results for AIRPLANE test data. Statistical significances are not available for these models.

	τ	Discr. (%)
Naive Entity Grid	.17	81
Relaxed Entity Grid	.02	87
Topic-based (naive)	.39	85
Topic-based (relaxed)	**.54**	**96**

Table 3.2: Results for 10-fold cross-validation on AIRPLANE training data.

Lee (2004), we do not initialize with an informative starting distribution.

When finding the probability of a test document, we do not do inference over the full Bayesian model, because the number of states, and the probability of different transitions, can change with every new observation, making dynamic programming impossible. Beal et al. (2001) proposes an inference algorithm based on particle filters, but we feel that in this case, the effects are relatively minor, so we approximate by treating the model as a standard HMM, using a fixed transition function based only on the training data. This allows us to use the conventional Viterbi algorithm. The backoff rates we choose at training time are typically too small for optimal inference in the ordering task. Before doing tests, we set them to higher values (determined to optimize ordering performance on held-out data) so that our emission distributions are properly smoothed.

3.1.1 Experiments

We evaluate this model on the AIRPLANE corpus, reporting both discrimination and ordering tasks (Table 3.1). For details, see section 2.2. Performance for Barzilay and Lee (2004) was calculated on our test permutations using the code at `http://people.csail.mit.edu/regina/code.html`. The result for Soricut and Marcu (2006) on discrimination is missing, because they do not report results on this task, except to say that their implementation of the entity grid performs comparably to (Barzilay and Lapata, 2005).

Since the ordering task requires a model to propose the complete structure for a set of sentences, it is very dependent on global features (Table 3.2). To perform adequately, a model must be able to locate the beginning and end of the document, and place intermediate sentences relative to these two points. The entity grid is best at deciding which sentences go with *one another*, but it has no good way of deciding how to order different topical segments. This is why the entity grid model has τ of approximately 0, meaning its optimal orderings are essentially uncorrelated with the correct orderings[2]. The HMM content model of (Barzilay and Lee, 2004), which does have global structure, performs much better on ordering, at τ of .44. However, local features can help substantially for this task, since models which use them are better at assembling related sentences into segments which can be ordered by the HMM. Using both sets of features, our topic-based

[2] Barzilay and Lapata (2005) do not report τ scores.

model achieves state of the art performance ($\tau = .5$) on the ordering task, comparable with the mixture model of (Soricut and Marcu, 2006)[3]

The need for good local coherence features is especially clear from the results on the discrimination task. Permuting a document may leave obvious "signposts" like the introduction and conclusion in place, but it almost always splits up many pairs of neighboring sentences, reducing local coherence. (Barzilay and Lee, 2004), which lacks local features, does quite poorly on this task (74%), while our model performs extremely well (94%).

Our observation that the local model gets essentially chance τ scores also indicates an issue with the τ metric– although the local model is good at assembling related sentences nearby to one another, if it puts these sentences in the wrong part of the document, it gets no credit. For instance, when ordering a document consisting of two paragraphs, A and B, intuitively, we would prefer our output to look like B, A rather than $A_1, B_1, A_2, B_2 \ldots$, but the τ scores for these two permutations will be roughly equal. For this reason, we recommend that ordering studies also report a metric that looks at groups of sentences, such as the Average Continuity (Bollegala et al., 2006).

Our combined model uses only entity-grid features and unigram language models, a strict subset of the feature set of (Soricut and Marcu, 2006). Their mixture includes an entity grid model and a version of the HMM of (Barzilay and Lee, 2004), which uses n-gram language modeling. It also uses a model of lexical generation based on the IBM-1 model for machine translation, which produces all words in the document conditioned on words from previous sentences. In contrast, we generate only entities conditioned on words from previous sentences; other words are conditionally independent given the topic variable. It seems likely therefore that using our model as a component of a mixture might improve on the state of the art result.

The main issue with this model, however, is its lack of scalability. There are two reasons the model does not scale. One is computational: learning and inference both require quadratic-time HMM operations. Considering that each different permutation to be scored requires us to run full-scale inference, the insertion task (which scores a quadratic number of permutations) takes $O(n^4)$ time, and ordering based on search becomes prohibitive.

The second problem is with the HMM as a representation. The problem is that it takes a very fine-grained view of topical transitions; news articles from less restrictive domains than AIRPLANE have larger-scale structures like paragraphs, and are not well-modeled by sentence-to-sentence transitions. Later global models (Chen et al., 2009; Eisenstein, 2009) attempt to capture these large-scale structures while retaining a small enough parameter space to make learning possible.

In the rest of the chapter, however, we take another direction, focusing on improving our local models rather than globalizing them. By doing so, we admittedly miss some aspects of document coherence, but as we will show, our tools are still capable of solving some relatively difficult problems, and doing so efficiently as well.

[3] As explained in section 2.3, we cannot compute statistical significance using averages alone, but would require model-by-model scores. Thus, unfortunately, we cannot compute it here, since for the prior work, we have only averages.

3.2 Discourse-new model

The entity grid, and the extensions of the previous section, treat all mentions in a text as equivalent. However, a mention of an entity contains more information than just its head and syntactic role. The referring expression itself contains discourse-motivated information distinguishing familiar entities from unfamiliar and salient from non-salient. These patterns have been studied extensively, by linguists (Prince, 1981; Fraurud, 1990) and in the field of coreference resolution. In this section, we begin applying these ideas from coreference to our coherence models.

In this section we present a model which distinguishes discourse-new from discourse-old noun phrases, using features based on Uryupina (2003). Discourse-new NPs are those whose referents have not been previously mentioned in the discourse. As noted by studies since Hawkins (1978), there are marked syntactic differences between the two classes.

In the task of discourse-new classification, the model is given a referring expression (as in previous work, we consider only NPs) from a document and must determine whether it is a first mention (*discourse-new*) or a subsequent mention (*discourse-old*). Features such as full names, appositives, and restrictive relative clauses are associated with the introduction of unfamiliar entities into discourse (Hawkins, 1978; Fraurud, 1990; Vieira and Poesio, 2000). Classifiers in the literature include (Poesio et al., 2005; Uryupina, 2003; Ng and Cardie, 2002a). The system of Nenkova and McKeown (2003) works in the opposite direction. It is designed to rewrite the references in multi-document summaries, so that they conform to the common discourse patterns.

We construct a maximum-entropy classifier using syntactic and lexical features derived from Uryupina (2003), and a publicly available learning tool (Daumé III, 2004). Our system scores 87.4% (F-score of the *disc-new* class on the MUC-7 formal test set); this is comparable to the state-of-the-art system of Uryupina (2003), which scores 86.9[4].

To model coreference with this system, we assign each NP in a document a label $L_{np} \in \{new, old\}$. Since the correct labeling depends on the coreference relationships between the NPs, we need some way to guess at this; we apply the *same-head heuristic*; that is, we take all NPs with the same head to be coreferent, as in the non-coreference version of (Barzilay and Lapata, 2005). Unfortunately, this represents a substantial sacrifice; as Poesio and Vieira (1998) show, only about 2/3 of definite descriptions which are anaphoric have the same head as their antecedent. We will return to the same-head heuristic in the following chapter. We then take the probability of a document as $\prod_{np:NPs} P(L_{np}|np)$.

We must make several small changes to the model to adapt it to this setting. For the discourse-new classification task, the model's most important feature is whether the head word of the NP to be classified has occurred previously (as in Ng and Cardie (2002a) and Vieira and Poesio (2000)). For coherence modeling, we must remove this feature, since it depends on document order, which is precisely what we are trying to predict. The coreference heuristic will also fail to resolve any pronouns, so we discard them.

Another issue is that NPs whose referents are familiar tend to resemble discourse-old NPs, even though they have not been previously mentioned (Fraurud, 1990). These include unique objects like *the FBI* or

[4] Poesio et al. (2005) score 90.2%, but on a different corpus.

	Disc. Acc	Disc. F	Ins.
Random	50.00	50.00	12.58
Entity Grid‡	76.17	77.55	19.57
Disc-New‡	70.35	73.47	16.27
EGrid+Disc-New‡	78.88	80.31	21.93

Table 3.3: Performance of discourse-new model on 1004 WSJ test documents. ‡ for individual models indicates significant improvement over chance on both tasks with p=.001; for the combined model, it indicates improvement over the entity grid.

generic ones like *danger* or *percent.* To avoid using these deceptive phrases as examples of discourse-newness, we attempt to heuristically remove them from the training set by discarding any NP whose head occurs only once in the document[5].

The labels we apply to NPs in our test data are systematically biased by the "same head" heuristic we use for coreference. This is a disadvantage for our system, but it has a corresponding advantage– we can use training data labeled using the same heuristic, without any loss in performance on the coherence task. NPs we fail to learn about during training are likely to be mislabeled at test time anyway, so performance does not degrade by much. To counter this slight degradation, we can use a much larger training corpus, since we no longer require gold-standard coreference annotations.

3.2.1 Experiments

We evaluate our models using the discrimination and insertion tasks described in section 2.2, experimenting on the Wall Street Journal corpus. Because the WSJ articles are longer than those of AIRPLANE, ordering is too computationally difficult to be a useful task. However, insertion remains tractable and is more sensitive than discrimination.

Our results are shown in Table 3.3. When run alone, the entity grid outperforms the discourse-new model. However, the model is still substantially better than chance. Combining it with the entity grid raises discrimination performance by 2.7% over the baseline and insertion by 2.4%[6].

3.3 Pronoun coreference for coherence

Pronoun coreference is an important aspect of coherence– if a pronoun is used too far away from any natural referent, it becomes hard to interpret, creating confusion. Too many referents, however, create ambiguity. To measure text quality using a pronoun resolution model, we take as our coherence score the probability of the text containing pronouns (denoted r_i), jointly with their referents a_i. This joint probability is modeled by generative systems, but not by conditional ones, which can resolve the pronouns without giving a distribution over texts.

[5] Bean and Riloff (1999) and Uryupina (2003) construct quite accurate classifiers to detect unique NPs. However, some preliminary experiments convinced us that our heuristic method worked well enough for the purpose.

[6] The significance tests in this section are performed with Wilcoxon's signed-rank test.

	Disc. Acc	Disc. F	Ins.
Random	50.00	50.00	12.58
Entity Grid‡	76.17	77.55	19.57
Pronouns (Ge et al. (1998))‡	55.77	62.27	13.95
EGrid+Disc-New‡	78.88	80.31	21.93
+Pronouns‡	79.60	81.02	22.98

Table 3.4: Performance of coreference-inspired models on 1004 WSJ test documents. ‡ for individual models indicates significant improvement over chance with p=.001; the combined model is significantly better than the entity grid, and the combined model with pronouns is significantly better than without.

Given access to the distribution $P(a_i, r_i)$, we can compute the probability of a document by summing out the antecedents a. Unfortunately, our models typically condition each a_i on the previous ones (for instance, if a pronoun would be resolved to another pronoun, we use transitivity to search backward until we find a full NP antecedent). Therefore, this cannot be done efficiently. Instead, we use a greedy search, assigning each pronoun left to right. Finally we report the probability of the resulting sequence of pronoun assignments.

We start by doing experiments using the model of Ge et al. (1998).

$$\begin{aligned} P(a_i, r_i | a_i^{i-1}) =& P(a_i | h(a_i), m(a_i)) \\ & P_{gen}(a_i, r_i) P_{num}(a_i, r_i) \end{aligned}$$

Here $h(a)$ is the Hobbs distance (Hobbs, 1976), which measures distance between a pronoun and prospective antecedent, taking into account various factors, such as syntactic constraints on pronouns. $m(a)$ is the number of times the antecedent has been mentioned previously in the document (again using "same head" coreference for full NPs, but also counting the previous antecedents a_i^{i-1}). P_{gen} and P_{num} are distributions over gender and number given words. The model is trained using the small hand-annotated corpus first used in Ge et al. (1998).

As before, we evaluate our model using the discrimination and insertion tasks on the Wall Street Journal corpus. Our results are shown in Table 3.4. The pronoun model is weaker than the entity grid or discourse-new models, but still better than random. Combining all three models raises discrimination performance by 3.5% over the baseline and insertion by 3.4%. Pronouns contribute significantly to the joint model; the *EGrid + Disc-New* model is significantly worse than the full combination[7].

Next we apply an unsupervised model given by Charniak and Elsner (2009). This model disambiguates more pronouns, has a wider feature space and uses more training data; thus, we expect it to be an improvement on the Ge et al. (1998) model.

Results for the model of Charniak and Elsner (2009) are shown in Table 3.5. These results show that the Charniak and Elsner (2009) model is more powerful than Ge et al. (1998), due to its ability to disambiguate more pronouns, but that the effect shrinks noticeably when the entity grid is added; although the model disambiguates more pronouns, it is still not as strong a source of information as the entity grid, and its proportional contribution to the mixture is small; in this case, the models are no longer significantly different.

[7] The significance tests in this section are performed with Wilcoxon's signed-rank test.

	Disc. Acc	Disc. F	Ins.
Random	50.0	50.0	12.6
Entity Grid	76.2	77.6	19.6
Pronouns (Ge et al. (1998))	55.8	62.3	14.0
Pronouns (Charniak and Elsner, 2009)	**64.6**‡	**69.6**	16.6†
Entity Grid + Pronouns (Ge et al., 1998)	77.3	78.7	20.8
Entity Grid + Pronouns (Charniak and Elsner, 2009)	78.3	79.7	22.1

Table 3.5: Performance of pronoun models on 1004 WSJ test documents. ‡ indicates a significant difference between Charniak and Elsner (2009) and Ge et al. (1998) at p=.001.

The pronoun model is not particularly powerful, but it does lead to significant improvements over the previous models, since it captures a source of information they cannot use. The model's main problem is that, although a pronoun may have been displaced from its original position, it can often find another seemingly acceptable referent nearby.

As mentioned, Barzilay and Lapata (2005) uses a coreference system to attempt to improve the entity grid, but with mixed results. Their method of combination is quite different from ours; they use the system's judgements to define the "entities" whose repetitions the system measures[8]. In contrast, we do not attempt to use any proposed coreference links; as Barzilay and Lapata (2005) point out, these links are often erroneous because the disorded input text is so dissimilar to the training data. Instead we exploit our models' ability to measure the probability of various aspects of the text.

3.4 Extending the entity grid

Referring expressions have more to tell us than simply which entity they refer to. In the previous section, we looked at the relationship between different mentions of the same entity. In this section, we look at what we can learn about the entity itself by examining the expressions that refer to it. Because doing so will require a slightly more flexible framework than we presented in chapter 2, we first discuss a few minor extensions to the basic entity grid. In the rest of the section, we create two models, one which distinguishes a variety of different entity types, and another which models the relationships between different entities using a topic-based representation.

As our first improvement, we loosen our mention detection, adding non-head nouns to the entity grid and assigning them the role **X**. Like many of the improvements in this chapter, this is motivated by work in coreference resolution. In order to achieve the highest levels of recall, coreference resolvers must also detect mentions that do not correspond to NPs. For instance, Haghighi and Klein (2010) associate a mention with every noun, regardless of whether it heads an NP, and Elsner and Charniak (2010b) find that recall of same-head coreference decisions is improved when considering non-head nouns as mentions. This is necessary to pick up premodifiers in phrases like "a **Bush** spokesman", since such modifiers can refer to entities. (High-precision models of the internal structure of flat Penn Treebank-style NPs were investigated by Vadas and Curran (2007).) Therefore, we add non-head nouns to our entity grid. The results of this change are shown

[8] We attempted this method for pronouns using our model, but found it ineffective.

	Disc. Acc	Disc. F	Ins.
Random	50.0	50.0	12.6
Entity Grid: NPs‡	74.4	76.2	21.3
Entity Grid: all nouns†	**77.8**	**79.7**	**23.5**

Table 3.6: Discrimination scores for entity grids with different mention detectors on WSJ development documents. ‡ indicates performance on both tasks is significantly different from the previous row of the table with p=.001.

	Discrimination F
Multinomial EGrid	79.6
Linear (independent) EGrid	78.2
Linear (cross-product) EGrid	79.7

Table 3.7: Discrimination scores for multinomial and linear entity grids on WSJ development data. The differences indicated here are not statistically significant at the .05 level.

in Table 3.6; discrimination performance increases about 4%, from 76% to 80%. In the remainder of the dissertation, we use this better-performing mention detection system.

Next, we modify our parameterization of the model so that it can account for more conditioning information. We retain the standard parameterization of the generative grid model in terms of $P(r_{i,j}|\vec{r}^{h}_{(i-1),j})$, the conditional probability of an entity taking a particular syntactic role in the next sentence. The usual (non-parametric) estimator for this distribution learns an independent multinomial for each conditioning vector, which means the number of parameters grows exponentially as we increase the amount of conditioning information, and none of the context elements can be real-valued. Since we want to add additional information, we make a parametric assumption about the form of the conditional, adopting logistic regression. Multilabel logistic regression is parameterized by a vector of coefficients w_y for each possible syntactic role y:

$$P(r_i = y|r_{i-1}...r_{i-h}) = \frac{exp(\vec{w_y} \cdot \vec{r}^{h}_{i-1})}{\sum_{j \in Y} exp(\vec{w_j} \cdot \vec{r}^{h}_{i-1})} \tag{3.1}$$

This models the contribution of each r (plus the salience, and any additional features) as independent and linear.

While this model is useful, dependences between the features tend to disrupt its performance. If we suspect two features f, g have a non-additive effect, we can add a "combination" feature which takes values in the cross-product space: $h \in (f \times g)$. As a simple example, we can learn a parameter for each of the 64 possible contexts, just like the maximum-likelihood entity grid. As one would expect, the two models have very similar performance at 80% (Table 3.7), while the linear model does a bit worse at 78%; this drop in performance is not significant.

The logistic regressor can be trained using any gradient method– we use OWLQN (Orthant-wise limited-memory quasi-Newton) (Andrew and Gao, 2007)[9]. For problems with lots of features and training data, this can be slow and uses a lot of space; we can speed up training by parallelizing. To do so, we split the data, estimate separately for each fold, and average the results (Mann et al., 2009).

[9] The implementation we use incorporates some modifications by Mark Johnson.

3.4.1 Entity-specific features

In this section, we add conditioning features that separate important entities from less important ones– or, in fact, from spurious entities erroneously created by our same-head coreference heuristic. The entity grid already has one feature that is supposed to do this, *salience*, which as discussed in chapter 2, is usually implemented by counting mentions. However, we can do better by implementing a larger collection of features which can make finer distinctions between entities. We list these features below:

Linkable This feature is active if the head word of the entity is marked as coreferent in MUC.

Unlinkable This feature is active if the head word of the entity occurs five times in MUC and is never marked as coreferent.

Has pronouns This feature is active if the head word of the entity has a pronominal coreferent 5 or more times in the NANC corpus. (Pronouns in NANC are automatically resolved using the unsupervised model of Charniak and Elsner (2009).

No pronouns This feature is active if the head word of the entity occurs 50 times in NANC and has a pronominal coreferent less than 5 times.

Proper This feature is active if the entity has a proper mention.

Named entity This feature is set to the majority named entity label for the coreferential chain, as assigned by Morton et al. (2005).

Modifiers This feature is set to the total number of modifiers used throughout the coreferential chain, divided by 5.

Plural This feature is active if the head word of the entity is plural.

Several of the features are designed to find head words that tend to be part of spurious entities (Elsner and Charniak, 2010b), or to pick out mentions of real, but unimportant entities. Head words which are *unlinkable* and have *no pronouns* are unlikely to have same-head coreferents– these include words like “miles” or “percent”. Words that are *linkable* or *have pronouns*, by contrast, are more likely to belong to “real” entities. Elsner and Charniak (2010b) also points out that *plurality* is an indicator for non-coreferent classes like generic NPs and quantity phrases. Adding features to distinguish these classes of NP frees the model to expect different patterns for them than for real entities.

Proper and *named entity* features, on the other hand, point out classes of entities likely to be not only real but also important (Nenkova, 2006). We particularly expect this to be true in the case of people and organizations, which are likely participants in the events described by a news article. Entities with many *modifiers* throughout the document are similarly likely to be more important, since this implies that the writer wishes to point out more information about them.

Our starting point for the extended entity grid is the linear cross-product entity grid from the previous section, which has a parameter for each of the 64 contexts h and has identical performance to the multinomial

	Disc. Acc	Disc. F	Ins.
Random	50.00	50.00	12.6
Entity Grid	79.5	80.9	21.4
Extended Entity Grid	**84.0**†	**84.5**	**24.2**
EGrid/Pronouns/Disc/IBM	82.6	84.0	24.3
ExEGrid/Pronouns/Disc/IBM	**86.0**†	**86.5**	**26.7**†

Table 3.8: Extended entity grid and combination model performance on 1004 WSJ test documents. †indicates an extended model score better than its baseline counterpart at p=.05.

entity grid. We take each of the features f just mentioned, and create a parameter for $(f \times h)$– that is, we model the way our belief about what will happen in context h changes non-additively when we know that the entity involved has feature f.

