Johns Hopkins LVCSR Workshop-97 Switchboard Discourse Language Modeling Project Final Report

Daniel Jurafsky *(University of Colorado)*, Rebecca Bates *(Boston University)*,
Noah Coccaro *(University of Colorado)*, Rachel Martin *(Johns Hopkins University)*, Marie Meteer *(BBN)*,
Klaus Ries *(CMU/Universität Karlsruhe)*, Elizabeth Shriberg *(SRI)*, Andreas Stolcke *(SRI)*,
Paul Taylor *(University of Edinburgh)*, Carol Van Ess-Dykema *(DoD)*

January 23, 1998

Abstract

We describe a new approach for statistical modeling and detection of discourse structure for natural conversational speech. Our model is based on 42 'Dialog Acts' (DAs), (question, answer, backchannel, agreement, disagreement, apology, etc). We labeled 1155 conversations from the Switchboard (SWBD) database (Godfrey *et al.* 1992) of human-to-human telephone conversations with these 42 types and trained a Dialog Act detector based on three distinct knowledge sources: sequences of words which characterize a dialog act, prosodic features which characterize a dialog act, and a statistical Discourse Grammar. Our combined detector, although still in preliminary stages, already achieves a 65% Dialog Act detection rate based on acoustic waveforms, and 72% accuracy based on word transcripts. Using this detector to switch among the 42 dialog-act-specific trigram LMs also gave us an encouraging but not statistically significant reduction in SWBD word error.

1 Introduction

The ability to model and automatically detect discourse structure is essential as we address problems like *understanding spontaneous dialog* (a meeting summarizer needs to know who said what to whom), *building human-computer dialog systems* (a conversational agent needs to know whether it just got asked a question or ordered to do something), and simply *transcription of conversational speech* (utterances with different discourse function also have very different words). This paper describes our preliminary work (as part of the 1997 Summer Workshop on Innovative Techniques in LVCSR) on automatically detecting discourse structure for speech recognition and understanding tasks. (See Jurafsky *et al.* (1997a), Shriberg *et al.* (1998), and Stolcke *et al.* (1998) for other publications describing aspects of this work).

Table 1 shows a sample of the kind of discourse structure we are modeling and detecting. Besides the usefulness of discourse structure detection for speech understanding, discourse structure can be directly relevant for speech recognition tasks. For example in the state-of-the-art HTK recognizer we used, the word **do** has an error rate of 72%. But **do** is in almost every **Yes-No-Question**; if we could detect **Yes-No-Question**s (for example by looking for utterances with rising intonation) we could increase the prior probability of **do** and hence decrease the error rate.

There are many excellent previous attempts to build predictive, stochastic models of dialog structure (Kita *et al.* 1996; Mast *et al.* 1996; Nagata and Morimoto 1994; Reithinger *et al.* 1996; Suhm and Waibel 1994; Taylor *et al.* 1998; Woszczyna and Waibel 1994; Yamaoka and Iida 1991), and our effort is in many ways inspired by this work, and indeed our group overlaps in personnel with some of these projects. Our project extends these earlier efforts particularly in its scale; our models were trained on 1155 dialog-annotated conversations comprising 205,000 utterances and 1.4 million words; an order of magnitude larger than any previous system. The direction of our project is also slightly different than most previous discourse structure recognition projects, which are based on short task-oriented dialogs, particularly from the Verbmobil domain. Our focus is on the longer, more spontaneous, less task-oriented dialogs that we found in the Switchboard database and which we expect to find in the CallHome dataset.

Table 1: *A fragment of a labeled switchboard conversation.*

Spkr	Dialog Act	Utterance
A	**Wh-Question**	What kind do you have now?
B	**Statement**	*Uh, we have a, a Mazda nine twenty nine and a Ford Crown Victoria and a little two seater CRX.*
A	**Acknowledge-Answer**	Oh, okay.
B	**Opinion**	*Uh, it's rather difficult to, to project what kind of, uh, -*
A	**Statement**	we'd, look, always look into, uh, consumer reports to see what kind of, uh, report, or, uh, repair records that the various cars have –
B	**Turn-Exit**	*So, uh, -*
A	**Yes-No-Quest**	And did you find that you like the foreign cars better than the domestic?
B	**Answer-Yes**	*Uh, yeah,*
B	**Statement**	*We've been extremely pleased with our Mazdas.*
A	**Backchannel-Quest**	Oh, really?
B	**Answer-Yes**	*Yeah.*

2 Summer Workshop Logistics, Datasets, Plan of Work, and Publications

There are a number of ways we could describe our work; describing the work in the order in which we performed it would mean dividing the work into **training**, **test**, and **analysis**. But for the reader, it is probably more coherent to read about the work in the order that best explains the significance and applicability of our final results. We have chosen the second method for the organization of the paper. Figure 1 outlines the major stages of our work and of this paper. We begin by discussing how we manually annotated 1155 conversations with hand-labeled discourse-tags. We then describe the 3 knowledge sources for dialog act detection (word-sequences, discourse grammar, and prosody), show how these knowledge sources can be combined to build a Dialog Act detector, and finally how to apply the detector to help improve word recognition of SWBD.

But to give an idea of the logistic organization of the project, we also include here some charts deriving from our summer Plan of Work in Figures 2, 3, and 4.

Finally, we introduce the data we used for all our experiments. As is usual in ASR experiments, we divided our data into 3 portions: Training (WS97-TRN), Development-Test (WS97-DEV), and Evaluation-Test (WS97-EVAL). As is usual in the LVCSR Summer Workshops, we never used our Evaluation-Test (WS97-EVAL) data, or the extra Eval Set (WS97-EVAL2). All the experimental results we report are on the DevTest data. The Venn diagram in Figure 5 shows the relationship among these.

For most experiments we trained on DI97-TRN + DI-97-HLD (which together comprise the complete WS97 training set) and tested by rescoring lattices which were generated on the dev-test set WS97-DEV. For perplexity experiments, where lattices were unnecessary, we trained on DI97-TRN and tested on DI97-HLD.