Table 3.8 gives some results for entity-specific features on the test set. The extended grid model is quite a bit better than the regular grid model (84% versus 81%). When we incorporate it into a combination model with pronouns, the discourse-new model and IBM-1, the performance increase remains (86% versus 84%). Though the improvement is not additive, a good deal of it is retained, demonstrating that these additions to the entity grid are mostly orthogonal to previously described models.

3.4.2 Topical entity grids

This section presents a variant of the generative entity grid which uses topical information to model similarities between entities. In the standard entity grid, the entities are independent, which means that the model penalizes coherent transitions between different entities, like: “The House voted yesterday. The Senate will consider the bill today.” To remedy this, we wish to add some kind of lexical information to the model, but we would like to do so while retaining a small parameter space, rather than building a fully lexicalized model such as IBM. This will enable us to deal effectively with rare words and to continue to use a multi-sentence history instead of just the previous sentence. While such a model was constructed for German by Filippova and Strube (2007), that model required the user to set a hard threshold for the degree of similarity, then simply merged all entities more similar than the threshold into a single column of the grid. In contrast, we work directly with a continuous representation of word similarity.

Instead of learning lexical parameters directly, as in a lexical model like IBM, the topical entity grid uses a vector representation of each word in the vocabulary and a fixed measurement of lexical similarity. The representations for words are derived from LDA (Blei et al., 2001)[10] with 200 latent topics. We represent each word as its vector of conditional probabilities for each latent topic: $p(t_i|w)$. We experimented with several ways to measure relationships between words in this space, starting with the standard cosine. However, the cosine can depend on quite small variations in probability (for instance, if w has most of its mass in dimension 1, then the cosine is greatly affected by the exact weight of v for topic 1, even if this essentially never happens).

To control for this tendency, we instead use the magnitude of the dimension of greatest similarity:

[10] www.cs.princeton.edu/~blei/topicmodeling.html

	Disc. Acc	Disc. F	Ins.
Entity grid	$79.5^{\ddagger}$	80.9	21.4
Topical entity grid	74.5	75.9	20.9

Table 3.9: Results of topical entity grid on WSJ test documents. ‡ indicates a better score at p=.001.

$$sim(w, v) = max_i \; min(w_i, v_i)$$

To model coherence, we generalize the binary history features of the standard entity grid, which detect, for example, whether entity e is the subject of the previous sentence. In the topical entity grid, we instead compute a real-valued feature which sums up the similarity between entity e and the subject(s) of the previous sentence:

$$subj_{e,prev} = \sum_{k:\text{subjects of prev. sentence}} sim(e, k)$$

Thus, in the "House" and "Senate" example, $sim(house, senate)$ should be high (assuming both words have high values for a "government" topic), and this increases the value of $subj_{house,-1}$, which should in turn help to predict that "House" will be a good subject for the current sentence.

The results (Table 3.9) show that the topical entity grid is not useful for ordering in the news domain. To anticipate chapter 5, however, we find its accuracy is much greater for disentanglement of phone conversations. A potential hint as to why comes from the insertion results, which are comparable to those for the standard entity grid, despite a drop of rougly 5% for discrimination. Since insertion looks at small changes between documents rather than completely random permutations, this model, which is designed to be sensitive to less obvious relationships between words, is better-equipped to differentiate. However, this sensitivity comes at the expense of increased noise, so when more good cues are available, the model does not do as well. Disentanglement in the conversational domain is an example of a case where good cues are rare, since, like insertion, it makes small changes to documents, and the style is less clear to begin with. We believe this is why the model's performance is so much improved on the new task.

3.5 Conclusion

In this chapter, we have shown several entity-based models for document coherence, and improved the state of the art for newswire sentence ordering. Our performance here raises an important question: how specific are these models to this particular task setting? In the next chapter, we begin our discussion of chat disentanglement, a new evaluation task for coherence models. Once we have defined reasonable standards for performance for this task, we use it as an evaluation for the models presented here, in chapter 5.

Chapter 4

Disentangling chat

While the previous chapter demonstrated that we can achieve state-of-the-art performance for ordering tasks on news data, we are left to wonder about the generality of our results. We wish to perform experiments in a different domain and task setting. Chat disentanglement fits both criteria, as well as having some practical applications. Disentanglement is the clustering task of dividing a transcript into a set of distinct conversations. Our major contributions in this chapter are *a publically available disentanglement corpus with interannotator agreement* (section 4.2) and *a strong baseline model for disentanglement* (section 4.3).

Disentanglement is an essential prerequisite for any kind of higher-level dialogue analysis: for instance, consider the multi-party exchange in figure 4.1.

Contextually, it is clear that this corresponds to two conversations, and Felicia's[1] response *excellent* is intended for Chanel and Regine. A straightforward reading of the transcript, however, might interpret it as a response to Gale's statement immediately preceding.

Chanel	Felicia: google works :)
Gale	Arlie: you guys have never worked in a factory before have you
Gale	Arlie: there's some real unethical stuff that goes on
Regine	hands Chanel a trophy
Arlie	Gale, of course ... thats how they make money
Gale	and people lose limbs or get killed
Felicia	excellent

Figure 4.1: Some (abridged) conversation from our corpus.

The connection between disentanglement and coherence is straightforward. A normal conversation between humans is a discourse, and should obey the same principles of coherence as text. A badly disentangled transcript, however, is not a discourse, and just like a permuted document, it should contain structural violations that lead us to disprefer it. Thus we should be able to distinguish good from bad disentanglement by maximizing the coherence of each individual conversational thread according to our models.

Before we apply this approach, however, we must design a meaningful experimental protocol and make

[1] Real user nicknames are replaced with randomly selected identifiers for ethical reasons.

it clear what baseline scores ought to look like. As explained in the introduction (section 1.3), previous work did not agree on metrics or standard corpora. In this chapter, we present our attempt to standardize work in this area. Section 4.1 explains some theoretical background. Section 4.2 describes our corpus of manually annotated chat room data and evaluates annotator reliability. We give a set of metrics (section 4.2.2) describing structural similarity both locally and globally. In section 4.3, we propose a model which uses supervised pairwise classification to link utterances from the same conversation, followed by a greedy inference stage which clusters the utterances into conversations. Our system uses unigram overlap features, essentially the simplest possible measurement of local coherence. Experimental results (section 4.4) show that its output is highly correlated with human annotations. In sections 4.5 and 4.6, we investigate two extensions to the basic model, specificity tuning and automatic detection of conversation starts. Finally, we discuss further work on partitioning (section 4.7), in which we improve our greedy search procedure with local search and bound its performance using semi-definite programming.

4.1 Background

While the notion of a “conversation” or “thread” is a fairly intuitive one, there are edge cases for which it helps to have a precise definition. In this study, we take a conversation to be an independent *floor* (Simpson, 2005); only one speaker at a time may have the floor, and other participants who wish to speak must acquire the floor for themselves. Simultaneous conversations, therefore, can potentially share both topics and participants, as long as there are two floors with their own independent patterns of turn-taking.

Simultaneous conversations seem to arise naturally in both informal social interactions and multi-party typed chat. Aoki et al. (2006)’s study of voice conversations among 8–10 people found an average of 1.76 conversations (floors) active at a time, and a maximum of four. In our chat corpus, the average is even higher, at 2.75. The typical conversation, therefore, does not form a contiguous segment of the chatroom transcript, but is frequently broken up by interposed utterances from other conversations.

In a popular chatroom, entirely new conversations are constantly forming (for instance, when a newcomer asks a question). However, some new conversations branch off from existing ones in a process which Sacks et al. (1974) calls *schisming*. During a schism, a new conversation is formed, not necessarily because of a shift in the topic, but because certain participants have refocused their attention onto each other, and away from whoever held the floor in the parent conversation. Aoki et al. (2006) discuss conversational features associated with schisming and the related process of *affiliation*, by which speakers attach themselves to a conversation. Schisms often branch off from asides or even normal comments (*toss-outs*) within an existing conversation. This means that there is no clear beginning to the new conversation — at the time when it begins, it is not clear that there are two separate floors, and this will not become clear until distinct sets of speakers and patterns of turn-taking are established. Speakers, meanwhile, take time to orient themselves to the new conversation. Example schisms are shown in Figure 4.2 and 4.3. This issue will prove problematic for our annotation scheme, an issue we discuss in section 4.2.2.

A significant question is whether the existence of other conversational floors, and the possibility that a schism might occur at any time, affects the devices used to create coherence within a single conversation. In

Santo	Madison: I grew up in Romania until I was 10.
Santo	And my parents are fucking crazy. Totally fucked with my head... couldn’t stand life so I dropped out of school even though I had a full ride. :(
Kandra	⇒ Santo: you’re at OSU? ⇐
Madison	Santo: you still speak romanian?
Santo	yes

Figure 4.2: A schism occurring in our corpus (abridged). The schism-inducing turn is Kandra’s comment, marked by arrows. Annotators 0 and 2 begin a new conversation with this turn; 1, 4 and 5 group it with the other utterances shown; 3 creates new conversations for both this turn and Madison’s question immediately following.

Lai	need money
Astrid	suggest a paypal fund or similar
Lai	→ Azzie [sic; typo for Astrid?]: my shack guy here said paypal too but i have no local bank acct ←
Gale	⇒ we should charge the noobs $1 per question to [Lai’s] paypal ⇐
Gale	we’d have the money in 2 days max
Azzie	⇒ Lai: hrm, Have you tried to set one up? ⇐
Felicia	Gale: all ubuntu users .. pay up!
Gale	and susers pay double
Lai	Azzie: not since being offline
Felicia	it doesn’t need to be “in state” either

Figure 4.3: A schism occurring in our corpus (abridged): not all annotators agree on where the thread about charging for answers to techical questions diverges from the one about setting up Paypal accounts. The schism begins just after Lai’s second comment (marked with arrows), to which Gale and Azzie both respond (marked with double arrows). Annotators 1, 2, 4 and 5 begin a new conversation with Gale’s response. Annotator 0 starts a new conversation with Azzie’s response. Annotator 3 makes an error, linking the two responses to each other, but not to the parent.

other words, are conversations in a multiparty environment really independent of one another, as our coherence framework for disentanglement assumes? We know of only one major difference, pointed out by O’Neill and Martin (2003). This is the frequency with which participants mention each others’ names. O’Neill and Martin (2003) hypothesize that name mentioning is a strategy which participants use to make disentanglement easier, compensating for the lack of cues normally present in face-to-face dialogue. Mentions (such as Gale’s comments to Arlie in figure 4.1) are very common in our corpus, occurring in 36% of comments, and provide a useful feature.

4.2 Dataset

Our dataset is recorded from the IRC (Internet Relay Chat) channel #LINUX at*free–node.net*, using the freely-available *gaim* client. #LINUX is an unofficial tech support line for the Linux operating system, selected because it is one of the most active chat rooms on freenode, leading to many simultaneous conversations, and because its content is typically inoffensive. Although it is notionally intended only for tech support, it includes large amounts of social chat as well, such as the conversation about factory work in the example above (figure 4.1).

The entire dataset contains 52:18 hours of chat, but we devote most of our attention to three annotated sections: development (706 utterances; 2:06 hr) and test (800 utts.; 1:39 hr) plus a short pilot section on which we tested our annotation system (359 utts.; 0:58 hr).

4.2.1 Annotation

We recruited and paid seven university students to annotate the test section. All had at least some familiarity with the Linux OS, although in some cases very slight. Annotation of the test dataset typically took them about two hours. In all, we produced six annotations of the test set.[2]

We have four annotations of the pilot set, by three volunteers and the experimenters. The pilot set was used to prototype our annotation software, and also as a validation corpus for our system. The development set was annotated only once, by the experimenter. This dataset is used for training.

Our annotation scheme marks each utterance as part of a single conversation. Annotators are instructed to create as many, or as few conversations as they need to describe the data. Our instructions state that a conversation can be between any number of people, and that, "We mean conversation in the typical sense: a discussion in which the participants are all reacting and paying attention to one another… it should be clear that the comments inside a conversation fit together." The annotation system itself is a simple Java program with a graphical interface, intended to appear somewhat similar to a typical chat client. Each speaker's name is displayed in a different color, and the system displays the elapsed time between comments, marking especially long pauses in red. Annotators group utterances into conversations by clicking and dragging them onto each other.

4.2.2 Metrics

Before discussing the annotations themselves, we will describe the metrics we use to compare different annotations; these measure both how much our annotators agree with each other, and how well our model and various baselines perform. Comparing clusterings with different numbers of clusters is a non-trivial task, and metrics for agreement on supervised classification, such as the κ statistic, are not applicable.

To measure global similarity between annotations, we use *one-to-one accuracy*. This measure describes how well we can extract whole conversations intact, as required for summarization or information extraction. To compute it, we pair up conversations from the two annotations to maximize the total overlap by computing an optimal max-weight bipartite matching, then report the percentage of overlap found.[3] One-to-one accuracy is a standard metric in unsupervised part-of-speech tagging (for instance Haghighi and Klein (2006)), and is equivalent to *mention-based CEAF* (Luo, 2005) for coreference resolution.

If we intend to monitor or participate in the conversation as it occurs, we will care more about local judgements. The *local agreement* metric is a constrained form of the Rand index for clusterings (Rand, 1971)

[2] One additional annotation was discarded because the annotator misunderstood the task.

[3] The matching can be computed efficiently with the so-called Hungarian algorithm or by reduction to max flow. The widely used greedy algorithm is a two-approximation, although we have not found large differences in practice.

which counts agreements and disagreements for pairs within a context k. We consider a particular utterance: the previous k utterances are each in either the *same* or a *different* conversation. The loc_k score between two annotators is their average agreement on these k same/different judgements, averaged over all utterances. For example, loc_1 counts pairs of adjacent utterances for which two annotations agree.

Several related papers use some variant of the *F-score* metric to measure accuracy. The most complete treatment is given in Shen et al. (2006). They use a micro-averaged F-score, which is defined by constructing a multiway matching between conversations in the two annotations. For a gold conversation i with size n_i, and a proposed conversation j with size n_j, with overlap of size n_{ij}, they define precision and recall (plus the standard balanced F-score). The F-score of an entire annotation is a weighted sum over the matching:

$$P = \frac{n_{ij}}{n_j} \qquad R = \frac{n_{ij}}{n_i} \qquad F(i,j) = \frac{2PR}{P+R} \qquad F = \sum_i \frac{n_i}{n} max_j F(i,j) \tag{4.1}$$

This is the F-score we report for comparative purposes. Since the match is multiway, the score is not symmetric; when measuring agreement between pairs of human annotators (where there is no reason for one to be considered gold), we map the high-entropy transcript to the lower one (the entropy of a transcript is defined below, in equation 4.2). Micro-averaged f-scores are also popular in work on document clustering. In general, scores using this metric are correlated with our other measurement of global consistency, the one-to-one accuracy.

Adams and Martell (2008) also report F-score, but using a somewhat different definition. They define F-score only between a particular pair of conversations, and report the score for a single selected conversation. They do not describe how this reference conversation is chosen. It is also unclear how they determine which proposed conversation to match to it — the one with the best F-score, or the one which contains the first (or “root”) utterance of the reference conversation. (The latter, although it may be more useful for some applications, has an obvious problem — if the conversation is retrieved perfectly *except* for the root utterance, the score will be zero.) For these reasons we do not evaluate their metric.

4.2.3 Discussion

A statistical examination of our data (table 4.1) shows that there is a substantial amount of disentanglement to do: the average number of conversations active at a time (the *density*) is 2.75. Our annotators have high agreement on the local metric (average of 81.1%). On the 1-to-1 metric, they disagree more, with a mean overlap of 53.0% and a maximum of only 63.5%. Though this level of agreement is low, naive baselines score even lower (see section 4.4). Therefore the metric does indeed distinguish human-like from baseline performance. Thus measuring 1-to-1 overlap with our annotations is a reasonable evaluation for computational models. However, we feel that the major source of disagreement is one that can be remedied in future annotation schemes: the specificity of the individual annotations.

To measure the level of detail in an annotation, we use the information-theoretic *entropy* of the random variable which indicates which conversation an utterance is in. This variable has as many potential values as the number of conversations in the transcript, each value having probability proportional to its size. Thus, for

	Mean	Max	Min
Conversations	81.33	128	50
Average Conversation Length	10.6	16.0	6.2
Average Conversation Density	2.75	2.92	2.53
Entropy	4.83	6.18	3.00
1-to-1	52.98	63.50	35.63
loc_3	81.09	86.53	74.75
Many-to-1 (by entropy)	86.70	94.13	75.50
Shen F (by entropy)	53.87	66.08	35.43

Table 4.1: Statistics on 6 annotations of 800 utterances of chat transcript. Inter-annotator agreement metrics (below the line) are calculated between distinct pairs of annotations.

a transcript of length n, with conversations i each having size n_i, the entropy is:

$$H(c) = \sum_i \frac{n_i}{n} log_2 \frac{n_i}{n} \tag{4.2}$$

This quantity is non-negative, increasing as the number of conversations grow and their size becomes more balanced. It reaches its maximum, 9.64 bits for this dataset, when each utterance is placed in a separate conversation. In our annotations, it ranges from 3.0 to 6.2. This large variation shows that some annotators are more specific than others, but does not indicate how much they agree on the general structure. To measure this, we introduce the *many-to-one accuracy*. This measurement is asymmetrical, and maps each of the conversations of the *source* annotation to the single conversation in the *target* with which it has the greatest overlap, then counts the total percentage of overlap. This is not a statistic to be optimized (indeed, optimization is trivial: simply make each utterance in the source into its own conversation), but it can give us some intuition about specificity. In particular, if one subdivides a coarse-grained annotation to make a more specific variant, the many-to-one accuracy from fine to coarse remains 1. When we map high-entropy annotations (fine) to lower ones (coarse), we find high many-to-one accuracy, with a mean of 86%, which implies that the more specific annotations have mostly the same large-scale boundaries as the coarser ones.

By examining the local metric, we can see even more: local correlations are good, at an average of 81.1%. This means that, in the three-sentence window preceding each sentence, the annotators are often in agreement. If they recognize subdivisions of a large conversation, these subdivisions tend to be contiguous, not mingled together, which is why they have little impact on the local measure.

We find that annotators' disagreement about appropriate levels of detail occurs mostly around schisms, where new conversations break off from old ones. Our annotation scheme requires annotators to mark each utterance as part of a single conversation, and distinct conversations are not related in any way. If a schism occurs, the annotator is faced with two options: if it seems short, they may view it as a mere digression and label it as part of the parent conversation. If it seems to deserve a place of its own, they will have to separate it from the parent, but this severs the initial comment (an otherwise unremarkable aside) from its context. One or two of the annotators actually remarked that this made the task confusing. Our annotators seem to be either "splitters" or "lumpers" — in other words, each annotator seems to aim for a consistent level of detail, but each one has their own idea of what this level should be.

As a final observation about the dataset, we test the appropriateness of the assumption (used in previous

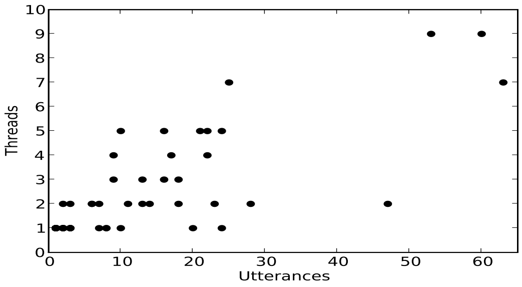

Figure 4.4: Utterances versus conversations participated in per speaker on development data.

work) that each speaker takes part in only one conversation. In our data, the average speaker takes part in about 3.3 conversations (the actual number varies for each annotator). The more talkative a speaker is, the more conversations they participate in, as shown by a plot of conversations versus utterances (Figure 4.4). The assumption is not very accurate, especially for speakers with more than 10 utterances.

4.3 Model

Our model for disentanglement fits into the general class of graph partitioning algorithms (Roth and Yih, 2004) which have been used for a variety of tasks in NLP, including coreference resolution (Soon et al., 2001) and the related task of meeting segmentation (Malioutov and Barzilay, 2006). These algorithms operate in two stages: first, a binary classifier marks each pair of items as alike or different, and second, a consistent partition is extracted. The partitioning problem was identified as *correlation clustering*, an NP-hard problem, by McCallum and Wellner (2004). A more extensive theoretical review of the problem is given in section 4.7.

4.3.1 Classification

We use a maximum-entropy classifier (Daumé III, 2004) to decide whether a pair of utterances x and y are in *same* or *different* conversations. The most likely class is *different*, which occurs 57% of the time in development data. We describe the classifier's performance in terms of raw accuracy (correct decisions / total), precision and recall of the *same* class, and F-score, the harmonic mean of precision and recall. Our classifier uses several types of features (table 4.2). The chat-specific features yield the highest accuracy and precision. Discourse and content-based features have poor accuracy on their own (worse than the baseline), since they work best on nearby pairs of utterances, and tend to fail on more distant pairs. Paired with the time gap feature, however, they boost accuracy somewhat and produce substantial gains in recall, encouraging the model to group related utterances together.

The classifier is trained on our single annotation of the 706-utterance development section and validated against the 359-utterance pilot section.

The time gap, as discussed above, is the most widely used feature in previous work. Our choice of a logarithmic binning scheme is intended to capture two characteristics of the distribution of pause lengths

	Chat-specific (Acc 73: Prec: 73 Rec: 61 F: 66)
Time	The time between *x* and *y* in seconds, discretized into logarithmically sized bins.
Speaker	*x* and *y* have the same speaker.
Mention x-y	*x* mentions the speaker of *y* (or vice versa). For example, this feature is true for a pair such as: *Felicia "Gale: ...* and any utterance spoken by Gale.
Mention same	Both *x* and *y* mention the same name.
Mention other	either *x* or *y* mentions a third person's name.
	Discourse (Acc 52: Prec: 47 Rec: 77 F: 58)
Cue words	Either *x* or *y* uses a greeting (*hello* etc.), an answer (*yes*, *no* etc.), or thanks.
Question	Either asks a question (explicitly marked with *?*).
Long	Either is long (> 10 words).
	Content (Acc 50: Prec: 45 Rec: 74 F: 56)
Repeat(i)	The number of words shared between *x* and *y* which have unigram probability i, binned logarithmically.
Tech	Whether both *x* and *y* use technical jargon, neither do, or only one does.
	Combined (Acc 75: Prec: 73 Rec: 68 F: 71)

Table 4.2: Feature functions with performance on development data.

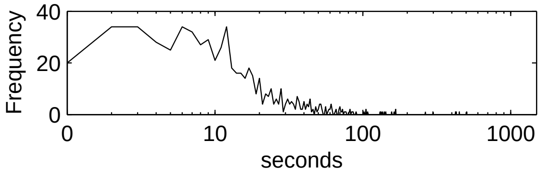

Figure 4.5: Distribution of pause length (log-scaled) between utterances in the same conversation.

(shown in figure 4.5). The curve has its maximum at 1–3 seconds, and pauses shorter than a second are less common. This reflects turn-taking behavior among participants; participants in the same conversation prefer to wait for each others' responses before speaking again. On the other hand, the curve is quite heavy-tailed to the right, leading us to bucket long pauses fairly coarsely. The specific discretization we adopt for a time gap Δ is $bin(\Delta) = floor(log_{1.5}(\Delta + 1))$. The particular choice of 1.5 was chosen by hand to fit the observed scale of the curve.

Our discourse-based features model some pairwise relationships: questions followed by answers, short comments reacting to longer ones, greetings at the beginning and thanks at the end.

Word repetition is a key feature in nearly every model for segmentation or coherence, so it is no surprise that it is useful here. We discard the 50 most frequent words. Then we bin all words by their unigram probability ($bin(w) = floor(log_{10}(p(w)))$) and create an integer-valued feature for each bin, equal to the number of repeated words in that bin. Unigram probabilities are calculated over the entire 52 hours of transcript. The binning scheme allows us to deal with "noise words" which are repeated coincidentally, since these occur in high-probability bins where repetitions are given less weight.

```
$k \leftarrow 0$ // number of clusters created so far
for $i = 1 \dots n$ do
  for $c = 1 \dots k$ do
    $Quality_c \leftarrow \sum_{j \in C[c]} w_{ij}$
  $c^* \leftarrow \arg\max_{1 \leq c \leq k} Quality_c$
  if $Quality_{c^*} > 0$ then
    $C[c^*] \leftarrow C[c^*] \cup \{i\}$
  else
    $C[k{+}{+}] \leftarrow \{i\}$ // form a new cluster
```

Figure 4.6: VOTE algorithm

The point of the repetition feature is of course to detect sentences with similar topics. We also find that sentences with technical content are more likely to be related than non-technical sentences. We label an utterance as technical if it contains a web address, a long string of digits, or a term present in a guide for novice Linux users [4] but not in a large news corpus (Graff, 1995).[5] This is a light-weight way to capture one "semantic dimension" or cluster of related words. The technical word feature was included because it improves our development classification score slightly, but it does not have a significant effect on overall performance. Adams (2008) attempts to add more semantic dimensions learned via Latent Dirichlet Allocation, and similarly finds no improvement.

Pairs of utterances which are widely separated in the discourse are unlikely to be directly related — even if they are part of the same conversation, the link between them is probably a long chain of intervening utterances. Thus, if we run our classifier on a pair of very distant utterances, we expect it to default to the majority class, which in this case will be *different*, and this will damage our performance in case the two are really part of the same conversation. To deal with this, we run our classifier only on utterances separated by 129 seconds or less. This is the last of our logarithmic buckets in which the classifier has a significant advantage over the majority baseline. For 99.9% of utterances in an ongoing conversation, the previous utterance in that conversation is within this gap, and so the system has a chance of correctly linking the two.

On test data, the classifier has a mean accuracy of 68.2 (averaged over annotations). The mean precision of *same conversation* is 53.3 and the recall is 71.3, with mean F-score of 60. This error rate is high, but the partitioning procedure allows us to recover from some of the errors, since if nearby utterances are grouped correctly, the bad decisions will be outvoted by good ones.

4.3.2 Partitioning

The next step in the process is to cluster the utterances. We wish to find a set of clusters for which the weighted accuracy of the classifier would be maximal; this is an example of *correlation clustering* (Bansal et al., 2004), which is NP-complete. The input to our partitioning procedure is a graph with a node for each utterance; if the classifier connects utterances i and j with probability p, we take the weight w_{ij} of edge

[4] "Introduction to Linux: A Hands-on Guide". Machtelt Garrels. Edition 1.25 from http://tldp.org/LDP/intro-linux/html/intro-linux.html .

[5] Our data came from the LA times, 94–97 — helpfully, it predates the current wide coverage of Linux in the mainstream press.

	Annotators	Model	Time/ment.	Perf. Seg.	Best Baseline
Mean 1-to-1	52.98	41.23	38.62	26.20	35.08 (Pause 35)
Max 1-to-1	63.50	52.12	44.12	36.50	56.00 (Pause 65)
Min 1-to-1	35.63	31.62	30.62	15.38	27.50 (Blocks 80)
Mean loc_3	81.09	72.94	68.69	75.98	62.16 (Speaker)
Max loc_3	86.53	74.70	70.93	85.40	69.05 (Speaker)
Min loc_3	74.75	70.77	66.37	69.05	54.37 (Speaker)
Mean Shen F	53.87	43.47	41.31	35.50	36.58 (Speaker)
Max Shen F	66.08	57.57	48.85	46.70	46.79 (Speaker)
Min Shen F	35.43	32.97	32.07	21.83	29.09 (Blocks 65)

Table 4.3: Metric values between proposed annotations and human annotations. Model scores typically fall between inter-annotator agreement and baseline performance.

ij to be the log odds $log(p_{ij}/(1-p_{ij}))$.[6] We create a variable x_{ij} for each pair of utterances, which is 1 if the utterances are placed in the same conversation, and 0 if they are separated. The log probability of the clustering, treating the edges as independent, is $\sum_{ij:i<j} w_{ij}x_{ij}$. We attempt to maximize this quantity, subject to the constraint that the x_{ij} must form a legitimate clustering such that $x_{ij} = x_{jk} = 1$ implies $x_{ij} = x_{ik}$.

Finding an exact solution proves to be difficult; the problem has a quadratic number of variables (one for each pair of utterances) and a cubic number of triangle inequality constraints (three for each triplet).[7] With 800 utterances in our test set, even solving the linear relaxation of the problem with CPLEX (Ilog, Inc., 2003) is too expensive to be practical.

A relatively good solution can be obtained using a greedy voting algorithm (figure 4.6). In this algorithm, we assign utterance j by examining all previously assigned utterances i, and treating the classifier's judgement w_{ij} as a vote for $cluster(i)$. If the maximum vote is greater than 0, we set $cluster(j) = argmax_c\ vote_c$. Otherwise j is put in a new cluster.

If the utterances are considered in order, this is a natural online algorithm — it assigns each utterance as it arrives, without reference to the future. In section 4.7, we explore more complex offline randomized and local search methods and attempt to compute an upper bound on performance. We show there that more complicated search procedures allow improvements of 5-6% on the one-to-one and F-score metrics. The loc_3 metric is insensitive to these more complex search procedures.

4.4 Experiments

We annotate the 800 line test transcript using our system. The annotation obtained has 62 conversations, with mean length 12.90. The average density of conversations is 2.86, and the entropy is 3.72. This places it within the bounds of our human annotations (see table 4.1), toward the more general end of the spectrum.

[6] The original version of our system used a different weighting scheme, $w_{ij} = p_{ij} - .5$. The log-odds ratio behaves similarly for our basic algorithm, but appears to be more robust to other partitioning algorithms or tuning (see 4.5), so, for simplicity, we present it here as well.

[7] There is a triangle inequality constraint for each triplet i, j, k: $(1 - x_{ik}) \leq (1 - x_{ij}) + (1 - x_{jk})$.

	All Diff	All Same	Speaker	Blocks	Pause
Mean 1-to-1	10.16	20.93	31.31	(Blocks of 40) 33.73	(Pause of 35) 35.08
Max 1-to-1	16.00	53.50	38.25	(Blocks of 180) 44.00	(Pause of 65) 56.00
Min 1-to-1	6.25	7.13	24.12	(Blocks of 80) 27.5	(Pause of 25) 26.62
Mean loc_3	52.93	47.07	62.16	(Blocks of 10) 49.57	(Pause of 15) 49.15
Max loc_3	62.15	57.47	69.05	(Blocks of 145) 57.89	(Pause of 40) 57.76
Min loc_3	42.53	37.85	54.37	(Blocks of 10) 42.87	(Pause of 10) 45.55
Mean F	15.00	15.09	36.58	(Blocks of 40) 34.80	(Pause of 20) 36.15
Max F	23.40	41.42	46.79	(Blocks of 75) 42.59	(Pause of 65) 45.00
Min F	9.11	4.03	28.68	(Blocks of 65) 29.09	(Pause of 25) 28.88

Table 4.4: Metric values for all baselines.

As a standard of comparison for our system, we provide results for several baselines — trivial systems which any useful annotation should outperform.

All different Each utterance is a separate conversation.

All same The whole transcript is a single conversation.

Blocks of k Each consecutive group of k utterances is a conversation.

Pause of k Each pause of k seconds or more separates two conversations.

Speaker Each speaker's utterances are treated as a monologue.

For each particular metric, we calculate the best baseline result among all of these. To find the best block size or pause length, we search over multiples of 5 between 5 and 300. This makes these baselines appear better than they really are, since their performance is optimized with respect to the test data. A complete table of baseline results is shown in Table 4.4.

We also calculate results for two more systems. One is a non-trivial baseline:

Time/mention Our system, using only time gap and mention-based features.

The other is an oracle, designed to test how well a segmentation system designed for meeting or lecture data might possibly do on this task. If no conversation were ever interrupted, such a system would be perfect (up to the limit of annotator agreement).

Perfect segments The transcript is divided into contiguous segments, where all utterances in a segment belong to the same conversation. The conversation assignments are determined by the human annotation whose agreement with the others is highest.

Our results, in Table 4.3, are encouraging. On average, annotators agree more with each other than with any artificial annotation, and more with our model than with the baselines. For the 1-to-1 accuracy metric, we cannot claim much beyond these general results. The range of human variation is quite wide, and there are annotators who are closer to baselines than to any other human annotator. As explained earlier, this is

Result	F-score	Notes
this model	43.4	
Wang and Oard (2009)	54	message expansion features
Shen et al. (2006)	61.2	different corpus

Table 4.5: Results reported by others on the same task.

because some human annotations are much more specific than others. For very specific annotations, the best baselines are short blocks or pauses. For the most general, marking all utterances the same does very well (although for all other annotations, it is extremely poor).

For the local metric, the results are much clearer. There is no overlap in the ranges; for every test annotation, agreement is highest with other annotators, then our model and finally the baselines. The most competitive baseline is one conversation per speaker, which makes sense, since if a speaker makes two comments in a four-utterance window, they are very likely to be related.

The Shen F-score metric seems to perform similarly to the one-to-one accuracy, unsurprising since they are both measures of global consistency. The largest difference between them is that the speaker baseline outperforms blocks and pauses in F-score (although not by very much), perhaps because it is more precise.

Shen et al. (2006) reports higher F-scores for their own best model: it obtains an F-score of 61.2, while our model's mean score is only 43.4. Because the corpora involved are different, we are unable to determine whether the difference is due to corpus effects, or differences between the models; we point out below that, on some corpora, our model can reach and even exceed this number. Better results are also reported in Wang and Oard (2009).

Mention information alone is not sufficient for disentanglement; with only name mention and time gap features, mean 1-to-1 is 38 and loc_3 is 69. However, name mention features are critical for our model. Without them, the classifier's development F-score drops from 71 to 56. The disentanglement system's test performance decreases proportionally; mean 1-to-1 falls to 36, and mean loc_3 to 63, essentially baseline performance. For some utterances, of course, name mentions provide the only reasonable clue to the correct decision, which is why humans mention names in the first place. But our system is probably overly dependent on them, since they are very reliable compared to our other features.

Because of the frequency with which conversations interleave, perfect segmentation alone is not sufficient to optimize either global metric, and generally does not outperform the baselines. For the local metric, however, it generally does better than the model. Here, performance depends mainly on whether the system can find the boundaries between one conversation and another, and it is less important to link the segments of a particular conversation to one another, since these different segments often lie outside the 3-utterance horizon. Systems designed to detect segment boundaries, like those for meetings, might contribute to improvement of this metric.

Adams (2008) annotates several larger corpora using our annotation software and protocol. Their recordings come from Freenode IRC's #IPHONE, #PHYSICS and #PYTHON channels, which, like #LINUX, are educational and technical support forums with some social interaction. We evaluate our system on these

	Annotators	Model	Baseline
Mean 1-to-1	76.30	60.01	48.86 (Pause 35)
Max 1-to-1	81.60	71.64	63.06 (Pause 65)
Min 1-to-1	72.52	56.63	44.88 (Blocks 80)
Mean *loc* $_3$	90.70	74.28	52.37 (Speaker)
Max *loc* $_3$	92.40	75.80	54.15 (Speaker)
Min *loc* $_3$	88.92	72.89	50.52 (Speaker)
Mean Shen F	80.27	69.19	55.61 (Speaker)
Max Shen F	84.33	71.63	57.80 (Speaker)
Min Shen F	75.70	66.65	46.86 (Blocks 65)

Table 4.6: Metric values for 5898 #IPHONE utterances recorded over 13 sessions.

	Annotators	Model	Baseline
Mean 1-to-1	81.65	45.44	31.96 (Pause 35)
Max 1-to-1	85.05	47.95	50.26 (Pause 65)
Min 1-to-1	78.89	42.90	51.69 (Blocks 80)
Mean *loc* $_3$	93.12	66.13	44.51 (Speaker)
Max *loc* $_3$	95.22	67.14	45.96 (Speaker)
Min *loc* $_3$	91.45	64.65	42.29 (Speaker)
Mean Shen F	84.77	57.75	54.15 (Speaker)
Max Shen F	88.14	60.04	55.44 (Speaker)
Min Shen F	81.67	56.05	55.76 (Blocks 65)

Table 4.7: Metric values for 4957 #PHYSICS utterances recorded over 12 sessions.

	Annotators	Model	Baseline
Mean 1-to-1	74.36	57.87	43.51 (Pause 35)
Max 1-to-1	80.49	61.00	54.71 (Pause 65)
Min 1-to-1	69.25	55.12	43.03 (Blocks 80)
Mean *loc* $_3$	87.33	75.64	49.41 (Speaker)
Max *loc* $_3$	89.52	77.34	50.87 (Speaker)
Min *loc* $_3$	85.22	73.83	47.73 (Speaker)
Mean Shen F	77.83	67.17	48.10 (Speaker)
Max Shen F	82.51	69.43	49.84 (Speaker)
Min Shen F	73.94	65.14	46.58 (Blocks 65)

Table 4.8: Metric values for 8726 #PYTHON utterances recorded over 13 sessions.

corpora, comparing our results against interannotator agreement[8] and against baselines. (For each metric, instead of searching for the best baseline, we compute the score for the baseline that did best on #LINUX; this minimizes use of the test data.) Results are shown in Tables 4.6, 4.7 and 4.8.

As discussed in Adams (2008), interannotator agreement is generally higher for their datasets than for ours (their mean one-to-one agreements range betweeen 82% and 74%, while ours is 53%). They suggest that the lowest score, for #PYTHON, is affected by the high amount of jargon and technical material; potentially this is also an issue for #LINUX. Another possibility is that their three annotators disagree less than our six simply because, with more people, one is more likely to sample the full range of possible human responses. Finally, it is possible that the thread structures are simply clearer in these three corpora, as suggested by the generally higher scores achieved by the model and baselines, though this does not explain the results on #PHYSICS, where interannotator agreement is excellent, but model performance is low.

In general, we observe the same trends in model performance on these corpora as on #LINUX; the model scores lie between human agreement and baselines. An exception is #PHYSICS, where global clustering performance is generally poor– max and min one-to-one are both higher for baselines than the model (for instance, max of 50% versus 48%)[9]. On the other two corpora, however, global scores are not only superior to the baselines, but substantially better than their #LINUX counterparts, with mean one-to-ones of 60% and 58%, compared to only 41% on #LINUX, implying either that the human annotators provide an easier target here, or that the model might be more useful for applications on these domains. These results also serve as context for the reported F-score of 61.2% from Shen et al. (2006); on the two easier datasets, our model attains similar or even higher scores (69% on #IPHONE, 67% on #PYTHON).

4.5 Specificity Tuning

While our analysis shows that individual annotators can produce more or less specific annotations of the same conversation, the system described above can produce only a single annotation (for any given set of training data) with a fixed specificity. Here we attempt to control the specificity parametrically, producing more and less specific annotations on demand, without retraining the classifier.

The parameter we choose to alter is the bias of our pairwise classifier. A maximum-entropy classifier has the form:

$$y(x) = \frac{1}{1 + exp(-(w \cdot x + b))} \tag{4.3}$$

Here w represents the vector of feature weights and b is the bias term; a positive b shifts all judgements toward high-confidence *same conversation* decisions and a negative b shifts them away. To alter the classifier, we add a constant λ to b. In general, increasing the number and confidence of *same* decisions leads to larger, coarser partitionings, while decreasing it creates smaller, finer ones. We measure specificity by examining

[8] For one-to-one overlap and loc_3, we report the numbers from Adams and Martell (2008), which our software replicates exactly. Since that study did not report Shen F, we compute it ourselves. Adams (2008) also gives detailed scores and statistics for each individual chat sessions.

[9] The mean is lower, probably because the *Pause of 35 seconds* baseline which optimized mean one-to-one overlap on #LINUX is not optimal for #PHYSICS, but higher max and min imply that it must be possible to find a baseline with higher mean as well.

the entropy of the output annotation. While entropy is generally an increasing function of λ, the relationship is not always smooth, nor is it completely monotonic. Figure 4.7 plots entropy as a function of λ.

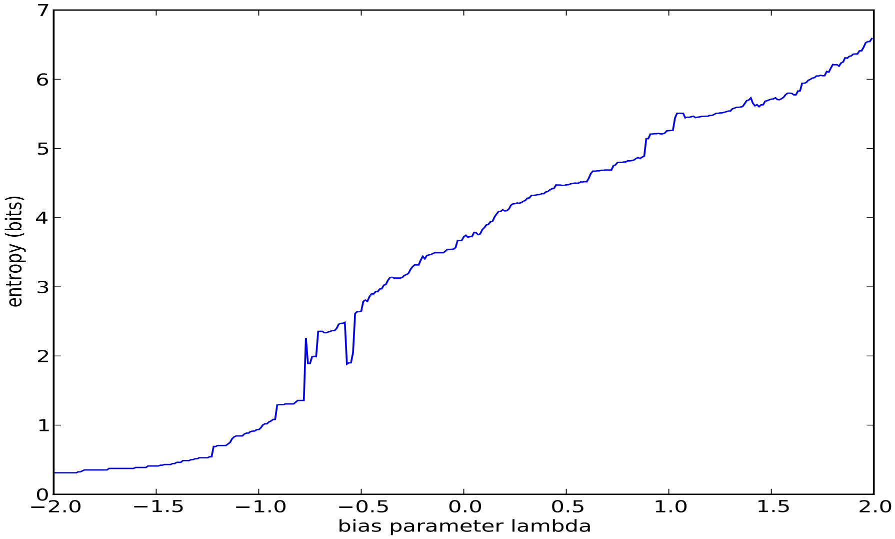

Figure 4.7: Entropy of the output annotation produced with bias factor λ on test data. $\lambda = 0$ corresponds to the unbiased system.