3 The Tag Set and the Manual Tagging Effort

In order to use discourse knowledge to enrich our LMs, we must choose a level at which to model discourse knowledge. For example, at the level of plans and intentions, we could describe a conversation in terms of the high-level goals and plans of the participants (Perrault and Allen 1980; Litman and Allen 1987). At the level of focus, we could describe a conversation in terms of local attentional centers or foci (Grosz *et al.* 1995; Walker and Prince 1993). We might call these intentional or attentional models DEEP DISCOURSE STRUCTURE. At the level of speech acts, we can model the speech act type of each utterance (Searle (1969) and computational versions of speech-act-like units (Nagata and Morimoto 1994; Reithinger *et al.* 1996; Carletta *et al.* 1997)). Or we can model sociolinguistic facts about conversation structure such as how participants might expect one type of conversational units to be responded to by another (the **adjacency pairs** of Schegloff (1968) or Sacks *et al.* (1974)). We refer to these latter two types of discourse structure as SHALLOW DISCOURSE STRUCTURE.

We chose to follow a recent standard for shallow discourse structure markup, the Discourse Annotation and Markup System of Labeling (DAMSL) tag-set, which was recently designed by the natural-language processing community

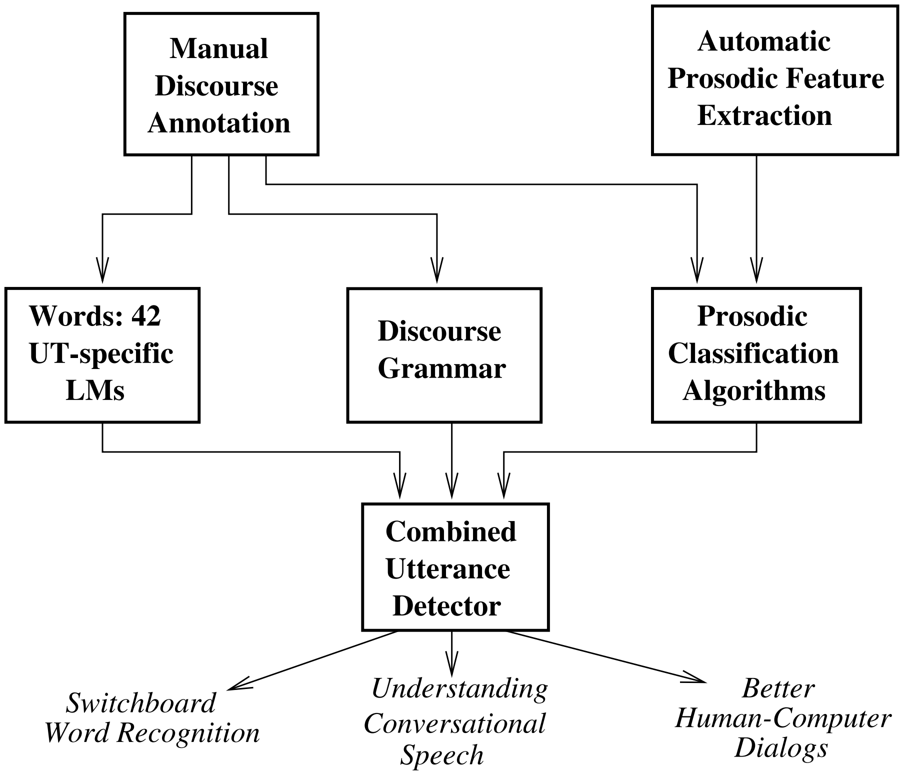

Figure 1: Outline of paper and project.

(Allen and Core 1997). We began with this markup system and modified in a number of ways to make it more useful for our purposes in annotating Switchboard.

Our initial tag-set consists of approximately 60 basic tags, many of which can be combined. We used this set to label 1155 conversations, comprising 205,000 utterances and 1.4 million words, from the Switchboard corpus of telephone conversations. The resulting set of 220 tags (combinations of basic tags used by the taggers) was then clustered by hand into 42 clusters. Table 2 shows the resulting 42 classes with their final counts in the WS97 training set (out of 197,489 training-set utterances, 1.4M words, 1115 conversations).

Note that our label-set incorporates both traditional sociolinguistic and discourse-theoretic rhetorical relations (or adjacency-pairs) as well as some more-form-based labels. Furthermore, the labelset is structured so as to allow labelers to annotate a Switchboard conversation in about 30 minutes, by editing it with any platform-independent editor (hence the short label-names, and the use of some rich cross-dimension labels). We expect these labeled conversations also to be useful for NLP and Conversational Analysis (CA) research. The labels were designed to be applied based on the Switchboard written transcriptions; this caused the label set to be somewhat more shallow than it could have been with the ability to listen to each utterance. We hope that this shallowness was balanced by the coverage; labeling quickly allowed us to cover much more data.

The labeling project started March 1, 1997, and finished July 5, 1997. The 8 labelers were CU Boulder linguistics grad students: Debra Biasca (supervisor), Marion Bond, Traci Curl, Anu Erringer, Michelle Gregory, Lori Heintzelman, Taimi Metzler, and Amma Oduro. Most of the utterances were presegmented by the Linguistic Data Consortium (Meteer *et al.* 1995), although a few had not been segmented, and had to be segmented by the labelers prior to labeling. By the end of the labeling the labelers took just under 30 minutes to label a conversation (conversations averaged 144-turns, 271 utterances). We used the Kappa statistic (Carletta 1996 and Carletta et al (in press)) to assess labeling accuracy; average pairwise Kappa (as of the end of the project) was 0.80. (0.8 or higher is considered high reliability (Carletta 1996; Flammia and Zue 1995).)

There is a deterministic mapping between about 80% of the "SWBD-DAMSL" labels we used and the standard DAMSL labels that we started from (Allen and Core 1997). In a few cases a mapping is not possible, usually for one of two reasons: either we and the coders were unable to accurately mark a distinction which the DAMSL standard

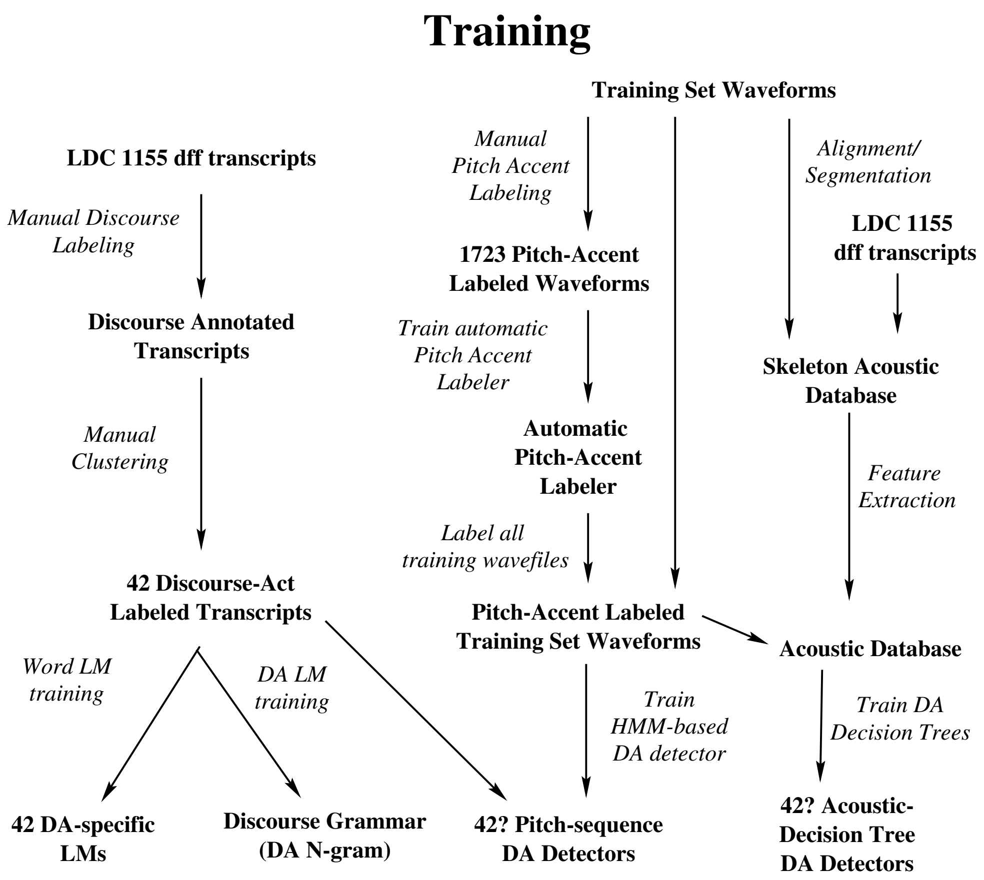

Figure 2: Plan of work for WS97: Training.

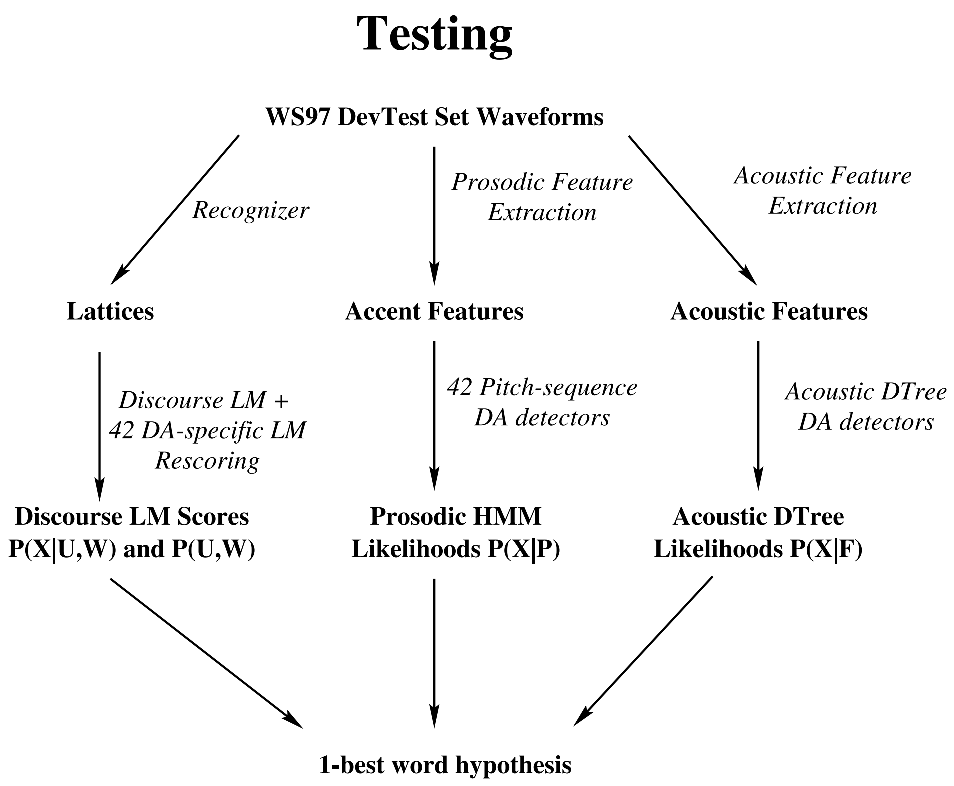

Figure 3: Plan of work for WS97: Testing.