In figure 4.8, we plot the one-to-one match between each test annotation and the altered annotations produced by this method, as a function of the entropy. The unbiased system creates an annotation with entropy 3.7. While this yields reasonable results for all human annotations, each of the annotations has a point of higher performance at a different bias level. For instance, the line uppermost on the left side of the plot shows overlap with a human transcript whose entropy is 3.0 bits; lower-entropy system annotations correspond better with this annotator's judgements.

For each human annotation, we evaluate the tuned system's performance at the entropy level of the original annotation. (This point is marked by the large dot on each line in the figure.) To do this, we perform a line search over λ until we produce a clustering whose entropy is within .25 bits of the original's, then evaluate. In other words, we measure performance given an additional piece of supervision — the annotator's preferred specificity level.

Results on the one-to-one metric are fairly good: extreme and average scores are listed in table 4.9. The effects of this technique on the local metric are small (and in many cases negative). This is not entirely surprising, as the local metric is less sensitive to specificity of annotations. Slight positive effects occur only for the most and least specific annotation, which are presumably so extreme that specificity begins to have a slight effect even on local decisions.

Despite fairly large performance increases on the test set, we do not consider this technique really reliable, since the relationship between the bias parameter and final score is not smooth. Small changes in the bias can cause large shifts in entropy, and small changes in entropy can have large effects on quality. (For instance,

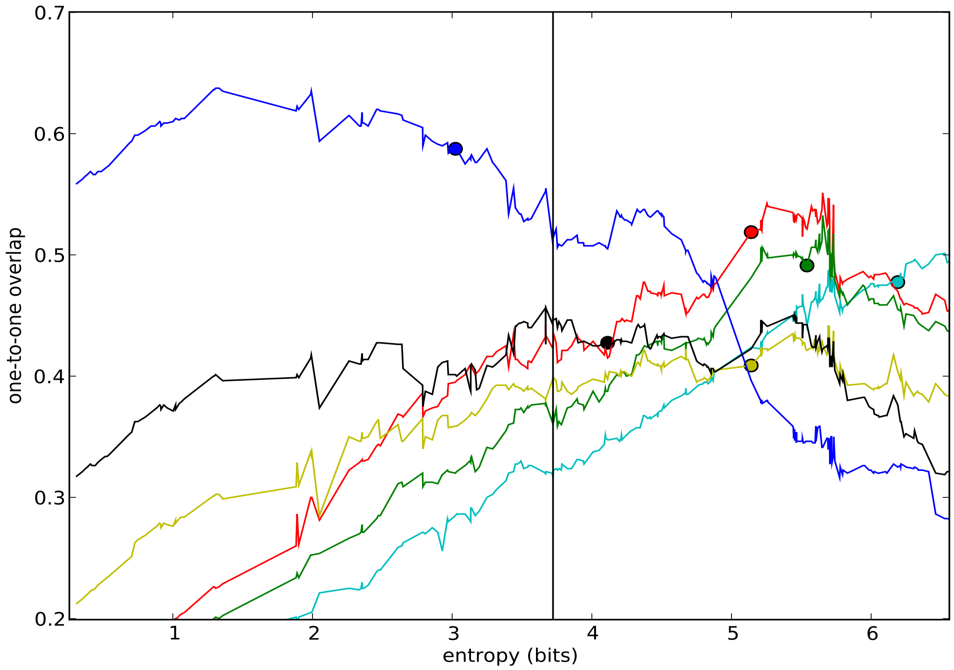

Figure 4.8: One-to-one accuracy between biased system annotations and each test annotation, as a function of entropy. The vertical line (at 3.72 bits) marks the scores obtained by the unbiased system with $\lambda = 0$. The large dot on each line is the score obtained at the entropy of the human annotation.

	Unbiased Model	Tuned Model
Mean 1-to-1	41.23	48.52
Max 1-to-1	52.13	58.75
Min 1-to-1	31.66	40.88
Mean loc_3	72.94	73.64
Max loc_3	74.70	75.87
Min loc_3	70.77	69.95

Table 4.9: Metric values between proposed annotations and human annotations on test data. The tuned model (evaluated at the entropy of the human annotations) improves on one-to-one accuracy but not on loc_3.

two annotations have a sharp decline in score at about entropy 5.7, losing about 5% of performance with a change of just over .1 bits.) Therefore it is not clear exactly how to choose a bias parameter which will yield good performance. Matching the entropy of a human annotation seems to work on the test data, but fails to improve scores on our development data. Moreover, although for methodological simplicity we assume access to the exact target entropy for each annotation, it is unlikely that a real user could express their desired specificity so precisely. Figuring out a way to let the user select the desired entropy remains a challenge.

4.6 Detecting Conversation Starts

In this section, we investigate better ways to find the beginnings of new conversations. In the pairwise linkage representation presented above, a new conversation is begun when none of the previous utterances is strongly linked to the current utterance. This representation spreads out the responsibility for detecting a new conversation over many pairwise decisions. We are inspired by the use of discourse-new classifiers (also called anaphoricity detectors) in coreference classification (Ng and Cardie, 2002a) to find NPs which begin coreferential chains. Oracle experiments show that a similar detector for utterances which begin conversations could improve disentanglement scores if it were available. We attempt to develop such a detector, but without much success.

As a demonstration of the gains possible if a good classifier could be developed, we show the oracle improvements possible on test, using an optimal new-conversation detector as a hard constraint on inference (table 4.10). The oracle detector detects a conversation start if it occurs in the majority of human annotations, and the inference algorithm is forced to start a new conversation if and only if the oracle has detected one. Good conversation detection is capable of improving not only one-to-one accuracy but local accuracy as well.

	Original Model	+Oracle New Conversations
Mean 1-to-1	41.23	46.75
Max 1-to-1	52.13	53.50
Min 1-to-1	31.66	42.13
Mean loc_3	72.94	73.90
Max loc_3	74.70	76.49
Min loc_3	70.77	70.72

Table 4.10: Metric values using an oracle new-conversation detector on test data.

We can track the performance of realistic, non-oracle new-conversation detection via the precision, recall and f-score of the *new conversation* class (table 4.11). As a starting point, we report the accuracy obtained by the pairwise-linkage model and greedy inference already presented. At 49% F-score, it is clearly not doing a good job.

It is possible to do better than this using information already represented in the pairwise classifier: the time since the speaker of the utterance last spoke (logarithmically bucketed), and whether the utterance mentions a name. A better representation for the problem allows the classifier to make somewhat more effective use of these features. For reasons we cannot explain, adding discourse features like the presence of a question or greeting does not improve performance. The simple classifier does improve slightly on the baseline,

	Precision	Recall	F-score
Pairwise system	56.08	43.44	48.96
Time/Mention Features	68.06	40.16	50.52
Human Annotators	64.30	61.70	61.14

Table 4.11: Precision, recall and F-score of the **new conversation** class on test data (average 81 conversations).

up to 51%. These test results, however, are somewhat surprising to us. On our development corpus, the corresponding scores are 69% and 75%. Since that corpus contains an average (over three annotations) of 34 conversations, it is likely that we were mislead by coincidentally good results.

On the development set, where the classifier works well, its decisions can be integrated with inference to yield substantial improvements in actual system performance. Mean loc_3 increases from 72% to about 78% and mean one-to-one accuracy from 41% to about 66%. However, we find no improvement at all on test, since the classifier has very low recall, and the resulting test annotations have far too few conversations.

4.7 Improved partitioning

4.7.1 Motivation

In the previous sections, we set up the chat disentanglement task as a correlation clustering problem in which we partition a graph to minimize the number of unrelated pairs that are clustered together, plus the number of related pairs that are separated. Unfortunately, this minimization problem is NP-hard (Ailon et al., 2008) and we resorted to a heuristic procedure to solve it.

In this section, we evaluate a variety of solutions for correlation clustering. We show, as in previous work on consensus clustering (Goder and Filkov, 2008), that local search can improve the solutions found by commonly-used methods. We investigate the relationship between the clustering objective and external evaluation metrics such as F-score and one-to-one overlap, showing that optimizing the objective is usually a reasonable aim, but that other measurements like number of clusters found should sometimes be used to reject pathological solutions. We prove that the best heuristics are quite close to optimal, using the first implementation of the semi-definite programming (SDP) relaxation to provide tighter bounds.

The specific algorithms we investigate are, of course, only a subset of the large number of possible solutions, or even of those proposed in the literature. We chose to test a few common, efficient algorithms that are easily implemented. Our use of a good bounding strategy means that we do not need to perform an exhaustive comparison; we will show that, though the methods we describe are not perfect, the remaining improvements possible with any algorithm are relatively small.

4.7.2 Previous Work

Correlation clustering was first introduced by Ben-Dor et al. (1999) to cluster gene expression patterns. The correlation clustering approach has several strengths. It does not require users to specify a parametric form for the clusters, nor to pick the number of clusters. Unlike fully unsupervised clustering methods, it can use training data to optimize the pairwise classifier, but unlike classification, it does not require samples from the specific clusters found in the test data. For instance, it can use messages about cars to learn a similarity function that can then be applied to messages about atheism.

Correlation clustering is a standard method for coreference resolution. It was introduced to the area by Soon et al. (2001), who describe the first-link heuristic method for solving it. Ng and Cardie (2002b) extend this work with better features, and develop the best-link heuristic, which finds better solutions. McCallum and Wellner (2004) explicitly describe the problem as correlation clustering and use an approximate technique (Bansal et al., 2004) to enforce transitivity. Recently Finkel and Manning (2008) show that the optimal ILP solution outperforms the first and best-link methods. Cohen and Richman (2002) experiment with various heuristic solutions for the cross-document coreference task of grouping references to named entities.

Finally, correlation clustering has proven useful in several discourse tasks. Barzilay and Lapata (2006) use it for content aggregation in a generation system. In Malioutov and Barzilay (2006), it is used for topic segmentation—since segments must be contiguous, the problem can be solved in polynomial time.

Bertolacci and Wirth (2007), Goder and Filkov (2008) and Gionis et al. (2007) conduct experiments on the closely related problem of *consensus clustering*, often solved by reduction to correlation clustering. The input to this problem is a set of clusterings; the output is a "median" clustering which minimizes the sum of (Rand) distance to the inputs. Although these papers investigate some of the same algorithms we use, they use an unrealistic lower bound, and so cannot convincingly evaluate absolute performance. Gionis et al. (2007) give an external evaluation on some UCI datasets, but this is somewhat unconvincing since their metric, the *impurity index*, which is essentially precision ignoring recall, gives a perfect score to the all-singletons clustering. The other two papers are based on objective values, not external metrics.[10]

A variety of approximation algorithms for correlation clustering with worst-case theoretical guarantees have been proposed: (Bansal et al., 2004; Ailon et al., 2008; Demaine et al., 2006; Charikar et al., 2005; Giotis and Guruswami, 2006). Researchers including (Ben-Dor et al., 1999; Joachims and Hopcroft, 2005; Mathieu and Schudy, 2008) study correlation clustering theoretically when the input is generated by randomly perturbing an unknown ground truth clustering.

4.7.3 Algorithms

We begin with some notation and a formal definition of the problem. Our input is a complete, undirected graph G with n nodes; each edge in the graph has a probability p_{ij} reflecting our belief as to whether nodes i and j come from the same cluster. Our goal is to find a clustering, defined as a new graph G' with edges $x_{ij} \in \{0, 1\}$, where if $x_{ij} = 1$, nodes i and j are assigned to the same cluster. To make this consistent, the edges must define an equivalence relationship: $x_{ii} = 1$ and $x_{ij} = x_{jk} = 1$ implies $x_{ij} = x_{ik}$.

Our objective is to find a clustering as consistent as possible with our beliefs—edges with high probability should not cross cluster boundaries, and edges with low probability should. We define w^+_{ij} as the cost of cutting an edge whose probability is p_{ij} and w^-_{ij} as the cost of keeping it. Mathematically, this objective can

[10] Bertolacci and Wirth (2007) gave normalized mutual information for one algorithm and data set, but almost all of their results study objective value only.

be written (Ailon et al., 2008; Finkel and Manning, 2008) as:

$$\min \sum_{ij:i<j} x_{ij} w^-_{ij} + (1 - x_{ij}) w^+_{ij}. \tag{4.4}$$

There are two plausible definitions for the costs w^+ and w^-, both of which have gained some support in the literature. We can take $w^+_{ij} = p_{ij}$ and $w^-_{ij} = 1 - p_{ij}$ (*additive* weights) as in (Ailon et al., 2008) and others, or $w^+_{ij} = \log(p_{ij}), w^-_{ij} = \log(1 - p_{ij})$ (*logarithmic* weights) as in (Finkel and Manning, 2008). The logarithmic scheme has a tenuous mathematical justification, since it selects a maximum-likelihood clustering under the assumption that the p_{ij} are independent and identically distributed given the status of the edge ij in the true clustering. If we obtain the p_{ij} using a classifier, however, this assumption is obviously untrue—some nodes will be easy to link, while others will be hard—so we evaluate the different weighting schemes empirically.

Greedy Methods

We use four greedy methods drawn from the literature; they are both fast and easy to implement. All of them make decisions based on the *net weight* $w^{\pm}_{ij} = w^+_{ij} - w^-_{ij}$.

These algorithms step through the nodes of the graph according to a permutation π. We try 100 random permutations for each algorithm and report the run which attains the best objective value (typically this is slightly better than the average run; we discuss this more in the experimental sections). To simplify the pseudocode we label the vertices $1, 2, \ldots n$ in the order specified by π. After this relabeling $\pi(i) = i$ so π need not appear explicitly in the algorithms.

Three of the algorithms are given in Figure 4.9. All three algorithms start with the empty clustering and add the vertices one by one. The BEST algorithm adds each vertex i to the cluster with the strongest $w^{\pm}$ connecting to i, or to a new singleton if none of the $w^{\pm}$ are positive. The FIRST algorithm adds each vertex i to the cluster containing the most recently considered vertex j with $w^{\pm}_{ij} > 0$. The VOTE algorithm adds each vertex to the cluster that minimizes the correlation clustering objective, i.e. to the cluster maximizing the total net weight or to a singleton if no total is positive. This is the algorithm used for disentanglement in previous sections.

Ailon et al. (2008) introduced the PIVOT algorithm, given in Figure 4.10, and proved that it is a 5-approximation if $w^+_{ij} + w^-_{ij} = 1$ for all i, j and π is chosen randomly. Unlike BEST, VOTE and FIRST, which build clusters vertex by vertex, the PIVOT algorithm creates each new cluster in its final form. This algorithm repeatedly takes an unclustered pivot vertex and creates a new cluster containing that vertex and all unclustered neighbors with positive weight.

Local Search

We use the straightforward local search previously used by Gionis et al. (2007) and Goder and Filkov (2008). The allowed *one element moves* consist of removing one vertex from a cluster and either moving it to another cluster or to a new singleton cluster. The best one element move (BOEM) algorithm repeatedly makes the most profitable best one element move until a local optimum is reached. *Simulated Annealing* (SA) makes a random single-element move, with probability related to the difference in objective it causes and the current

```
$k \leftarrow 0$ // number of clusters created so far
for $i = 1 \ldots n$ do
  for $c = 1 \ldots k$ do
    if Best then
      $Quality_c \leftarrow \max_{j \in C[c]} w^{\pm}_{ij}$
    else if First then
      $Quality_c \leftarrow \max_{j \in C[c] : w^{\pm}_{ij} > 0} j$
    else if Vote then
      $Quality_c \leftarrow \sum_{j \in C[c]} w^{\pm}_{ij}$
  $c^* \leftarrow \arg\max_{1 \leq c \leq k} Quality_c$
  if $Quality_{c^*} > 0$ then
    $C[c^*] \leftarrow C[c^*] \cup \{i\}$
  else
    $C[k$++$] \leftarrow \{i\}$ // form a new cluster
```

Figure 4.9: Best/First/Vote algorithms

```
$k \leftarrow 0$ // number of clusters created so far
for $i = 1 \ldots n$ do
  $P \leftarrow \bigcup_{1 \leq c \leq k} C[c]$ // Vertices already placed
  if $i \notin P$ then
    $C[k$++$] \leftarrow \{i\} \cup \{\, i < j \leq n : j \notin P$ and $w^{\pm}_{ij} > 0 \,\}$
```

Figure 4.10: Pivot algorithm by Ailon et al. (2008)

temperature. Our annealing schedule is exponential and designed to attempt $2000n$ moves for n nodes. We initialize the local search either with all nodes clustered together, or at the clustering produced by one of our greedy algorithms (in our tables, the latter is written, eg. Pivot/BOEM, if the greedy algorithm is Pivot).

4.7.4 Bounding with SDP

Although comparing different algorithms to one another gives a good picture of relative performance, it is natural to wonder how well they do in an absolute sense—how they compare to the optimal solution. For very small instances, we can actually find the optimum using ILP, but since this does not scale beyond a few hundred points (see Section 4.7.5), for realistic instances we must instead bound the optimal value. Bounds are usually obtained by solving a *relaxation* of the original problem: a simpler problem with the same objective but fewer constraints.

The bound used in previous work (Goder and Filkov, 2008; Gionis et al., 2007; Bertolacci and Wirth, 2007), which we call the *trivial bound*, is obtained by ignoring the transitivity constraints entirely. To optimize, we link ($x_{ij} = 1$) all the pairs where w^{+}_{ij} is larger than w^{-}_{ij}; since this solution is quite far from being a clustering, the bound tends not to be very tight.

To get a better idea of how good a real clustering can be, we use a semi-definite programming (SDP) relaxation to provide a better bound. Here we motivate and define this relaxation.

One can picture a clustering geometrically by associating cluster c with the standard basis vector $e_c =$

$(\underbrace{0,0,\dots,0}_{c-1},1,\underbrace{0,\dots,0}_{n-c}) \in \mathbb{R}^n$. If object i is in cluster c then it is natural to associate i with the vector $r_i = e_c$. This gives a nice geometric picture of a clustering, with objects i and j in the same cluster if and only if $r_i = r_j$. Note that the dot product $r_i \bullet r_j$ is 1 if i and j are in the same cluster and 0 otherwise. These ideas yield a simple reformulation of the correlation clustering problem:

$$\begin{gathered}\min_r \textstyle\sum_{i,j:i<j} (r_i \bullet r_j) w_{ij}^- + (1 - r_j \bullet r_j) w_{ij}^+ \\ \text{s.t. } \forall i \;\; \exists c : r_i = e_c\end{gathered}$$

To get an efficiently computable lower-bound we relax the constraints that the r_is are standard basis vectors, replacing them with two sets of constraints: $r_i \bullet r_i = 1$ for all i and $r_i \bullet r_j \geq 0$ for all i, j.

Since the r_i only appear as dot products, we can rewrite in terms of $x_{ij} = r_i \bullet r_j$. However, we must now constrain the x_{ij} to be the dot products of some set of vectors in $\mathbb{R}^n$. This is true if and only if the symmetric matrix $X = \{x_{ij}\}_{ij}$ is *positive semi-definite*. We now have the standard semi-definite programming (SDP) relaxation of correlation clustering (e.g. (Charikar et al., 2005; Mathieu and Schudy, 2008)):

$$\begin{gathered}\min_x \textstyle\sum_{i,j:i<j} x_{ij} w_{ij}^- + (1 - x_{ij}) w_{ij}^+ \\ \text{s.t.} \begin{cases} x_{ii} = 1 & \forall i \\ x_{ij} \geq 0 & \forall i, j \\ X = \{x_{ij}\}_{ij} \text{ PSD} & \end{cases} .\end{gathered}$$

This SDP has been studied theoretically by a number of authors; we mention just two here. Charikar et al. (2005) give an approximation algorithm based on rounding the SDP which is a 0.7664 approximation for the problem of maximizing agreements. Mathieu and Schudy (2008) show that if the input is generated by corrupting the edges of a ground truth clustering B independently, then the SDP relaxation value is within an additive $O(n\sqrt{n})$ of the optimum clustering. They further show that using the PIVOT algorithm to round the SDP yields a clustering with value at most $O(n\sqrt{n})$ more than optimal.