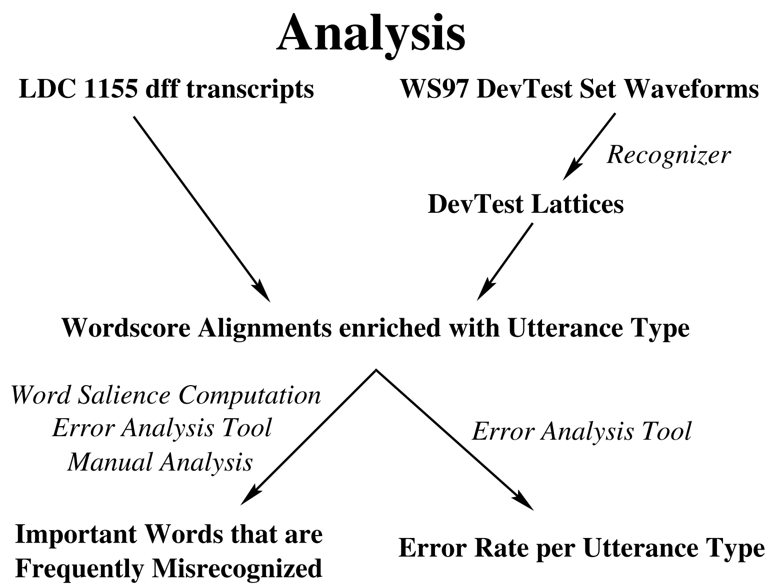

Figure 4: Plan of work for WS97: Analysis.

Table 2: *42 Dialog Acts with counts from the 197K-utterance training set. This is after we clustered various labels; the abbreviation sometimes but not always indicates which labels were clustered together.*

Tag	**Abbrev**	**Example**	**Count**	**%**
Statement-non-opinion	sd	*Me, I'm in the legal department.*	72,824	36%
Acknowledge (Backchannel)	b	*Uh-huh.*	37,096	19%
Statement-opinion	sv	*I think it's great*	25,197	13%
Agree/Accept	aa	*That's exactly it.*	10,820	5%
Abandoned or Turn-Exit	% . . . -/	*So, -/*	10,569	5%
Appreciation	ba	*I can imagine.*	4,633	2%
Yes-No-Question	qy	*Do you have to have any special training?*	4,624	2%
Non-verbal	x	*<Laughter>,<Throat_clearing>*	3,548	2%
Yes answers	ny	*Yes.*	2,934	1%
Conventional-closing	fc	*Well, it's been nice talking to you.*	2,486	1%
Uninterpretable	%	*But, uh, yeah*	2,158	1%
Wh-Question	qw	*Well, how old are you?*	1,911	1%
No answers	nn	*No.*	1,340	1%
Response Acknowledgement	bk	*Oh, okay.*	1,277	1%
Hedge	h	*I don't know if I'm making any sense or not.*	1,182	1%
Declarative Yes-No-Question	qy^d	*So you can afford to get a house?*	1,174	1%
Other	o,fo	*Well give me a break, you know.*	1,074	1%
Backchannel in question form	bh	*Is that right?*	1,019	1%
Quotation	^q	*You can't be pregnant and have cats*	934	.5%
Summarize/reformulate	bf	*Oh, you mean you switched schools for the kids.*	919	.5%
Affirmative non-yes answers	na	*It is.*	836	.4%
Action-directive	ad	*Why don't you go first*	719	.4%
Collaborative Completion	^2	*Who aren't contributing.*	699	.4%
Repeat-phrase	b^m	*Oh, fajitas*	660	.3%
Open-Question	qo	*How about you?*	632	.3%
Rhetorical-Questions	qh	*Who would steal a newspaper?*	557	.2%
Hold before answer/agreement	^h	*I'm drawing a blank.*	540	.3%
Reject	ar	*Well, no*	338	.2%
Negative non-no answers	ng	*Uh, not a whole lot.*	292	.1%
Signal-non-understanding	br	*Excuse me?*	288	.1%
Other answers	no	*I don't know*	279	.1%
Conventional-opening	fp	*How are you?*	220	.1%
Or-Clause	qrr	*or is it more of a company?*	207	.1%
Dispreferred answers	arp,nd	*Well, not so much that.*	205	.1%
3rd-party-talk	t3	*My goodness, Diane, get down from there.*	115	.1%
Offers, Options & Commits	oo,cc,co	*I'll have to check that out*	109	.1%
Self-talk	t1	*What's the word I'm looking for*	102	.1%
Downplayer	bd	*That's all right.*	100	.1%
Maybe/Accept-part	aap/am	*Something like that*	98	<.1%
Tag-Question	^ g	*Right?*	93	<.1%
Declarative Wh-Question	qw^d	*You are what kind of buff?*	80	<.1%
Apology	fa	*I'm sorry.*	76	<.1%
Thanking	ft	*Hey thanks a lot*	67	<.1%

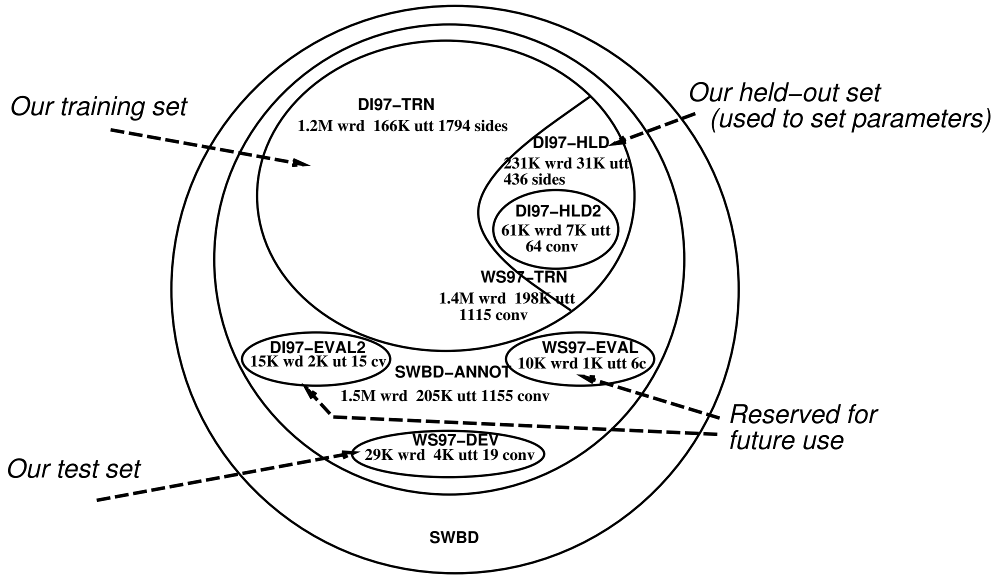

Figure 5: Training and test sets for Discourse LM group for WS97.

requires (for example the distinction between Assert and Reassert), or we felt the need to mark extra distinctions which DAMSL doesn't require. However in a few other cases we have proposed minor augmentations to DAMSL. One such example is modifying Self-Talk to include not one but 2 kinds of non-second-person- directed talk; self-talk and third-party talk). We have not attempted in this report to map these DAMSL-style tags into other theories of speech acts, intention-tracking in discourse, discourse commitment, centering, etc. See our Coders' Manual (Jurafsky *et al.* 1997b) for more theoretical justifications for the particular tagging philosophy.

3.1 Examples of the Dialog Acts

See Jurafsky *et al.* (1997b) for a complete description of the discourse acts. Here we will just summarize the types which are likely to play a significant role in our utterance-detection efforts.

3.1.1 Statements

The most common types of utterances (49% of the utterance tokens, covering 83% of the word tokens) were **Statements** which were divided into **Statement (sd)** and **Opinion (fx.sv)**. This split distinguishes "descriptive, narrative, or personal" statements (sd) from "other-directed opinion statements" (sv). The distinction was designed to capture the different kinds of responses we saw to opinions (which are often countered or disagreed with via further opinions) and to statements (which more often get continuers/backchannels). We have not yet decided whether this sd/sv distinction has been fruitful. In one experiment (see §6.1), we trained separate trigram language models on the two sets, and they looked somewhat distinct. But the distinction was very hard to make by labelers, and accounted for a large proportion of our interlabeler error.

sv's often include such hedges as *I think*, *I believe*, *It seems*, *I mean*, and others.

We did combine the **Statement** and **Opinion** classes for some experiments on dimensions in which they did not differ.

3.1.2 Questions

We focused for the summer on questions that tend to have rising intonation. These included **Yes-No-Questions**, **Tag Questions**, **Declarative-Questions**, **Or-Questions** and other question types that we have clustered together with these. But we labeled many other kinds of questions as well.

The **Yes-No-Question** label only includes utterances which both have the pragmatic force of a yes-no-question *and* have the syntactic and prosodic markings of a yes-no question (i.e. subject-inversion, question intonation).

Table 3: *Sample statements.*

Dialog Act	Utterance
Statement	Well, we have a cat, um,
Statement	He's probably, oh, a good two years old, big, old, fat and sassy tabby.
Statement	He's about five months old
Opinion	Well, rabbits are darling.
Opinion	I think it would be kind of stressful.

Table 4: *Sample questions.*

Dialog Act	Utterance
Yes-No-Question	Do you have to have any special training?
Yes-No-Question	But that doesn't eliminate it, does it?
Yes-No-Question	Uh, I guess a year ago you're probably watching C N N a lot, right?
Declarative-Question	So you're taking a government course?
Wh-Question	Well, how old are you?

Tag Questions (originally **qy^g**) have been clustered with **qy**. (Tag Questions have declarative syntax until the last word or two, which is either an auxiliary and usually a pronoun (`does it, isn't it, aren't they`), or a tag like `right?` or *huh?*. Here's two examples:

```
But that doesn't eliminate it, does it? /

Uh,  I guess a year ago you're probably watching C N N a lot, right? /
```

Declarative-Questions (qy^d) are utterances which function pragmatically as questions but which do not have "question form". By this we mean that declarative questions normally have no wh-word as the argument of the verb (except in "echo-question" format), and have "declarative" word order in which the subject precedes the verb. See Weber (1993) for a survey of declarative question and their various realizations. Since declarative questions seemed to often have rising (question) intonation, we ended up clustering these together with the other rising questions (i.e. with **Yes-No-Questions**). Here's an example:

```
So you're taking a government course? /
```

3.1.3 Backchannels

A backchannel is a short utterance which plays discourse-structuring roles like indicating that the speaker should go on talking. These are usually referred to in the CA literature as a "continuer", and there is an extensive literature on them (Jefferson 1984; Schegloff 1982; Yngve 1970). Recognizing them is important first because of their discourse-structuring role (knowing that the hearer expects the speaker to go on talking tells us something about the course of the narrative) and second because they seem to occur at certain kinds of syntactic boundaries; detecting a backchannel may thus help in segmentation and in word grammar.

For an intuition about what backchannels look like, Table 5 shows the most common realizations of the approximately 300 types (35,827 tokens) of backchannels in our SWBD corpus. Table 6 shows examples of backchannels in the context of a Switchboard conversation.

Table 5: *Realizations of backchannels.*

%	Backchannel
38%	uh-huh
34%	yeah
9%	right
3%	oh
2%	yes
2%	okay
2%	oh yeah
1%	huh
1%	sure
1%	um
1%	huh-uh
1%	uh

Table 6: *Examples: Backchannels.*

Spkr	Dialog Act	Utterance
B	**Statement**	*but, uh, we're to the point now where our financial income is enough that we can consider putting some away –*
A	**Backchannel**	Uh-huh. /
B	**Statement**	*– for college, /*
B	**Statement**	*so we are going to be starting a regular payroll deduction –*
A	**Backchannel**	Um. /
B	**Statement**	*— in the fall /*
B	**Statement**	*and then the money that I will be making this summer we'll be putting away for the college fund.*
A	**Appreciation**	Um. Sounds good.