4.7.5 Experiments

Scalability

Using synthetic data, we investigate the scalability of the linear programming solver and SDP bound. To find optimal solutions, we pass the complete ILP[11] to CPLEX. This is reasonable for 100 points and solvable for 200; beyond this point it cannot be solved due to memory exhaustion. As noted below, despite our inability to compute the LP bound on large instances, we can sometimes prove that they must be worse than SDP bounds, so we do not investigate LP-solving techniques further.

The SDP has fewer constraints than the ILP ($O(n^2)$ vs $O(n^3)$), but this is still more than many SDP solvers can handle. For our experiments we used one of the few SDP solvers that can handle such a large number of constraints: Christoph Helmberg's ConicBundle library (Helmberg, 2009; Helmberg, 2000). This solver can handle several thousand datapoints. It produces loose lower-bounds (off by a few percent) quickly

[11] Consisting of the objective plus constraints $0 \leq x_{ij} \leq 1$ and triangle inequality (Ailon et al., 2008).

but converges to optimality quite slowly; we err on the side of inefficiency by running for up to 60 hours. Of course, the SDP solver is only necessary to bound algorithm performance; our solvers themselves scale much better.

Twenty Newsgroups

In this section, we test our approach on a typical benchmark clustering dataset, 20 Newsgroups, which contains posts from a variety of Usenet newsgroups such as `rec.motorcycles` and `alt.atheism`. Since our bounding technique does not scale to the full dataset, we restrict our attention to a subsample of 100 messages[12] from each newsgroup for a total of 2000—still a realistically large-scale problem. Our goal is to cluster messages by their newsgroup of origin. We conduct experiments by holding out four newsgroups as a training set, learning a pairwise classifier, and applying it to the remaining 16 newsgroups to form our affinity matrix.[13]

Our pairwise classifier uses three types of features previously found useful in document clustering. First, we bucket all words[14] by their log document frequency (for an overview of TF-IDF see (Joachims, 1997)). For a pair of messages, we create a feature for each bucket whose value is the proportion of shared words in that bucket. Secondly, we run LSA (Deerwester et al., 1990) on the TF-IDF matrix for the dataset, and use the cosine distance between each message pair as a feature. Finally, we use the same type of shared words features for terms in message subjects. We make a training instance for each pair of documents in the training set and learn via logistic regression.

The classifier has an average F-score of 29% and an accuracy of 88%—not particularly good. (An affinity matrix produced by the classifier is shown in Figure 4.11.) We should emphasize that the clustering task for 20 newsgroups is much harder than the more common classification task—since our training set is entirely disjoint with the testing set, we can only learn weights on feature categories, not term weights. Our aim is to create realistic-looking data on which to test our clustering methods, not to motivate correlation clustering as a solution to this specific problem. In fact, Zhong and Ghosh (2003) report better results using generative models.

We evaluate our clusterings using three different metrics (see Meila (2007) for an overview of clustering metrics). The *Rand* measure counts the number of pairs of points for which the proposed clustering agrees with ground truth. This is the metric which is mathematically closest to the objective. However, since most points are in different clusters, any solution with small clusters tends to get a high score. Therefore we also report the more sensitive *F-score* with respect to the minority (“same cluster”) class. We also report the *one-to-one* score, which measures accuracy over single points. For this metric, we calculate a maximum-weight matching between proposed clusters and ground-truth clusters, then report the overlap between the two.

When presenting objective values, we locate them within the range between the trivial lower bound discussed in Section 4.7.4 and the objective value of the singletons clustering ($x_{ij} = 0, i \neq j$). On this scale, lower is better; 0% corresponds to the trivial bound and 100% corresponds to the singletons clustering. It is

[12] Available as `mini_newsgroups.tar.gz` from the UCI machine learning repository.

[13] The experiments below are averaged over four disjoint training sets.

[14] We omit the message header, except the subject line, and also discard word types with fewer than 3 occurrences.

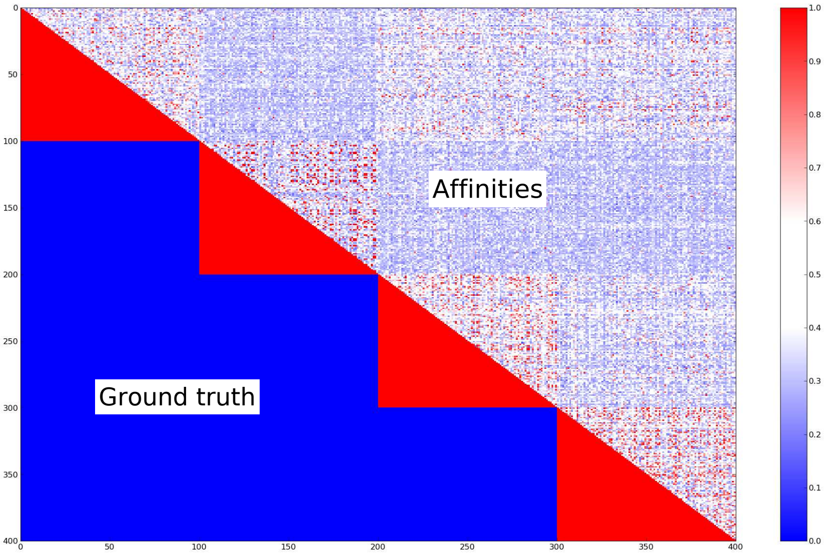

Figure 4.11: Affinity matrix produced by the classifier for 20 Newsgroups. Above the line, red points have positive net weight (ought to be clustered together) and blue points have negative net weight (ought to be separated). Below the line is ground truth.

possible to find values greater than 100%, since some particularly bad clusterings have objectives worse than the singletons clustering. Plainly, however, real clusterings will not have values as low as 0%, since the trivial bound is so unrealistic.

Our results are shown in Table 4.12. The best results are obtained using logarithmic weights with VOTE followed by BOEM; reasonable results are also found using additive weights, and annealing, VOTE or PIVOT followed by BOEM. On its own, the best greedy scheme is VOTE, but all of them are substantially improved by BOEM. First-link is by far the worst. Our use of the SDP lower bound rather than the trivial lower-bound of 0% reduces the gap between the best clustering and the lower bound by over a factor of ten. It is easy to show that the LP relaxation can obtain a bound of at most 50%[15]—the SDP beats the LP in both runtime and quality!

We analyze the correlation between objective values and metric values, averaging Kendall's tau[16] over the four datasets (Table 4.13). Over the entire dataset, correlations are generally good (large and negative), showing that optimizing the objective is indeed a useful way to find good results. We also examine correlations for the solutions with objective values within the top 10%. Here the correlation is much poorer; selecting the solution with the best objective value will not necessarily optimize the metric, although the correspondence is slightly better for the log-weights scheme. The correlations do exist, however, and so the solution with the

[15] The solution $x_{ij} = \frac{1}{2}\mathbb{1}\left(w_{ij}^- > w_{ij}^+\right)$ for $i < j$ is feasible in the LP.

[16] The standard Pearson correlation coefficient is less robust to outliers, which causes problems for this data.

Logarithmic Weights

	Obj	Rand	F	1-1
SDP bound	51.1%	-	-	-
VOTE/BOEM	55.8%	93.80	33	41
SA	56.3%	93.56	31	36
PIVOT/BOEM	56.6%	93.63	32	39
BEST/BOEM	57.6%	93.57	31	38
FIRST/BOEM	57.9%	93.65	30	36
VOTE	59.0%	93.41	29	35
BOEM	60.1%	93.51	30	35
PIVOT	100%	90.85	17	27
BEST	138%	87.11	20	29
FIRST	619%	40.97	11	8

Additive Weights

	Obj	Rand	F	1-1
SDP bound	59.0%	-	-	-
SA	63.5%	93.75	32	39
VOTE/BOEM	63.5%	93.75	32	39
PIVOT/BOEM	63.7%	93.70	32	39
BEST/BOEM	63.8%	93.73	31	39
FIRST/BOEM	63.9%	93.58	31	37
BOEM	64.6%	93.65	31	37
VOTE	67.3%	93.35	28	34
PIVOT	109%	90.63	17	26
BEST	165%	87.06	20	29
FIRST	761%	40.46	11	8

Table 4.12: Score of the solution with best objective for each solver, averaged over newsgroups training sets, sorted by objective.

best objective value is typically slightly better than the median. Figure 4.12 shows a scatter-plot of objective values versus one-to-one scores, with the top 10% zoomed in.

In Figure 4.13, we show the distribution of one-to-one scores obtained (for one specific dataset) by the best solvers. From this diagram, it is clear that log-weights and VOTE/BOEM usually obtain the best scores for this metric, since the median is higher than other solvers' upper quartile scores. All solvers have quite high variance, with a range of about 2% between quartiles and 4% overall. We omit the F-score plot, which is similar, for space reasons.

Chat Disentanglement

We now apply the methods developed here to the chat disentanglement task, using the same 800-utterance test set and 6 test annotations as above. The affinity matrix is shown in Figure 4.14.

In this dataset, the SDP is a more moderate improvement over the trivial lower bound, reducing the gap between the best clustering and best lower bound by a factor of about 3 (Table 4.14).

Optimization of the objective does not correspond to improvements in the global metrics (Table 4.14); for

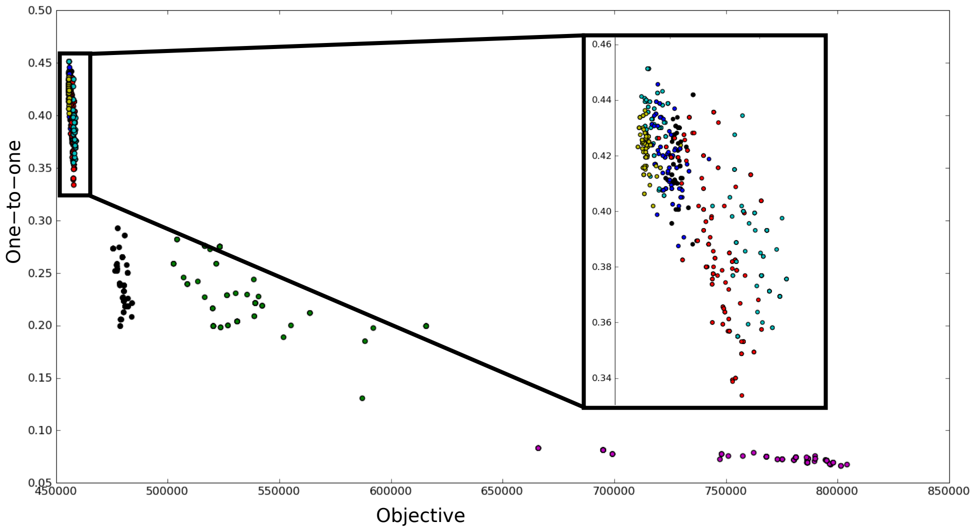

Figure 4.12: Correlation between objective values and one-to-one scores for newsgroup data. Different colors represent different solvers; the objective (minimum is better) is plotted on the X axis while one-to-one scores are on the Y axis.

	Rand	F	1-1
Log-wt	-.60	-.73	-.71
Top 10 %	-.14	-.22	-.24
Add-wt	-.60	-.67	-.65
Top 10 %	-.13	-.15	-.14

Table 4.13: Kendall’s tau correlation between objective and metric values, averaged over newsgroup datasets, for all solutions and top 10% of solutions.

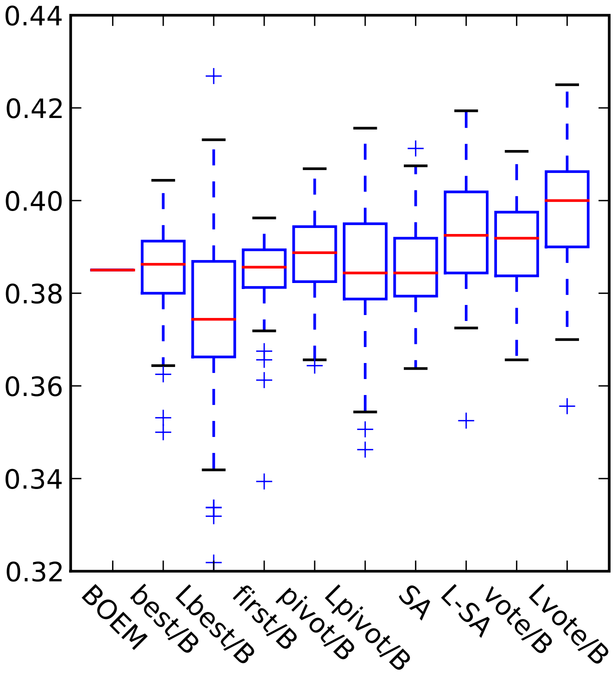

Figure 4.13: Box-and-whisker diagram (outliers as +) for one-to-one scores obtained by the best few solvers on a particular newsgroup dataset. L means using log weights. B means improved with BOEM.

instance, the best objectives are attained with FIRST/BOEM, but VOTE/BOEM yields better one-to-one and F scores. Correlation between the objective and these global metrics is extremely weak (Table 4.16). The local metric is somewhat correlated.

Local search does improve metric results for each particular greedy algorithm. For instance, when BOEM is added to VOTE (with log weights), one-to-one increases from 44% to 46%, local from 72% to 73% and F from 48% to 50%. This represents a moderate improvement on the inference scheme we described in section 4.3.2. This scheme used voting with logarithmic weights, but rather than performing multiple runs over random permutations, it processed utterances in the order they occur. (We experimented with processing in order; the results are unclear, but there is a slight trend toward worse performance, as in this case.) These results (also shown in the table) are 41% one-to-one, 73% local and .44% F-score. Our improvement on the global metrics (12% relative improvement in one-to-one, 13% in F-score) is modest, but was achieved with better inference on exactly the same input.

Since the objective function fails to distinguish good solutions from bad ones, we examine the types of solutions found by different methods in the hope of explaining why some perform better than others. In this setting, some methods (notably local search run on its own or from a poor starting point) find far fewer clusters than others (Table 4.15; log weights not shown but similar to additive). Since the classifier abstains for utterances more than 129 seconds apart, the objective is unaffected if very distant utterances are linked on the basis of little or no evidence; this is presumably how such large clusters form. (This raises the question of whether abstentions should be given weaker links with $p < .5$. We leave this for future work.) Algorithms

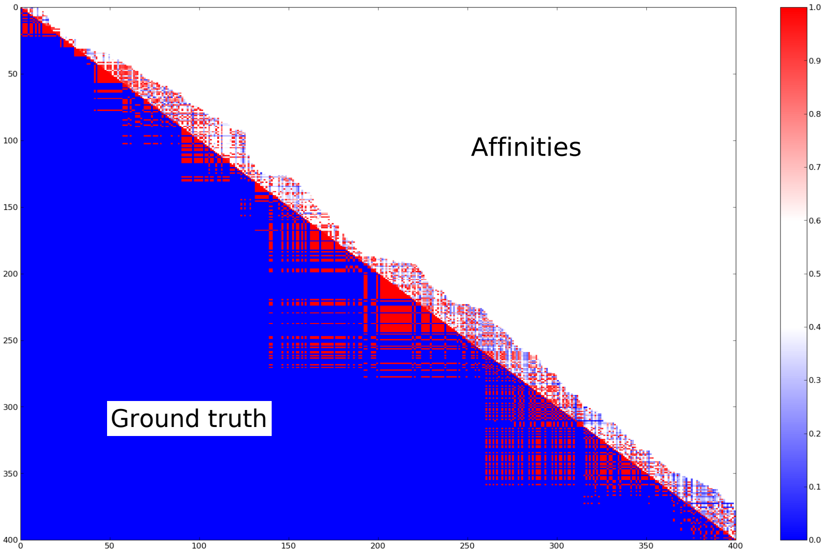

Figure 4.14: Affinity matrix produced by the classifier for chat test data. Above the line, red points have positive net weight (ought to be clustered together) and blue points have negative net weight (ought to be separated). Below the line is ground truth.

which find reasonable numbers of clusters (VOTE, PIVOT, BEST and local searches based on these) all achieve good metric scores, although there is still no reliable way to find the best solution among this set of methods.

4.7.6 Conclusions

It is clear from these results that heuristic methods can provide good correlation clustering solutions on datasets far too large for ILP to scale. The particular solver chosen has a substantial impact on the quality of results obtained, in terms of external metrics as well as objective value.

For general problems, our recommendation is to use log weights and run VOTE/BOEM. This algorithm is fast, achieves good objective values, and yields good metric scores on our datasets. Although objective values are usually only weakly correlated with metrics, our results suggest that slightly better scores can be obtained by running the algorithm many times and returning the solution with the best objective. This may be worth trying even when the datapoints are inherently ordered, as in chat.

Whatever algorithm is used to provide an initial solution, we advise the use of local search as a post-process. BOEM always improves both objective and metric values over its starting point.

The objective value is not always sufficient to select a good solution (as in the chat dataset). If possible, experimenters should check statistics like the number of clusters found to make sure they conform roughly to expectations. Algorithms that find far too many or too few clusters, regardless of objective, are unlikely to be useful. This type of problem can be especially dangerous if the pairwise classifier abstains for many pairs

Log Weights

	Obj	1-1	Loc_3	Shen F
SDP bound	13.0%	-	-	-
First/BOEM	19.3%	41	**74**	44
Vote/BOEM	20.0%	**46**	**73**	**50**
SA	20.3%	42	**73**	45
Best/BOEM	21.3%	43	**73**	47
BOEM	21.5%	22	72	21
Pivot/BOEM	22.0%	**45**	72	**50**
Vote	26.3%	44	72	48
Best	37.1%	40	67	44
Pivot	44.4%	39	66	44
First	58.3%	39	62	41

Additive Weights

	Obj	1-1	Loc_3	Shen F
SDP bound	16.2%	-	-	-
First/BOEM	21.7%	40	**73**	44
BOEM	22.3%	22	**73**	20
Best/BOEM	22.7%	44	**74**	**49**
Vote/BOEM	23.3%	**46**	**73**	**50**
SA	23.8%	41	72	46
Pivot/BOEM	24.8%	**46**	73	**50**
Vote	30.5%	44	71	**49**
EC '08	-	41	**73**	44
Best	42.1%	43	69	47
Pivot	48.4%	38	67	44
First	69.0%	40	59	41

Table 4.14: Score of the solution with best objective found by each solver on the chat test dataset, averaged over 6 annotations, sorted by objective.

of points.

SDP provides much tighter bounds than the trivial bound used in previous work, although how much tighter varies with dataset (about 12 times smaller for newsgroups, 3 times for chat). This bound can be used to evaluate the absolute performance of our solvers; the Vote/BOEM solver whose use we recommend is within about 5% of optimality. Some of this 5% represents the difference between the bound and optimality; the rest is the difference between the optimum and the solution found. If the bound were exactly optimal, we could expect a significant improvement on our best results, but not a very large one—especially since correlation between objective and metric values grows weaker for the best solutions. While it might be useful to investigate more sophisticated local searches in an attempt to close the gap, we do not view this as a priority.

	Num clusters
Max human annotator	*128*
PIVOT	122
VOTE	99
PIVOT/BOEM	89
VOTE/BOEM	86
Mean human annotator	*81*
BEST	70
FIRST	70
Elsner and Charniak (2008b)	*63*
BEST/BOEM	62
SA	57
FIRST/BOEM	54
Min human annotator	*50*
BOEM	7

Table 4.15: Average number of clusters found (using additive weights) for chat test data.

	1-1	Loc_3	Shen F
Log-wt	-.40	-.68	-.35
Top 10 %	.14	-.15	.15
Add-wt	-.31	-.67	-.25
Top 10 %	-.07	-.22	.13

Table 4.16: Kendall's tau correlation between objective and metric values for the chat test set, for all solutions and top 10% of solutions.

4.8 Conclusion

This work provides a corpus of annotated data for chat disentanglement, which, along with our proposed metrics, provides a solid foundation for further experiments. Our annotations are consistent with one another, especially with respect to local agreement. We show that simple coherence features are helpful in disentanglement, and that our baseline model can outperform a variety of baselines.

From the perspective of mainstream coherence modeling, disentanglement poses a variety of challenges. The domain is conversational rather than informative, and informal rather than formal. Moreover, systems like parsers, dictionaries, or even POS taggers, which we expect to have available for normal applications, are hard to develop for IRC chat. In addition, the task itself is algorithmically different than ordering. In the next chapter, we apply our local coherence models in spite of these difficulties, and evaluate their performance relative to the baseline model we have just presented.