3.1.4 Turn Exit and Abandoned

Abandoned utterances are those that the speaker breaks off without finishing, and are followed by a restart. Turn exits resemble abandoned utterances in that they are often syntactically broken off, but they are used mainly as a way of passing speakership to the other speaker. Turn exits tend to be single words, often `so` or `or`.

3.1.5 Answers and Agreements

The **Answer** category includes any sort of answers to questions. We were mainly interested in modeling two simple kinds of answers: **Answer-Yes** and **Answer-No**. **Answer-Yes** includes "yes", "yeah", "yep", "uh-huh", and such other variations on "yes", when they are acting as an answer to a **Yes-No-Question**. Detecting **Answer**s can help tell us that the previous utterance was a **Yes-No-Question**. Answers are also semantically quite significant as they are very likely to contain important new information.

The **Agreements** (**Accept**, **Reject**, **Partial Accept** etc) all mark the degree to which speaker accepts some previous proposal, plan, opinion, or statement. We are mostly interested in **Agree/Accepts**, which for convenience we will refer to as **Agree**s. These are very often `yes` or `yeah` and so they look a lot like **Answers**. But where answers follow questions, agreements often follow opinions or proposals, so distinguishing these can be important for the discourse.

Table 7: *Examples: Abandoned and Turn Exits.*

Spkr	Dialog Act	Utterance
A	**Statement**	we're from, uh, I'm from Ohio /
A	**Statement**	and my wife's from Florida /
A	**Turn-Exit**	so, -/
B	**Backchannel**	*Uh-huh. /*
A	**Hedge**	so, I don't know, /
A	**Abandoned**	it's <lipsmack>, - /
A	**Statement**	I'm glad it's not the kind of problem I have to come up with an answer to because it's not –

3.1.6 Clustering 220 tags into 42 clusters

This section gives a quick discussion on how we came up with 42 clusters. The taggers made use of 220 tags in the coding; 130 of these occurred less than 10 times each, so for our initial experiments we clustered the 220 tags into 42 larger classes. The approximately 60 'basic tags' combined into 220 final tags because some tags marked independent dimensions which could be combined. For example the speakers often had a meta-discussion on the task of having and recording a conversation, including utterances like the following:

```
I almost forgot what the topic <laughter> was. /
```

These were marked with a special tag, **About-Task** (`^t`) in addition to their normal tag of **Non-Opinion-Statement**. There were a number of such dimensions, each indicated by a carat (^2,^g,^m,^r,^e,^q,^d). For the purposes of clustering tags, we removed all of these carats with 5 exceptions. The exceptions: we left qy^d (Declarative yes-no Questions) , qw^d (Declarative wh-questions) and b^m (Signal-Understanding-via-Mimic), and we folded the few examples of nn^e into ng, and ny^e into na. Then, we grouped together some tags that had very little training data; those tags that appear in the following list were grouped with other tags on the same line. We did this grouping by looking at the words in the utterance, and the discourse function of the utterance.

```
qr qy
fe ba
oo co cc
fx sv
fo o fw " by bc
aap am
arp nd
```

We also removed any line with a "@" (since @ marked slash-units with bad segmentation).

4 Dialog Act Detection

The goal of our dialog act detection algorithms is to automatically assign the correct tag from our 42 DA set to each of the presegmented utterance wavefiles. As we suggested in the Introduction, we achieved a 65% detection accuracy, based on automatic word recognition and prosodic analysis. This compares with a baseline of 35% if we simply chose the most frequent dialog act each time. Human labelers were able to do significantly better (average pairwise agreement of human labelers was 84%). However, note that the human labeling was based purely on word transcripts. Using actual, rather than recognized words, our DA detection algorithm achieved 72% accuracy, so we can expect substantially improved automatic detection simple as a result of continually improving recognition accuracy.

Our algorithm is based on combining three sources of knowledge: **prosodic** knowledge, information about **word-sequences**, and **discourse grammar**, i.e., knowledge about sequences of dialog acts. We first summarize and motivate these knowledge sources, and then describe each of them in detail, as well as how they are combined.

Word-based DA detection is based on separate trigram language models for each of the 42 dialog acts, and choosing the dialog act that assigns the highest likelihood to the word string. This technique is common in subtopic identification (Hearst 1997) and in the cue-word literature (Garner et al 1996, Hirschberg and Litman 1993, etc).

This technique relies on the fact that utterances have very distinct word strings, and indeed they seem to. For example, **92.4%** of the **"uh_huh"**'s occur in **Backchannels 88.4%**, while of the trigrams **"<start> do you"** occur in **Yes-No-Questions**.

Prosodic detection uses CART-style decision trees which take assorted raw and derived acoustic features (such as pitch and speaking rate) to predict the dialog act of an utterance. Our work extends earlier work by others on the use of prosodic knowledge for dialog act prediction (Mast *et al.* 1996; Taylor *et al.* 1996; Taylor *et al.* 1997; Terry *et al.* 1994; Waibel 1988)

Prosodic information is essential for utterance recognition because the words alone aren't sufficiently distinguishing. This is true in reference strings, but is even more true in errorful hypothesized word strings, particular given the high deletion rate of utterance-initial words. For example our word-based detector only has **32%** accuracy in detecting questions. But **Yes-No-Questions** can usually be detected by looking for their final F0 rise.

Our final knowledge source is the discourse grammar which constrains the sequence of possible dialog acts. We trained a bigram discourse grammar and used this to assign prior probabilities to an utterance realizing a certain dialog act in a given context. The use of N-gram discourse grammars was motivated by previous work by (Kita *et al.* 1996; Mast *et al.* 1996; Nagata and Morimoto 1994; Suhm and Waibel 1994; Taylor *et al.* 1996; Taylor *et al.* 1997; Woszczyna and Waibel 1994; Yamaoka and Iida 1991) Indeed, the discourse grammar bigrams are quite distinct from a uniform prior (which would be 1/42 or 0.024 for each DA). For example, out of the 42 possible dialog acts a **Command** will be **Agreed** to with probability **0.23**, a **Yes-No-Question** will receive a **Yes** answer with probability **0.30**.

4.1 Dialog Act Segmentation

The utterance detection algorithm we describe is based on hand-segmented utterance boundaries. That is, both our training and test sets were segmented by hand into turns and utterances. This was a purely pragmatic decision; we found the detection problem difficult and interesting enough without the added complication of segmentation. Furthermore, we did not want to confound the issue of DA classification with DA segmentation, or to treat DAs at turn boundaries (easy to segment) better than those not at turn boundaries (harder to segment).

For our discourse grammars to be embedded in a complete speech recognizer, we will eventually need to automatically detect utterance boundaries; that is, we will need to segment the input stream into utterances that can then be labeled with dialog acts. This segmentation problem has begun to be addressed by the community; prosodic knowledge plays an important role. For example Stolcke and Shriberg (1996) reported preliminary results on utterance segmentation in Switchboard (assuming knowledge of the correct word sequence), and many others have looked at segmentation algorithms on various domains and for various purposes. Mast *et al.* (1996) achieved 92.5% accuracy in an automatic utterance segmentation algorithm for spontaneous VERBMOBIL dialogs. For recent work on segmentation see also (Grosz and Hirschberg 1992; Hearst 1997; Hirschberg and Nakatani 1996; Lavie *et al.* 1996a; Lavie *et al.* 1996b; Ostendorf and Veilleux 1994; Passonneau and Litman 1993).

Segmentations were hand-labeled between words in the text transcription. But we also needed to extend these segmentations to the acoustic waveform. To estimate the locations of the boundaries in the speech waveforms, a forced alignment of the acoustic training data was merged with linguistically annotated training transcriptions from the LDC. This yielded word and pause times of the training data with respect to the acoustic segmentations. Using these word times along with the linguistic segmentation marks, the start and stop times for linguistic segments were found.

This technique was not perfect, however, for several reasons. One is that many of the words included in the more careful linguistic transcription had been excised from the acoustic training data. This happened in two different ways. Some speech segments were considered not useful for acoustic training so were excluded deliberately. Also, the alignment program was allowed to skip words at the beginning and ending of an acoustic segment if not enough evidence of the word existed. While this could be due to a long pause between words, it may also be due to a compressed pronunciation of some words such as "Did you" be pronounced as "Dja". If times were available for some words in an utterance, even though the end words were missing times, we noted the available times as well as how many words were missing from the times and if they were at the beginning or end (or both) of the utterance.

Errors in the boundary times for DAs crucially effect the prosodic analyses, since prosodic features are extracted assuming the boundaries are reasonably correct. Incorrect estimates affect the accuracy of global features (e.g., DA duration), and may render local features (e.g. F0 measured at the supposed end of the utterance) meaningless. Since features for DAs with known problematic end estimates would be misleading in the prosodic analyses, they were omitted from our training (TRN) and held-out test (HLD) data.

Overall, we were missing 30% of the training utterances because of problems with time boundaries. While the majority of the words in the training data were included (i.e., enough data for acoustic modeling purposes), we were missing up to 45% of some types of utterances, backchannels in particular. While these utterances may not contribute to a significant drop in error rate, they are important for modeling the flow of the conversation.

The time boundaries of the DEV development set, however, were carefully handmarked for other purposes (many thanks to Joe Picone and Jon Hamaker for the handmarking), so we were able to use exact values for this test set. It should be noted that this difference in segmentation method makes the DEV set somewhat mismatched with respect to the training data.

4.2 Hidden Markov modeling of discourse

Our goal is to perform DA detection using a probabilistic framework, giving us a principled approach for combining multiple knowledge sources (using the laws of probability), as well as the ability to derive model parameters automatically from a corpus, using statistical inference techniques.

Given all available evidence E about a conversation, our goal is to find the DA sequence U that has the highest *posterior probability* $P(U|E)$ given that evidence. Applying Bayes' Rule we get

$$\begin{aligned} U^* &= \operatorname*{argmax}_U P(U|E) \\ &= \operatorname*{argmax}_U \frac{P(U)P(E|U)}{P(E)} \\ &= \operatorname*{argmax}_U P(U)P(E|U) \end{aligned} \tag{1}$$

Here $P(U)$ represents the prior probability of a DA sequence, and $P(E|U)$ is the likelihood of U given the evidence. The likelihood is usually much more straightforward to model than the posterior itself, which has to do with the fact that our models are generative or causal in nature, i.e., they describe how the evidence is produces by the underlying DA sequence U.

Estimating $P(U)$ amounts to building a probabilistic discourse grammar, a statistical model of DA sequences. We will do so using familiar techniques from language modeling for speech recognition, although the sequenced objects in this case are, of course, not words but DA labels.

4.2.1 DA Likelihoods

The computation of likelihoods $P(E|U)$, on the other hand, depends on the types of evidence used. In our experiments we used the following sources of evidence, either alone or in combinations:

Transcribed words: The likelihoods used in Eq. 1 are $P(W|U)$, where W refers to the true (hand-transcribed) words spoken in a conversation.

Recognizer acoustics: The evidence consists of recognizer acoustics A, and we seek to compute $P(A|U)$. As described later, this involves considering multiple alternative recognized word sequences.

Prosodic features: Evidence is given by the acoustic features F capturing various aspect of pitch, duration, energy, etc., of the speech signal; the associated likelihoods are $P(F|U)$.