Chapter 5

Coherence models for disentanglement

In previous chapters, we constructed a variety of local coherence models and evaluated them on sentence-ordering tasks on news data. We also presented the task of chat disentanglement, motivating it as another evaluation task for local coherence models– but the model we actually used to perform the task was relatively unsophisticated, relying on simple word overlap instead of the more complex features we used for news. Our main contribution in this chapter is to *apply sophisticated coherence models to disentanglement, demonstrating that they improve over the baseline* (section 5.4).

There are several differences between disentanglement and the newswire sentence-ordering tasks typically used to evaluate coherence models. As opposed to news data, internet chat comes from a different domain, with different topics and structural properties. There are no reliable syntactic annotations available. The disentanglement task measures different capabilities of a model, since it compares documents that are not permuted versions of one another. Finally, full disentanglement requires a large-scale search, which is computationally difficult. Therefore, we move toward disentanglement in stages, carrying out a series of experiments to measure the contribution of each of these factors.

As an intermediary between newswire and internet chat, we adopt the SWITCHBOARD (SWBD) corpus. SWBD contains recorded telephone conversations with known topics and hand-annotated parse trees; this allows us to control for the performance of our parser and other informational resources. To compare the two algorithmic settings, we use SWBD for ordering experiments (section 5.3), and also artificially entangle pairs of telephone dialogues to create synthetic multiparty transcripts which we can disentangle (section 5.4). Finally, we present results on actual internet chat corpora (section 5.5).

5.1 Models

We begin by briefly reviewing the models we intend to evaluate, all of which are more fully described in other sections of the thesis. For the dialogue experiments below, we train the models on SWBD (our test-training splits are explained below), augmented with a larger set of automatically parsed conversations from the

FISHER corpus[1]. Since the two corpora are quite similar, FISHER is a useful source for extra data; McClosky et al. (2010) uses it for this purpose in parsing experiments. (We continue to use SWBD/FISHER even for experiments on IRC, because we do not have enough disentangled training data to learn lexical relationships.)

Entity grid

We use a generative entity grid with a log-linear estimator using independent features, all-noun mention detection, and six sentences of history; the basic framework of the grid is explained in section 2.1.1, and the particular estimation procedure in section 3.4. The choice of this model variant was motivated by development experiments on SWBD ordering and disentanglement.

Notice that this is *not* the best entity grid for ordering WSJ. Using independent rather than cross-product features and six rather than two sentences of context each reduce performance slightly on newswire data, but increase it slightly on phone dialogues. This difference occurs mostly because nominal mentions of the same entity appear much more widely separated in dialogue than in news, requiring use of more context, but making the different components of the context relate more loosely to one another. The extended feature entity grid of subsection 3.4.1 also does not improve performance on SWBD, probably for the same reason that the discourse-new model does not– differences in referring behavior between the two domains are discussed below.

Topical entity grid

The topical entity grid is described in subsection 3.4.2. Again, we use a six-sentence history, though a two-sentence history works somewhat better for news.

IBM-1

The IBM model is described in section 2.1.2, and we use it exactly as presented there.

Pronouns

We use the pronoun model of Charniak and Elsner (2009), in the manner described in section 3.3, for news. For dialogue, we adapt the parameters by using them as a starting point, then running a few iterations of EM training on the FISHER data.

Discourse-new

Our discourse-new model is described in section 3.2, and we use it exactly as presented there.

[1] This corresponds to our use of WSJ augmented with automatically parsed NANC for news.

5.2 Chat-specific features

As we point out in chapter 4, good disentanglement depents on non-linguistic information as well as lexical features. In fact, the results in section 4.4 show that timestamps and speaker identities are usually *better* cues than utterance content. Therefore, we evaluate our coherence models in combination with three essential non-linguistic features, for which we build simple generative models.

The first feature is the time gap between one utterance and the next within the same thread. Consistent short gaps are a sign of normal turn-taking behavior; long pauses do occur, but much more rarely (Aoki et al., 2003). In our dataset, all time gaps are rounded to the nearest second. We model the distribution of time gaps using a histogram, with bucket sizes chosen adaptively so that each bucket contains at least four datapoints.

The second feature is speaker identity; conversations usually involve a small subset of the total number of speakers, and a few core speakers make most of the utterances. We capture these facts by modeling the distribution of speakers within a conversation as draws from a Chinese Restaurant Process (CRP) (Aldous, 1985) (tuned to maximize development peformance). The CRP's "rich-get-richer" dynamics capture our two intuitions, favoring conversations dominated by a few vociferous speakers.

Finally, we model name mentioning. Speakers in IRC chat often use their addressee's names to coordinate the chat (O'Neill and Martin, 2003), and this is a powerful source of information (section 4.4). Our model is fairly simple. It classifies each utterance into either the start or continuation of a conversational turn, by checking if the previous utterance was spoken by the same person. Given this status, it computes the probability of three types of mentioning: no name mention, a mention of someone who has previously spoken in the conversation, or a mention of someone who has not spoken. (The third of these options, of course, is extremely uncommon; this accounts for most of the model's predictive power). We learn parameters for the mention model on the #LINUX training data.

5.3 Ordering SWITCHBOARD

In this section, we investigate the performance of local coherence models on conversational data from SWBD. SWBD is a conversational domain, but the preselected topical prompts make content more predictable than on IRC, and the availability of gold parse trees means we are not limited by the lower performance of parsers outside the news domain (Foster, 2010; McClosky et al., 2010). Therefore, we can use it to measure the degree to which our models are news-specific as opposed to domain-independent.

We begin by running the same discrimination task used to evaluate sentence ordering performance on news. This enables us to measure differences in model perrformance caused by switching to a conversational domain. For SWBD, rather than compare permutations of the individual utterances, we permute conversational turns (sets of consecutive utterances by each speaker), since turns are natural discourse units in conversation. We take documents numbered 2000–3999 as training/development and the remainder as test, yielding 505 training and 153 test documents; we evaluate 20 permutations per document. As a comparison, we also show results for the same models on WSJ, using the train/test split from chapter 3.

In Table 5.1, we show the results for individual models, for the combined model, and ablation results

	WSJ	SWBD
EGrid	76.4‡	86.0
Topical EGrid	71.8‡	70.9‡
IBM-1	77.2‡	84.9†
Pronouns	69.6‡	71.7‡
Disc-new	72.3‡	55.0‡
Combined	81.9	88.4
-EGrid	81.0	87.5
-Topical EGrid	**82.2**	**90.5**
-IBM-1	79.0‡	88.9
-Pronouns	81.3	88.5
-Disc-new	**82.2**	88.4

Table 5.1: Discrimination F scores on news and dialogue. We tested significance of all individual and ablated models versus the combined model; ‡ indicates a difference significant at p=.001, and † at .05.

for mixtures without each component. WSJ is more difficult than SWBD overall because, on average, news articles are shorter than SWBD conversations. Short documents are harder, because permuting disrupts them less. The best SWBD result is 91%; the best WSJ result is 82% (both for mixtures without the topical entity grid).

Controlling for the fact that discrimination is easier on SWBD, most of the individual models perform similarly in both corpora. The strongest models in both cases are the entity grid and IBM-1 (at about 77% for news, 85% for dialogue). Pronouns and the topical entity grid are weaker. The major outlier is the discourse-new model, whose performance drops from 72% for news to only 55%, just above chance, for conversation.

The model combination results show that all the models are quite closely correlated, since leaving out any single model does not degrade the combination very much (only one of the differences is statistically significant). However, in nearly all cases, the individual models are significantly worse than the combination, showing that multiple sources of information are important. The most critical in news is IBM-1 (decreasing performance by 3% when removed); in conversation, it is the entity grid (decreasing by about 1%). The topical entity grid actually has a (nonsignificant) *negative* impact on combined performance, implying that its predictive power in this setting comes mainly from information that other models also capture, but that it is noisier and less reliable. In each domain, the combined models outperform the best single model, showing the information provided by the weaker models is not completely redundant.

Overall, these results suggest that most previously proposed local coherence models are domain-general; they work on conversation as well as news. The exception is the discourse-newness model, which benefits most from the specific conventions of a written style. Full names with titles (like "President Barack Obama") are more common in news, while conversation tends to involve fewer completely unfamiliar entities and more cases of bridging reference, in which grounding information is given implicitly (Nissim, 2006). Due to its poor performance, we omit the discourse-newness model in our remaining experiments.

The pronoun model also decreases in performance, suggesting some lack of domain generality, though not as much as the discourse-new model. This decrease is partly explained by the lower rate of pronoun use in conversational data. In Table 5.2, we compare pronoun use in our different corpora. System performance

Corpus	Deictics	Pronouns	Resolvable pronouns
WSJ	.04	0.64	0.52
SWBD	.12	1.18	0.39
#PYTHON	.09	0.92	0.31

Table 5.2: Frequency of pronominals per sentence in three corpora, for all pronominals, pronouns only, and pronouns useful to the resolution system only.

depends on the number of pronouns resolvable by our model. This model is a version of Charniak and Elsner (2009), except that resolution of first and second-person pronouns is disabled, since these pattern differently in dialogue than in news[2]. We also discard reflexive pronouns, which always resolve intrasententially. A WSJ article contains, on average, .5 resolvable pronouns per sentence; SWBD contains only .4, and #PYTHON, an IRC chat corpus, contains .3. Thus, the pronoun model is less effective on these corpora.

Although resolvable pronouns are less common in conversation, pronominals in general are more common. The category of all pronouns (including reflexives and first and second person pronouns) is more common in SWBD than WSJ, averaging 1.2 per sentence versus .6. The deictics this, that, these and those are also more common. While conversation appears to use pronominals more often than news in general, these pronouns often refer to the discourse participants, and are not useful for coherence purposes. Deictics, on the other hand, are a potentially useful extension to the model, and we predict that an anaphora resolution system which modeled them would be more successful on conversation.

5.4 Disentangling SWBD

We now turn to the task of disentanglement, testing whether models that are good at ordering also do well in this new setting. We would like to hold the domain constant, but we do not have any disentanglement data recorded from naturally occurring speech, so we create synthetic instances by merging pairs of SWBD dialogues. Doing so creates an artificial transcript in which two pairs of people appear to be talking simultaneously over a shared channel.

The situation is somewhat contrived in that each pair of speakers converses only with each other, never breaking into the other pair's dialogue and rarely using devices like name mentioning to make it clear who they are addressing. Since this makes speaker identity a perfect cue for disentanglement, we do not use it in this section. The only chat-specific model we use is time.

Because we are not using a speaker identity model, we remove all utterances which do not contain a noun before constructing our synthetic conversations– these are mostly backchannels like "Yeah". Such utterances cannot be correctly assigned by any of the coherence models we are evaluating, but could easily be dealt with by associating them with the nearest utterance from the same speaker.

Once the backchannels are stripped, we can create a synthetic transcript. For each dialogue, we first simulate timestamps by sampling the number of seconds between each utterance and the next from a discretized Gaussian: $\lfloor N(0, 2.5) \rfloor$. The interleaving of the conversations is dictated by the timestamps. We truncate the

[2] Resolvable first and second-person pronouns in news usually appear inside quotations and refer to the speaker of the quote.

longer conversation at the length of the shorter; this ensures a baseline score of 50% for the degenerate model that assigns all utterances to the same conversation.

We create synthetic instances of two types– those where the two entangled conversations had different topical prompts and those where they were the same. The specific topical prompts provided to the speakers are listed in the SWBD metadata. For instance, one topic is:

> FISHING 360 FIND OUT WHAT KIND OF FISHING THE OTHER CALLER ENJOYS. DO YOU HAVE SIMILAR OR DIFFERENT INTERESTS IN THE KIND OF FISHING YOU ENJOY?

Our development set consists of 110 instances of each type (210 in all) constructed from dialogues with indices beginning with 2; 10 of these are used to train mixtures, while the other 100 are used for validation. Our test set consists of 100 instances of each type from dialogues with indices beginning with 4.

When disentangling, we treat each conversational thread as independent of the others. In other words, the probability of the entire transcript is the product of the probabilities of the component threads. Our objective is to find the set of threads maximizing this probability. To do so, we use a search technique explained in subsection 5.4.2.

As a comparison to our own results, we use the model of chapter 4 as a baseline. To make their implementation comparable to ours, in this section we constrain it to find only two threads.

5.4.1 Disentangling a single utterance

Our first disentanglement task is to correctly assign a single utterance, given the true structure of the rest of the transcript. For each utterance, we compare two versions of the transcript, the original, and a version where it is swapped into the other thread. Our accuracy measures how often our models prefer the original. Unlike full-scale disentanglement, this task does not require a computationally demanding search, so it is possible to run experiments quickly. We also use it to train our mixture models for disentanglement, by construct a training example for each utterance i in our training transcripts.

Since the model from chapter 4 maximizes a correlation clustering objective which sums up independent edge weights, we can disentangle a single sentence exactly. (To do so, we set all other sentences to their correct clusters, then sum the contributions of all edges which lead to the sentence of interest and pick the maximum.) Since this replaces the approximate inference used for full partitioning with exact inference, results depend only on the classifier, rather than the search strategy adopted, so the improved inference methods of section 4.7 would have no effect.

Our results are shown in Table 5.3. Again, results for individual models are above the line, then our combined model, and finally ablation results for mixtures omitting a single model. The results show that, for a pair of dialogues that differ in topic, our best model can assign a single sentence with 87% accuracy. For the same topic, the accuracy is 80%. In each case, these results improve on the model from chapter 4, which scores 78% and 74%[3].

[3] Our results on disentanglement tasks are reported as averages over utterances, but we cannot test their significance, because utterances in the same document are not independent. See section 2.3 for details.

	Different	Same	Avg.
EGrid	80.2	72.9	76.6
Topical EGrid	81.7	73.3	77.5
IBM-1	70.4	66.7	68.5
Pronouns	53.1	50.1	51.6
Time	58.5	57.4	57.9
Combined	**86.8**	**79.6**	**83.2**
-EGrid	86.0	79.1	82.6
-Topical EGrid	85.2	78.7	81.9
-IBM-1	86.2	78.7	82.4
-Pronouns	**86.8**	79.4	83.1
-Time	84.5	76.7	80.6
E+C '08	78.2	73.5	75.8

Table 5.3: Average accuracy for disentanglement of a single utterance on 200 synthetic multiparty conversations from SWBD test.

Changing to this new task has a substantial impact on performance. The topical model, which performed poorly for ordering, is actually stronger than the entity grid in this setting. IBM-1 underperforms either grid model (69% to 77%); on ordering, it was nearly as good (85% to 86%). Despite their ordering performance of 72%, pronouns are essentially useless for this task, at 52%.

As before, the ablation results show that all the models are quite correlated, since removing any single model from the mixture causes only a small decrease in performance. The largest drop (83% to 81%) is caused by removing time; though time is a weak model on its own, it is completely orthogonal to the other models, since unlike them, it does not depend on the words in the sentences.

Comparing results between "different topic" and "same topic" instances shows that "same topic" is harder– by about 7% for the combined model. The IBM model has a relatively small gap of 3.7%, and in the ablation results, removing it causes a larger drop in performance for "same" than "different"; this suggests it is somewhat more robust to similarity in topic than entity grids.

Disentanglement accuracy is hard to predict given ordering performance; the two tasks plainly make different demands on models. One difference is that the models which use longer histories (the two entity grids) remain strong, while the models considering only one or two previous sentences (IBM and pronouns) do not do as well. Since the changes being considered here affect only a single sentence, while permutation affects the entire transcript, more history may help by making the model more sensitive to small changes.

5.4.2 Disentangling an entire transcript

We now turn to the task of disentangling an entire transcript at once. This is a practical task, motivated by applications such as search and information retrieval. However, it is more difficult than assigning only a single utterance, because decisions are interrelated– an error on one utterance may cause a cascade of poor decisions further down. It is also computationally harder. We use tabu search (Glover and Laguna, 1997) to find a good solution.

It is also computationally more difficult. Therefore, we attempt to find a good solution using local search.

	Different	Same	Avg.
EGrid	60.3	57.1	58.7
Topical EGrid	62.3	56.8	59.6
IBM-1	56.5	55.2	55.9
Pronouns	54.5	54.4	54.4
Time	55.4	53.8	54.6
Combined	**67.9**	**59.8**	**63.9**
E+C '08	59.1	57.4	58.3

Table 5.4: One-to-one overlap between disentanglement results and truth on 200 synthetic multiparty conversations from SWBD test.

	Different	Same	Avg.
EGrid	57.2	53.7	55.4
Topical EGrid	57.3	53.3	55.3
IBM-1	58.1	56.1	57.1
Pronouns	52.2	51.8	52.0
Time	51.2	51.3	51.3
Combined	**64.6**	**58.9**	**61.8**
E+C '08	57.8	54.9	56.3

Table 5.5: loc_3 between disentanglement results and truth on 200 synthetic multiparty conversations from SWBD test.

Our search algorithm is tabu search (Glover and Laguna, 1997), which is a modified version of greedy search. We begin by randomly assigning each utterance to one of the two possible conversation threads. We then repeatedly find and move the utterance which would most improve the model score if swapped from one thread to the other. However, unlike greedy search, tabu search is also constrained not to repeat a configuration that it has recently visited. When it reaches a local maximum, it keeps exploring, moving away from the maximum, which is now considered tabu and cannot be found a second time. We run 500 iterations of tabu search (usually finding the first local maximum after about 100) and return the best solution found.

We report two of the evaluation metrics from chapter 4; one-to-one overlap (Table 5.4) maps the two clusters to the two gold dialogues, then measures how many utterances are placed in the correct dialogue. For transcripts with different topics, our disentanglement has 68% overlap with truth, extracting about two thirds of the structure correctly; this is substantially better than the model from chapter 4, which scores 59%. Where the entangled conversations have the same topic, performance is lower, about 60%, but still better than the comparison model with 57%. The loc_3 metric (Table 5.5) measures agreement in the neighborhood of a single utterance, checking whether its assignment is correct relative to the previous three utterances. This metric appears more difficult, with scores of 66% for different topics and 59% for the same, but our model is still superior to that of chapter 4. Since correlations with the previous section are fairly reliable, and the disentanglement procedure is computationally intensive, we omit ablation experiments.

As we expect, full disentanglement is substantially more difficult than single-sentence disentanglement (combined scores drop by about 20%), but the single-sentence task is a good predictor of relative model performance. Entity grid models do best, the IBM model remains useful, but less so than for discrimination, and pronouns are very weak. The IBM model performs similarly under both metrics (56% and 57%), while other models perform worse on loc_3. This supports our suggestion that IBM's decline in performance from ordering is indeed due to its using a single sentence history; it is still capable of getting local structures right, but misses global ones.

5.5 Disentangling IRC Chat

In this section, we move from synthetic data to real multiparty discourse recorded from internet chat rooms. We use two datasets: the #LINUX corpus described in section 4.2, and three larger corpora, #IPHONE, #PHYSICS and #PYTHON (Adams, 2008). We use the 1000-line "development" section of #LINUX for tuning our mixture models and the 800-line "test" section for development experiments. We reserve the Adams (2008) corpora for testing; together, they consist of 29061 lines of chat, with each file containing 500 to 1000 lines.

In order to use syntactically-based models like the entity grid, we parsed the transcripts using the self-trained Charniak parser McClosky et al. (2006). Performance was bad, although the parser does identify most of the NPs and sometimes even gets their roles correct; as reported in Foster (2010), poor results are typical when using a standard parser on chat data. We postprocessed the parse trees to change the tags of "lol", "haha" and "yes" to UH (they had been variously tagged as NN, NNP and JJ).

In this section, we use all three of our chat-specific models (time, speaker and mention) as a baseline. This baseline is relatively strong, so we evaluate our other models in combination with it.

5.5.1 Disentangling a single sentence

As before, we show results on correctly disentangling a single sentence, given the correct structure of the rest of the transcript. In each case, we average performance over each annotation of a transcript, then average the different transcripts, weighing results by transcript length so that each utterance has equal weight.

Our results for this task on our development corpus, #LINUX, are given in Table 5.6. Our best result, for the chat-specific features plus entity grid, is 79%, improving on the comparison model from chapter 4, which gets 76%. (Although the table only presents an average over all annotations of the dataset, this model is also more accurate for each individual annotator than the comparison model.) We then ran the same model, chat-specific features plus entity grid, on the test corpora from Adams (2008). These results (Table 5.7) are also better than chapter 4 at an average of 93% over 89%.

As pointed out in section 4.4, the chat-specific features are quite powerful in this domain, and it is hard to improve over them. The model from chapter 4, which has simple lexical features, mostly based on unigram overlap, increases performance over baseline by 2%. Both IBM and the topical entity grid achieve similar gains. The entity grid does better, increasing performance to 79%. Pronouns, as before for SWBD, are useless.

We believe that the entity grid's good performance here is due mostly to two factors: its use of a long history, and its lack of lexicalization. The grid looks at the previous six sentences, which differentiates it from the IBM model and from chapter 4, where we treat each pair of sentences independently. Using this long

Chat-specific	74.0
+EGrid	**79.3**
+Topical EGrid	76.8
+IBM-1	76.3
+Pronouns	73.9
E+C ‘08b	76.4
+EGrid/Topic/IBM-1	78.3

Table 5.6: Average accuracy for disentanglement of a single utterance, averaged over annotations of 800 lines of #LINUX data.

	#IPHONE	#PHYSICS	#PYTHON
E+C ‘08b	89.0	90.2	88.4
EGrid	**92.3**	**96.6**	**91.1**

Table 5.7: Average accuracy for disentanglement of a single utterance for 19581 total lines from Adams (2008).

history helps to distinguish important nouns from unimportant ones better than frequency alone. We suspect that our lexicalized models, IBM and the topical entity grid, are hampered by poor parameter settings, since their parameters were learned on FISHER rather than IRC chat. In particular, we believe this explains why the topical entity grid, which slightly outperformed the entity grid on SWBD, is much worse here.

5.5.2 Full disentanglement

Running our tabu search algorithm on the full disentanglement task yields disappointing results. Accuracies on the #LINUX dataset are not only worse than previous work, but also worse than simple baselines like creating one thread for each speaker. The model is very poor at predicting the number of threads– the dataset contains about 81 (averaging over annotations), but the model detects over 300. This appears to be related to biases in our chat-specific models as well as in the entity grid; the time model (which generates gaps between adjacent sentences) and the speaker model (which uses a CRP) both assign probability 1 to single-utterance conversations. The entity grid also has a bias toward short conversations, because unseen entities are empirically more likely to occur toward the beginning of a conversation than in the middle.

A major weakness in our model is that we aim only to maximize coherence of the individual conversations, with no prior on the likely length or number of conversations that will appear in the transcript. This allows the model to create far too many conversations. Integrating a prior into our framework is not straightforward because we currently train our mixture to maximize single-utterance disentanglement performance, and the prior is not useful for this task.

We experimented with fixing parts of the transcript to the solution obtained by the correlation clustering model, then using tabu search to fill in the gaps. This constrains the number of conversations and their approximate positions. With this structure in place, we were able to obtain scores comparable to the model from chapter 4, but not improvements; since even these scores are no longer state-of-the-art on this dataset, we regard this as a negative result. It appears that our performance increase on single-sentence disentanglement does not transfer to this task because of cascading errors and the necessity of using external constraints.

5.6 Conclusion

This chapter investigates the generalization of local coherence models. Our first finding is that entity grids and the IBM model generalize well to the conversational domain, which is encouraging because it suggests they do indeed capture universal aspects of coherence; Centering transitions and lexical cohesion seem to work well in describing the flow of dialogue as well as written language.

However, conversation appears to have different patterns of reference than writing, which poses problems for some of the models. The discourse-new model, the pronoun coreference model and the extended entity grid, all of which use information gathered from referring expressions, fail to perform well on conversation. This suggests that our understanding of how entities are introduced, grounded and referred to in discourse is insufficiently broad, and needs to be extended to cover non-written language.

Generalization from ordering to the disentanglement task is not straightforward. Models with many context sentences do well, but those without have reduced performance. Since disentanglement involves moving single utterances between conversations, the most successful models are those that are sensitive to small changes in the document structure, even if this sensitivity comes at the expense of more noisy performance on ordering. For instance, the topical entity grid, which uses a lot of noisy information, does better on disentanglement than ordering, while the pronoun model, which detects rare but clear-cut violations– pronouns with no antecedents– does worse. This implies that full task-generality may require models which use large amounts of information, even if results on the newswire discrimination task make such models seem less effective.

Finally, our results moving from phone dialogues to IRC show that while both lexical and entity-based approaches to disentanglement are viable, lexical models suffer when they are not provided with appropriate in-domain training data. This explains their failure to perform as well on IRC as they do on phone dialogues. In particular, our results on SWBD data confirm the conjecture of Adams (2008) that LDA topic modeling is in principle a useful tool for disentanglement. While they suspected this based on the use of topical information in other domains, they were unable to demonstrate it, probably because they failed to extract a good set of topics on IRC. The comparison between our phone and IRC results suggest that disentanglement would benefit substantially from improved low-level resources for internet chat, such as larger training sets, or a protocol for transfer learning.

Chapter 6

Conclusion

This thesis attempts to answer the question of which models of local coherence are both effective and general. We demonstrate several models which improve the state-of-the-art on conventional ordering tasks. Most of these extend the popular framework of entity-based coherence by adding information derived from referring expressions.

To measure generality, we motivate chat disentanglement as a coherence modeling task with a different target domain and algorithmic properties. We create and annotate a corpus for this task, measure interannotator agreement, and propose a baseline model with fairly good performance.

We apply our coherence models to disentanglement of phone conversations and IRC chat. While a few of them seem restricted to news or to permutation-based tasks, the rest transfer well to phone conversations, demonstrating that entity-based, topical and lexical systems are domain and task-general. Our results for IRC chat show this domain is very difficult for lexical systems, but that entity-based models continue to do well.

Our results suggest several wider questions, which, however, are beyond the scope of the present work. While we provide evidence that local and global coherence interact, we do not provide a scalable method for integrating the two into a single model. We also show that good disentanglement requires models of features unrelated to coherence, such as the likely duration of a conversation, and the number of conversations a speaker might participate in; however, we do not investigate how this information can be usefully incorporated into the model. Finally, although we demonstrate the generality of our methods across different English corpora, we leave open the question of extensions to other languages, where the specific syntactic and lexical patterns we detect may not exist.

From an even broader perspective, coherence is only one pragmatic constraint on the universe of possible discourse. Social concerns like register and genre impose requirements beyond mere intelligibility. For instance, a fable should be educational, a mystery suspenseful, a comedy funny and a journalistic report objective. These differing purposes lead to very different kinds of speech or writing, even when the basic facts of the narrative are the same. While these varied styles go far beyond coherence, the syntactic, lexical and rhetorical devices used to implement them are quite similar to the ones we investigate here. We believe a good approach to the problem of style must begin at coherence and work upward, possibly adapting some of the same models investigated here.

Appendix A

Topic-based model

In this appendix, we give a full explanation of the mathematics behind the integrated entity grid and HMM model presented in section 3.1.

The model assigns probabilities to documents, where a document consists of sequence of sentences S_i. The words of each sentence are divided deterministically into three sets: the set of words mentioning previously unknown entities, N_i, the set of words mentioning entities which have already been mentioned, E_i, and the set of words which do not mention entities, W_i. (As in our other models, coreference is resolved deterministically using the same-head heuristic, so this division can be done as a preprocess, using the parse trees to find mentions.)

Like Barzilay and Lee (2004), our model uses a Hidden Markov structure to assign each sentence a hidden topic and model how it evolves over time. q_i denotes the topical state of sentence S_i, which is conditioned on the previous state q_{i-1}. (We introduce a dummy start state, q_s, on which we condition the first state q_0.) Our generative process for N and W, all the words in the document except those referring to previously mentioned entities, is a non-parametric Bayesian version of Barzilay and Lee (2004); it produces these words from state-specific language models. Specifically, we have:

$$\begin{aligned} q_i &\sim DP(\cdot|q_{i-1}) \\ W_i &\sim PY(\cdot|q_i; \theta_{LM}, discount_{LM}) \\ N_i &\sim PY(\cdot|q_i; \theta_{NN}, discount_{NN}) \end{aligned}$$

Here, DP refers to the Dirichlet Process and PY to the Pitman-Yor Process (Teh, 2006). The parameters θ and $discount$ are set by hand.

For the previously mentioned entities E, however, we apply a more complex procedure, which conditions future mentions not only on the topic state, but on whether the entities were mentioned in the two previous sentences. To do so, we create a model we refer to as the *relaxed* entity grid. (As stated in the body of the thesis, this model does not perform better than the standard entity grid, as we erroneously reported in the original paper Elsner et al. (2007), except on the AIRPLANE dataset.)

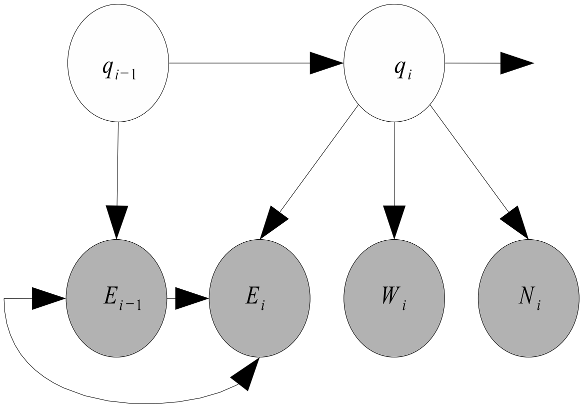

Figure A.1: The graphical model for a single time-slice of our HMM; the hidden topic state q_i is generated conditioned on the previous state q_{i-1}; it determines the words W and new entities N, while the previously-existing entities E are conditioned both on the state and directly on the previous sentences.

The relaxed grid takes as given the set of syntactic roles available to be filled by previously-mentioned entities in sentence S_i. For instance, the sentence may have slots for a subject, an object and two other NPs, and the subject may be filled with some novel entity from N_i; this leaves three slots for entities in E_i, with grid roles **O**, **X** and **X**. The relaxed grid models the probability that some entity e (drawn from the set of all previously mentioned entities, $E_0 ... E_{i-1}$) will fill each slot r, which we write $P(r \leftarrow e)$. This probability is conditioned on the type of the slot (**S**, **O**, or **X**) t_r and the standard entity grid history $\vec{r}^h_{i-1}$ (see section 2.1.1).

We estimate the probability heuristically, by counting the number of times an entity has history $\vec{r}^h_{i-1}$ and fills a slot with type t_r, and dividing by the number of times an entity has history $\vec{r}^h_{i-1}$ and a slot of type t_r is filled by any entity:

$$P_{EG}(r \leftarrow e | t_r, \vec{r}^h_{i-1}) = \frac{\#(t_r \leftarrow \vec{r}^h_{i-1})}{\#(\vec{r}^h_{i-1}; t_r \leftarrow \cdot)}$$

When we combine this model with the HMM, we can also condition on the state which we are in, and the lexical identity of the entity e. In this case, we have $P(r \leftarrow e | t_r, \vec{r}^h_{i-1}, q_i, lex(e))$. If we applied the heuristic estimator directly, we would have sparsity problems, so we instead sample the parameters from a common Dirichlet prior, which results in the conditional probability:

$$P_{LEX}(r \leftarrow e | t_r, \vec{r}^h_{i-1}, q_i, lex(e)) = \frac{\#(t_r \leftarrow \vec{r}^h_{i-1}, q_i, lex(e)) + \theta_{EG} P_{EG}(r \leftarrow e)}{\#(\vec{r}^h_{i-1}, q_i, lex(e); t_r \leftarrow \cdot) + \theta_{EG}}$$

We learn parameters by Gibbs sampling; the hidden variable of greatest interest is q, the hidden state. (There are also hidden variables associated with the hierarchical Dirichlet and Pitman-Yor processes for the words. The sampling for these is identical to Teh (2006).) We need:

$$P(q_i | q_{-i}, E_i, W_i, N_i; \theta, discount)$$

(The notation q_{-i} means the states for all sentences other than i.)

This conditional distribution is the product of several terms. There are emission terms associated with each of E_i,W_i,N_i, plus a transition term dependent on q_{i-1} and q_{i+1}.

$$
\begin{aligned}
P(q_i = q|E_i, W_i, N_i) \propto & \prod_{r \in S_i, e \in E_i} P_{LEX}(r \leftarrow e|t_r, \vec{r}^{h}_{i-1}, q_i, lex(e)) \\
& \prod_{w \in W_i} P_{LM}(w|q) \\
& \prod_{n \in N_i} P_{NN}(n|q) \\
& P_{trans}(q_i = q|q_{i-1}) \times P_{trans}(q_{i+1}|q_i = q)
\end{aligned}
$$

The parameters for P_{LM} and P_{NN} are sampled from Pitman-Yor processes and integrated out. We use a Chinese Restaurant representation; supposing it has a set of tables T_q, with w_t being the label (word) for a particular table, and n_t its number of customers, we have the following probability for a word to be emitted from state q:

$$
\begin{aligned}
P_{LM}(w|q) = & \frac{\sum_{t \in T_q : w_t = w}(n_t - discount)}{\sum_{t \in T_q} n_t + \theta} \\
& + \frac{\theta + discount \times |T_q|}{\sum_{t \in T_q} n_t + \theta} P_{LM0}(w)
\end{aligned}
$$

The formula for P_{LM0}, the prior over unigrams, is similar:

$$
\begin{aligned}
P_{LM0}(w) = & \frac{\sum_{t \in T : w_t = w}(n_t - discount)}{\sum_{t \in T} n_t + \theta} \\
& + \frac{\theta + discount \times |T|}{\sum_{t \in T} n_t + \theta} P_{G0}(w)
\end{aligned}
$$

Finally, $P_{G0}(w)$ is uniform over a fixed vocabulary.

The parameters of P_{trans} are sampled from Dirichlet processes, which are equivalent to the Pitman-Yor process with $discount = 0$. Thus, the conditional distribution $P_{trans}(q_i = q|q_{i-1})$ with the parameters integrated out is:

$$
P_{trans}(q_i = q|q_{i-1} = q') = \frac{n_{q_j = q' \wedge q_{j+1} = q}}{n_{q_j = q'} + \theta} + \frac{\theta}{n_{q_j = q'} + \theta} P_{trans0}(q)
$$

$P_{trans0}(q)$ is a Chinese Restaurant Process, with each table representing a different state.

The remaining term of the conditional, P_{LEX}, is explained above.

For inference, as stated in the main text, we fix the number of states and use the sum-product algorithm to compute document probabilities.

Appendix B

Multiply-orderable texts

While presenting research on coherence models and ordering evaluations (discussed in chapter 2), I have occasionally been asked whether it is always the case that permuted versions of a text are less coherent than the original, and whether small perturbations of the original ordering are always less disruptive than large ones. For short texts, this appears to be true (Lapata, 2006). For longer ones, of course, permutations that move whole paragraphs or sections are likely to be better than those that disrupt them (Bollegala et al., 2006), a criticism we also point out while comparing local and global ordering models in section 3.1. In general, however, texts do not remain coherent after wholesale, sentence-by-sentence reorderings such as reversals.

One way to demonstrate that this is the case is to present a few purposefully-written texts which *are* coherent after a large-scale reordering. These are not naturally occurring texts, but carefully planned language games. Their multiple coherent orderings are created by ambiguities which are resolved differently depending on context, plus occasionally the conventions of poetic form. Considering the high degree of precision and planning they exhibit, it should be clear that this property will not occur by chance in normal discourse.

B.1 Tiff and I

Tiff and I sit
in Tompkins Square Park
reading poetry
under a sky
full of clapping pigeons.
He calls them flying rats
but I think
the pink and green circles
around their necks
like greasy oil puddles are
beautiful.

Tiff says
all my poems sound better
backwards.

Backwards
all my poems sound better
Tiff says.
Beautiful
like greasy oil puddles are
around their necks
the pink and green circles
but I think
he calls them flying rats.
Full of clapping pigeons
under a sky
reading poetry
in Tompkins Square Park
Tiff and I sit.

–Leslea Newman (“Sweet Dark Places”, quoted in “Contemporary Linguistics: An Introduction”, 5th ed., O’Grady et al)

B.2 Once Upon a Time, There Was ...

1. A child of a man and the principle of fate, and
3. deciding, “In my life, I shall only experience every other event,”
5. He too slept with a principle of fate.
7. And he had a son named Rival.
9. He saw that the child was beautiful, and loved him, but
11. A servant took Rival in the dead of night and carried him away.
13. In a distant place, unknowing of his father, Rival grew to a youth.
15. He learned the art of the discus.
17. He found himself in his father’s city on the day of a discus match.
19. And his father looked down from the stands. And quite by accident,
21. The father’s eyes met his son’s. “You,” he said. “You are my blood.”
23. And Rival became his heir,
25. And if time is not circular, then here the story ends.

**

2. If time is circular, he took the name Successor,
4. redefining time so that every other event flowed in sequence.
6. Fate grew pregnant with Successor’s child

8. And fate prophesied that this child, his Rival, would kill him.

10. Successor ordered his son slain, so his son fled–

12. Weeping bitter tears.

14. In a distant place, where his father could not find him, his Rival grew into a man.

16. When he was strong, his Rival set out in search of Successor’s home.

18. He took up a great stone disc.

20. He hurled it at his father’s head.

22. And the stone struck Successor dead.

24. And if time is circular, then the one man is the other–

– Jenna K. Moran, (“Hitherby Dragons”, 10 November 2004, `imago.hitherby.com/?p=302`)

B.3 from “Crab Canon”

Achilles I don’t know. But one thing for certain is that I don’t worry about arguments of taste. *De gustibus non disputandum.*

Tortoise Tell me, what’s it like to be your age? Is it true that one has no worries at all?

Achilles To be precise, one has no frets.

Tortoise Oh, well, it’s all the same to me.

Achilles Fiddle. It makes a big difference, you know.

Tortoise Say, don’t you play the guitar?

Achilles That’s my good friend. He often plays, the fool. But I myself wouldn’t touch a guitar with a ten-foot pole!

(Suddenly, the Crab, appearing out of nowhere, wanders up excitedly, pointing to a rather prominent black eye.)

Crab Hallo! Hullo! What’s up? What’s new? You see this bump, this from Warsaw - a colossal bear of a man - playing a lute. He was three meters tall, if I’m a day. I mosey on up to the chap, reach skyward and manage to tap him on the knee, saying, “Pardon me, sir, but you are Pole-luting our park with your mazurkas.” But WOW! he had no sense of humor - not a bit, not a whit - and POW! - he lets loose and belts me one, smack in the eye! Were it in my nature, I would crab up a storm, but in the time-honored tradition of my species, I backed off. After all, when we walk forwards, we move backwards. It’s in our genes, you know, turning round and round. That reminds me - I’ve always wondered, “which came first - the Crab or the Gene?” That is to say, “Which came last - the Gene, or the Crab?” I’m always turning things round and round, you know. It’s in our genes, after all. When we walk backwards we move forwards. Ah me, oh my! I must lope along on my merry way - so off I go on such a fine day. Sing “ho!” for the life of a Crab! TATA! Ole!

(And he disappears as suddenly as he arrived.)

Tortoise That’s my good friend. He often plays the fool. But I myself wouldn’t touch a ten-foot Pole with a guitar.

Achilles Say, don’t you play the guitar?

Tortoise Fiddle. It makes a big difference, you know.

Achilles Oh, well, it’s all the same to me.

Tortoise To be precise, one has no frets.

Achilles Tell me, what’s it like to be your age? Is it true that one has no worries at all?

Tortoise I don’t know. But one thing for certain is that I don’t worry about arguments of taste. *Disputandum non est de gustibus.*

– Douglas Hofstadter, (“Gödel, Escher, Bach: an Eternal Golden Braid”, 2nd edition, Basic Books, 1999)

Bibliography

Evrim Acar, Seyit Ahmet Camtepe, Mukkai S. Krishnamoorthy, and Blent Yener. 2005. Modeling and multiway analysis of chatroom tensors. In Paul B. Kantor, Gheorghe Muresan, Fred Roberts, Daniel Dajun Zeng, Fei-Yue Wang, Hsinchun Chen, and Ralph C. Merkle, editors, *ISI*, volume 3495 of *Lecture Notes in Computer Science*, pages 256–268. Springer.

Paige H. Adams and Craig H. Martell. 2008. Topic detection and extraction in chat. *International Conference on Semantic Computing*, 0:581–588.

Paige H. Adams. 2008. *Conversation Thread Extraction and Topic Detection in Text-based Chat*. Ph.D. thesis, Naval Postgraduate School.

Nir Ailon, Moses Charikar, and Alantha Newman. 2008. Aggregating inconsistent information: Ranking and clustering. *Journal of the ACM*, 55(5):Article No. 23.

David Aldous. 1985. Exchangeability and related topics. In *Ecole d'Ete de Probabilities de Saint-Flour XIII 1983*, pages 1–198. Springer.

Ernst Althaus, Nikiforos Karamanis, and Alexander Koller. 2004. Computing locally coherent discourses. In *Proceedings of the 42nd ACL*, Barcelona.

Galen Andrew and Jianfeng Gao. 2007. Scalable training of L1-regularized log-linear models. In *ICML '07*.

Paul M. Aoki, Matthew Romaine, Margaret H. Szymanski, James D. Thornton, Daniel Wilson, and Allison Woodruff. 2003. The mad hatter's cocktail party: a social mobile audio space supporting multiple simultaneous conversations. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 425–432, New York, NY, USA. ACM Press.

Paul M. Aoki, Margaret H. Szymanski, Luke D. Plurkowski, James D. Thornton, Allison Woodruff, and Weilie Yi. 2006. Where's the "party" in "multi-party"?: analyzing the structure of small-group sociable talk. In *CSCW '06: Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, pages 393–402, New York, NY, USA. ACM Press.

Nikhil Bansal, Avrim Blum, and Shuchi Chawla. 2004. Correlation clustering. *Machine Learning*, 56(1-3):89–113.

Regina Barzilay and Mirella Lapata. 2005. Modeling local coherence: an entity-based approach. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05).*

Regina Barzilay and Mirella Lapata. 2006. Aggregation via set partitioning for natural language generation. In *HLT-NAACL*.

Regina Barzilay and Lillian Lee. 2004. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *HLT-NAACL 2004: Proceedings of the Main Conference*, pages 113–120.

Regina Barzilay, Noemie Elhadad, and Kathleen McKeown. 2002. Inferring strategies for sentence ordering in multidocument news summarization. *Journal of Artificial Intelligence Results (JAIR)*, 17:35–55.

Matthew J. Beal, Zoubin Ghahramani, and Carl Edward Rasmussen. 2001. The infinite Hidden Markov Model. In *NIPS*, pages 577–584.

David L. Bean and Ellen Riloff. 1999. Corpus-based identification of non-anaphoric noun phrases. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics (ACL'99)*, pages 373–380, Morristown, NJ, USA. Association for Computational Linguistics.

Doug Beeferman, Adam L. Berger, and John D. Lafferty. 1999. Statistical models for text segmentation. *Machine Learning*, 34(1-3):177–210.

Amir Ben-Dor, Ron Shamir, and Zohar Yakhini. 1999. Clustering gene expression patterns. *Journal of Computational Biology*, 6(3-4):281–297.

Michael Bertolacci and Anthony Wirth. 2007. Are approximation algorithms for consensus clustering worthwhile? In *SDM '07: Procs. 7^{th} SIAM International Conference on Data Mining*.

David M. Blei and Pedro J. Moreno. 2001. Topic segmentation with an aspect hidden markov model. In *SIGIR*, pages 343–348.

David Blei, Andrew Y. Ng, and Michael I. Jordan. 2001. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:2003.

Danushka Bollegala, Naoaki Okazaki, and Mitsuru Ishizuka. 2006. A bottom-up approach to sentence ordering for multi-document summarization. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 385–392, Sydney, Australia, July. Association for Computational Linguistics.

P. Brown, S. Della Pietra, V. Della Pietra, and R. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2).

Jill Burstein, Joel Tetreault, and Slava Andreyev. 2010. Using entity-based features to model coherence in student essays. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 681–684, Los Angeles, California, June. Association for Computational Linguistics.

Seyit Ahmet Camtepe, Mark K. Goldberg, Malik Magdon-Ismail, and Mukkai Krishnamoorty. 2005. Detecting conversing groups of chatters: a model, algorithms, and tests. In *IADIS AC*, pages 89–96.

Moses Charikar, Venkatesan Guruswami, and Anthony Wirth. 2005. Clustering with qualitative information. *J. Comput. Syst. Sci.*, 71(3):360–383.

Eugene Charniak and Micha Elsner. 2009. EM works for pronoun anaphora resolution. In *Proceedings of EACL*, Athens, Greece.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proc. of the 2005 Meeting of the Assoc. for Computational Linguistics (ACL)*, pages 173–180.

Lei Chen, Mary Harper, Amy Franklin, Travis R. Rose, Irene Kimbara, Zhongqiang Huang, and Francis Quek. 2006. A multimodal analysis of floor control in meetings. In *In Proc. of MLMI 06*.

Erdong Chen, Benjamin Snyder, and Regina Barzilay. 2007. Incremental text structuring with online hierarchical ranking. In *Proceedings of EMNLP*.

Harr Chen, S.R.K. Branavan, Regina Barzilay, and David R. Karger. 2009. Global models of document structure using latent permutations. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 371–379, Boulder, Colorado, June. Association for Computational Linguistics.

Lei Chen. 2008. *Incorporating Nonverbal Features into Multimodal Models of Human-to-Human Communication*. Ph.D. thesis, Purdue University.

Jackie Chi Kit Cheung and Gerald Penn. 2010. Entity-based local coherence modelling using topological fields. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 186–195, Uppsala, Sweden, July. Association for Computational Linguistics.

William W. Cohen and Jacob Richman. 2002. Learning to match and cluster large high-dimensional data sets for data integration. In *KDD '02*, pages 475–480. ACM.

Hal Daumé III. 2004. Notes on CG and LM-BFGS optimization of logistic regressio n. Paper available at http://pub.hal3.name#daume04cg-bfgs, implementation available at http://hal3.name/megam/, August.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407.

Erik D. Demaine, Dotan Emanuel, Amos Fiat, and Nicole Immorlica. 2006. Correlation clustering in general weighted graphs. *Theor. Comput. Sci.*, 361(2):172–187.

Robert L. Donaway, Kevin W. Drummey, and Laura A. Mather. 2000. A comparison of rankings produced by summarization evaluation measures. In *NAACL-ANLP 2000 Workshop on Automatic Summarization*, pages 69–78, Morristown, NJ, USA. Association for Computational Linguistics.

Jacob Eisenstein and Regina Barzilay. 2008. Bayesian unsupervised topic segmentation. In *EMNLP*, pages 334–343.

Jacob Eisenstein. 2009. Hierarchical text segmentation from multi-scale lexical cohesion. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 353–361, Boulder, Colorado, June. Association for Computational Linguistics.

Micha Elsner and Eugene Charniak. 2008a. Coreference-inspired coherence modeling. In *Proceedings of ACL-08: HLT, Short Papers*, pages 41–44, Columbus, Ohio, June. Association for Computational Linguistics.

Micha Elsner and Eugene Charniak. 2008b. You talking to me? a corpus and algorithm for conversation disentanglement. In *Proceedings of ACL-08: HLT*, pages 834–842, Columbus, Ohio, June. Association for Computational Linguistics.

Micha Elsner and Eugene Charniak. 2010a. Disentangling chat. *Computational Linguistics*, 36.

Micha Elsner and Eugene Charniak. 2010b. The same-head heuristic for coreference. In *Proceedings of ACL 10*, Uppsala, Sweden, July. Association for Computational Linguistics.

Micha Elsner and Warren Schudy. 2009. Bounding and comparing methods for correlation clustering beyond ilp. In *Proceedings of the NAACL/HLT 2009 Workshop on Integer Linear Programming for Natural Language Processing (ILP-NLP '09)*, Boulder, Colorado, June.

Micha Elsner, Joseph Austerweil, and Eugene Charniak. 2007. A unified local and global model for discourse coherence. In *Proceedings of HLT-NAACL '07*.

Katja Filippova and Michael Strube. 2007. Extending the entity-grid coherence model to semantically related entities. In *Proceedings of the Eleventh European Workshop on Natural Language Generation*, pages 139–142, Saarbrücken, Germany, June. DFKI GmbH. Document D-07-01.

Jenny Rose Finkel and Christopher D. Manning. 2008. Enforcing transitivity in coreference resolution. In *Proceedings of ACL-08: HLT, Short Papers*, pages 45–48, Columbus, Ohio, June. Association for Computational Linguistics.

Peter Foltz, Walter Kintsch, and Thomas Landauer. 1998. The measurement of textual coherence with latent semantic analysis. *Discourse Processes*, 25(2&3):285–307.

Jennifer Foster. 2010. "cba to check the spelling": Investigating parser performance on discussion forum posts. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 381–384, Los Angeles, California, June. Association for Computational Linguistics.

Kari Fraurud. 1990. Definiteness and the processing of noun phrases in natural discourse. *Journal of Semantics*, 7(4):395–433.

Michel Galley, Kathleen McKeown, Eric Fosler-Lussier, and Hongyan Jing. 2003. Discourse segmentation of multi-party conversation. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 562–569, Morristown, NJ, USA. Association for Computational Linguistics.

Niyu Ge, John Hale, and Eugene Charniak. 1998. A statistical approach to anaphora resolution. In *Proceedings of the Sixth Workshop on Very Large Corpora*, pages 161–171, Orlando, Florida. Harcourt Brace.

Aristides Gionis, Heikki Mannila, and Panayiotis Tsaparas. 2007. Clustering aggregation. *ACM Trans. on Knowledge Discovery from Data*, 1(1):Article 4.

Ioannis Giotis and Venkatesan Guruswami. 2006. Correlation clustering with a fixed number of clusters. *Theory of Computing*, 2(1):249–266.

Fred Glover and Manuel Laguna. 1997. *Tabu Search*. University of Colorado at Boulder.

Andrey Goder and Vladimir Filkov. 2008. Consensus clustering algorithms: Comparison and refinement. In *ALENEX '08: Procs. 10th Workshop on Algorithm Enginering and Experiments*, pages 109–117. SIAM.

David Graff. 1995. *North American News Text Corpus*. Linguistic Data Consortium. LDC95T21.

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

Aria Haghighi and Dan Klein. 2006. Prototype-driven learning for sequence models. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 320–327, New York City, USA, June. Association for Computational Linguistics.

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

Michael Halliday and Ruqaiya Hasan. 1976. *Cohesion in English*. Longman, London.

Timothy Hawes, Jimmy Lin, and Philip Resnik. 2008. Elements of a computational model for multi-party discourse: The turn-taking behavior of supreme court justices. Technical Report LAMP-TR-147/HCIL-2008-02, University of Maryland, College Park, January.

John A. Hawkins. 1978. *Definiteness and indefiniteness: a study in reference and grammaticality prediction*. Croom Helm Ltd.

Simon Haykin and Zhe Chen. 2005. The Cocktail Party Problem. *Neural Computation*, 17(9):1875–1902.

Marti A. Hearst. 1997. TextTiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64.

Cristoph Helmberg. 2000. Semidefinite programming for combinatorial optimization. Technical Report ZR-00-34, Konrad-Zuse-Zentrum für Informationstechnik Berlin.

Cristoph Helmberg, 2009. *The ConicBundle Library for Convex Optimization.* Ver. 0.2i from `http://www-user.tu-chemnitz.de /~helmberg/ConicBundle/`.

Derrick Higgins, Jill Burstein, Daniel Marcu, and Claudia Gentile. 2004. Evaluating multiple aspects of coherence in student essays. In *HLT-NAACL*, pages 185–192.

Julia Hirschberg and Diane J. Litman. 1993. Empirical studies on the disambiguation of cue phrases. *Computational Linguistics*, 19(3):501–530.

Graeme Hirst and David St-Onge, 1998. *WordNet: An electronic lexical database*, chapter Lexical chains as representations of context for the detection and correction of malapropisms, pages 305–332. The MIT Press, Cambridge, MA.

Jerry R. Hobbs. 1976. Pronoun resolution. Technical Report 76-1, City College New York.

Ilog, Inc. 2003. Cplex solver.

Hongyan Jing and Kathleen McKeown. 1999. The decomposition of human-written summary sentences. In *SIGIR*, pages 129–136.

Thorsten Joachims and John Hopcroft. 2005. Error bounds for correlation clustering. In *ICML '05*, pages 385–392, New York, NY, USA. ACM.

Thorsten Joachims. 1997. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In *International Conference on Machine Learning (ICML)*, pages 143–151.

Karen Spärck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21.

Natasa Jovanovic and Rieks op den Akker. 2004. Towards automatic addressee identification in multi-party dialogues. In Michael Strube and Candy Sidner, editors, *Proceedings of the 5th SIGdial Workshop*, pages 89–92, Cambridge, Massachusetts, USA, April 30 - May 1. Association for Computational Linguistics.

Natasa Jovanovic, Rieks op den Akker, and Anton Nijholt. 2006. Addressee identification in face-to-face meetings. In *EACL*. The Association for Computer Linguistics.

Nikiforos Karamanis, Massimo Poesio, Chris Mellish, and Jon Oberlander. 2004. Evaluating centering-based metrics of coherence. In *ACL*, pages 391–398.

Nikiforos Karamanis, Chris Mellish, Massimo Poesio, and Jon Oberlander. 2009. Evaluating centering for information ordering using corpora. *Computational Linguistics*, 35(1):29–46.

Mirella Lapata and Regina Barzilay. 2005. Automatic evaluation of text coherence: Models and representations. In *IJCAI*, pages 1085–1090.

Mirella Lapata. 2003. Probabilistic text structuring: Experiments with sentence ordering. In *Proceedings of the annual meeting of ACL, 2003.*

Mirella Lapata. 2006. Automatic evaluation of information ordering: Kendall's tau. *Computational Linguistics*, 32(4):1–14.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of HLT-EMNLP*, pages 25–32, Morristown, NJ, USA. Association for Computational Linguistics.

Igor Malioutov and Regina Barzilay. 2006. Minimum cut model for spoken lecture segmentation. In *ACL*. The Association for Computer Linguistics.

Gideon Mann, Ryan McDonald, Mehryar Mohri, Nathan Silberman, and Dan Walker. 2009. Efficient large-scale distributed training of conditional maximum entropy models. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1231–1239.

Daniel Marcu. 1997. From local to global coherence: A bottom-up approach to text planning. In *AAAI/IAAI*, pages 629–635.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Comp. Linguistics*, 19(2):313–330.

Claire Mathieu and Warren Schudy. 2008. Correlation clustering with noisy input. Unpublished manuscript available from http://www.cs.brown.edu/∼ws/papers/clustering.pdf.

Andrew McCallum and Ben Wellner. 2004. Conditional models of identity uncertainty with application to noun coreference. In *Proceedings of the 18th Annual Conference on Neural Information Processing Systems (NIPS)*, pages 905–912. MIT Press.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 152–159.

David McClosky, Eugene Charniak, and Mark Johnson. 2010. Automatic domain adaptation for parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 28–36, Los Angeles, California, June. Association for Computational Linguistics.

Marina Meila. 2007. Comparing clusterings–an information based distance. *Journal of Multivariate Analysis*, 98(5):873–895, May.

Chris Mellish, Alistair Knott, and Jon Oberl. 1998. Experiments using stochastic search for text planning. In *Proceedings of International Conference on Natural Language Generation*, pages 98–107.

G. Miller, A.R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. 1993. Introduction to wordnet: an on-line lexical database. Technical report, Princeton University.

Eleni Miltsakaki and K. Kukich. 2004. Evaluation of text coherence for electronic essay scoring systems. *Nat. Lang. Eng.*, 10(1):25–55.

Neville Moray. 1959. Attention in dichotic listening: Affective cues and the influence of instructions. *Quarterly Journal of Experimental Psychology*, 11(1):56–60.

Jane Morris and Graeme Hirst. 1991. Lexical Cohesion Computed by Thesaural Relations as an Indicator of the Structure of Text. *Computational Linguistics*, 17(1):21–48, March.

Thomas Morton, Joern Kottmann, Jason Baldridge, and Gann Bierner. 2005. Opennlp: A java-based nlp toolkit. http://opennlp.sourceforge.net.

Ani Nenkova and Kathleen McKeown. 2003. References to named entities: a corpus study. In *NAACL '03*, pages 70–72.

Ani Nenkova. 2006. *Understanding the process of multi-document summarization: content selection, rewrite and evaluation.* Ph.D. thesis, Columbia University.

Vincent Ng and Claire Cardie. 2002a. Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution. In *COLING*.

Vincent Ng and Claire Cardie. 2002b. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 104–111.

Malvina Nissim. 2006. Learning information status of discourse entities. In *Proceedings of EMNLP*, pages 94–102, Morristown, NJ, USA. Association for Computational Linguistics.

Jacki O'Neill and David Martin. 2003. Text chat in action. In *GROUP '03: Proceedings of the 2003 international ACM SIGGROUP conference on Supporting group work*, pages 40–49, New York, NY, USA. ACM Press.

Emily Pitler and Ani Nenkova. 2008. Revisiting readability: A unified framework for predicting text quality. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 186–195, Honolulu, Hawaii, October. Association for Computational Linguistics.

Emily Pitler, Mridhula Raghupathy, Hena Mehta, Ani Nenkova, Alan Lee, and Aravind Joshi. 2008. Easily identifiable discourse relations. In *Coling 2008: Companion volume: Posters*, pages 87–90, Manchester, UK, August. Coling 2008 Organizing Committee.

Massimo Poesio and Renata Vieira. 1998. A corpus-based investigation of definite description use. *Computational Linguistics*, 24(2):183–216.

Massimo Poesio, Mijail Alexandrov-Kabadjov, Renata Vieira, Rodrigo Goulart, and Olga Uryupina. 2005. Does discourse-new detection help definite description resolution? In *Proceedings of the Sixth International Workshop on Computational Semantics*, Tillburg.

Ellen Prince. 1981. Toward a taxonomy of given-new information. In Peter Cole, editor, *Radical Pragmatics*, pages 223–255. Academic Press, New York.

Matthew Purver, Konrad Körding, Thomas Griffiths, and Joshua Tenenbaum. 2006. Unsupervised topic modelling for multi-party spoken discourse. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL)*, pages 17–24, Sydney, Australia, July. Association for Computational Linguistics.

William M. Rand. 1971. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850.

Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Proceedings of CoNLL-2004*, pages 1–8. Boston, MA, USA.

Harvey Sacks, Emanuel A. Schegloff, and Gail Jefferson. 1974. A simplest systematics for the organization of turn-taking for conversation. *Language*, 50(4):696–735.

Dou Shen, Qiang Yang, Jian-Tao Sun, and Zheng Chen. 2006. Thread detection in dynamic text message streams. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 35–42, New York, NY, USA. ACM.

James Simpson. 2005. Conversational floors in synchronous text-based CMC discourse. *Discourse Studies*, 7:337–361.

Jonas Sjöbergh. 2007. Older versions of the ROUGEeval summarization evaluation system were easier to fool. *Information Processing and Management*, 43:1500–1505, November.

Marc Smith, J. J. Cadiz, and Byron Burkhalter. 2000. Conversation trees and threaded chats. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, CSCW '00, pages 97–105, New York, NY, USA. ACM.

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

Radu Soricut and Daniel Marcu. 2006. Discourse generation using utility-trained coherence models. In *Proceedings of the Association for Computational Linguistics Conference (ACL-2006)*.

Caroline Sporleder and Alex Lascarides. 2008. Using automatically labelled examples to classify rhetorical relations: An assessment. *Natural Language Engineering (JNLE)*, 14:369–416, July.

Caroline Sporleder and Linlin Li. 2009. Unsupervised recognition of literal and non-literal use of idiomatic expressions. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 754–762, Athens, Greece, March. Association for Computational Linguistics.

Y.W. Teh. 2006. A Bayesian interpretation of interpolated Kneser-Ney. Technical Report TRA2/06, National University of Singapore.

David R. Traum, Susan Robinson, and Jens Stephan. 2004. Evaluation of multi-party virtual reality dialogue interaction. In *In Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC*.

David Traum. 2004. Issues in multi-party dialogues. In F Dignum, editor, *Advances in agent communication*, pages 201–211. Springer Verlag Lecture Notes in AI 2922.

Olga Uryupina. 2003. High-precision identification of discourse new and unique noun phrases. In *Proceedings of the ACL Student Workshop*, Sapporo.

David Vadas and James Curran. 2007. Adding noun phrase structure to the Penn treebank. In *Proceedings of ACL*, pages 240–247, Prague, Czech Republic, June. Association for Computational Linguistics.

Renata Vieira and Massimo Poesio. 2000. An empirically-based system for processing definite descriptions. *Computational Linguistics*, 26(4):539–593.

Lidan Wang and Douglas W. Oard. 2009. Context-based message expansion for disentanglement of interleaved text conversations. In *Proceedings of NAACL-09*.

Yi-Chia Wang, Mahesh Joshi, William Cohen, and Carolyn Rosé. 2008. Recovering implicit thread structure in newsgroup style conversations. In *Proceedings of the 2nd International Conference on Weblogs and Social Media (ICWSM II)*.

Jen-Yuan Yeh and Aaron Harnly. 2006. Email thread reassembly using similarity matching. In *Conference on Email and Anti-Spam*.

Shi Zhong and Joydeep Ghosh. 2003. Model-based clustering with soft balancing. In *ICDM '03: Proceedings of the Third IEEE International Conference on Data Mining*, page 459, Washington, DC, USA. IEEE Computer Society.