To make both the modeling and the search for the best DA sequence feasible, we further require that our likelihood models are *decomposable by utterance*. This means that the likelihood given a complete conversation can be factored into likelihoods given the individual utterances. We use U_i for the ith DA label in the sequence U, i.e.,

$$U = (U_1, \ldots, U_i, \ldots, U_n)$$

where n is the number of utterances in a conversation. Also, we use E_i for that portion of the evidence that corresponds to the ith utterance, e.g., the words or the prosody of the ith utterance. Decomposability of the likelihood means that

$$P(E|U) = P(E_1|U_1) \cdot \ldots \cdot P(E_n|U_n)$$

Applied to the three types of evidence introduced before, it is clear that this property is not strictly true. For example, speakers might tend to reuse words found earlier in the conversation, violating the independence of the $P(W_i|U_i)$. Similarly, speakers might adjust their pitch or volume globally, e.g., to the conversation partner, violating the independence of the $P(F_i|U_i)$. As in other areas of statistical modeling, we count on the fact that these violations are are small compared to the properties actually modeled, namely the dependence of E_i on U_i.

4.2.2 Markov modeling

Returning to the prior of DA sequences $P(U)$, it is extremely convenient to make certain independence assumptions here, too. In particular, we will assume that the prior distribution U is Markovian, i.e., that each U_i depends only on a fixed number k of preceding DA labels:

$$P(U_i|U_1, \ldots, U_{i-1}) = P(U_i|U_{i-k}, \ldots, U_{i-1})$$

(k is the order of the Markov process describing U). The N-gram based discourse grammars we used have this property. As described later, $k = 1$ is a very good choice, i.e., conditioning on the DA types more than one removed from the current one does not improve the quality of the model by much.

The importance of the Markov assumption for the discourse grammar is that we can now view the whole system of discourse grammar and local utterance-based likelihoods as a kth-order *hidden Markov model* (HMM) (Rabiner and Juang 1986). The HMM states correspond to DAs, observations corresponds to utterances, transition probabilities are given by the discourse grammar, and observation probabilities are given by the local likelihoods $P(E_i|U_i)$. This allows us to use efficient dynamic programming algorithms to compute the relevant aspects of the model, such as:

- the most probable DA sequence (the Viterbi algorithm)
- the posterior probability of various DAs for a given utterance, after considering all the evidence (the forward-backward algorithm)

We could even try to learn discourse grammars and likelihood models in an unsupervised way, using a Baum-Welch type estimator (but this is beyond the scope of the present work).

4.2.3 Viterbi versus Forward-backward decoding

The Viterbi algorithm for HMMs finds the globally most probable state sequence. When applied to a discourse model with locally decomposable likelihoods and Markovian discourse grammar it will therefore find precisely the DA sequence with the highest posterior probability:

$$U^* = \operatorname*{argmax}_U P(U|E)$$

This maximizes the probability of getting the *entire* DA sequence correct, but it does not necessarily find the DA sequence that has the most DA labels correct (Stolcke *et al.* 1997). To minimize the overall utterance labeling error, we need to maximize the probability of getting each DA label correct individually, i.e., we need to maximize $P(U_i|E)$ for each $i = 1, \ldots, n$. We can compute the per-utterance posterior DA probabilities by summation:

$$P(u|E) = \sum_{U_i = u} P(U|E)$$

where the summation is over all sequences U whose ith element matches the label in question. The summation is efficiently carried out by the forward-backward algorithm for HMMs.

For 0th-order (unigram) discourse grammars Viterbi and forward/backward decoding yields the same results. For bigram and trigram discourse grammars we found that forward-backward decoding consistently gives slightly (up to 1% absolute) better accuracies, as expected. All results reported here, unless noted otherwise, were obtained using the forward-backward decoding method.

4.2.4 Probability scaling

Both the discourse grammar and the likelihood models used in this framework usually represent approximation to the true distributions they attempt to model. When combining different models we may therefore have to adjust the dynamic ranges of the probability estimates for optimal results in the joint maximization of Eq. 1. This is done by scaling the probabilities in the log domain, i.e., by actually maximizing

$$P(U)^{\delta} P(E|U)^{\mu}$$

The discourse grammar weight δ and the likelihood weight μ are determined through optimization on a held-out set. For Viterbi decoding only one weight is required, since the maximization depends only on the ratio δ/μ, but for forward-backward decoding both parameters matter.

4.3 Discourse Grammars

4.3.1 N-gram models

As motivated earlier, it is computationally attractive to model DA sequences as Markov chains. We implemented this approach by estimating backoff N-gram models (Katz 1987) from the hand-labeled DA sequences of the training data available to us. Backoff N-gram models consist of conditional probabilities

$$p(U_i|U_{i-N+1}, \ldots, U_{i-1})$$

based on the frequencies of N-tuples of tokens in the training data, i.e., they predict token sequences left-to-right, conditioning the predictions on the previous $N - 1$ tokens at each position. The Witten-Bell smoothing scheme was used to discount the relative frequencies (Witten and Bell 1991).

4.3.2 Modeling turns

So far we have ignore the fact that DAs are associated with multiple speakers. Surely it is important to model not only the DA sequence, but also which speaker said what. For example, in the sample conversation of Table 1, the grammar should compute the probability of the second utterance in Channel A being a **Ack-Answer**, given that the previous utterance was a **Statement** on Channel B and before that was a **Wh-Question** on Channel A.

We can think of the events in the sequence as consisting of pairs (U_i, T_i), where U_i is a DA label and T_i is a speaker label. Switchboard conversations involve exactly two speakers each, conventionally labeled **A** and **B** corresponding to the channels they were recorded on. The vocabulary of our discourse grammar therefore consisted of $42 \times 2 = 84$ token types, plus special tokens for the beginning and end of conversations.

We need to ensure that the discourse grammar is symmetrical with respect to the speaker labels A and B, since the assignment of channels during training is purely incidental. Our approach was to duplicate the training data with channels switched and pool it with the original data for purposes of discourse grammar training.

It should be noted that our modeling of turn exchanges is simplistic and inaccurate in many respects. It assumes that both speakers' utterances occur in strict sequence. In actuality spontaneous speech is characterized by considerable overlap of turns. Backchannels often occur in the middle of the other speaker's utterance, which therefore both precedes and follows the backchannel utterance in time. We made no attempt to encode such overlap; rather, utterances were serialized in the order given by their start times.

4.3.3 A random discourse sample

To give an intuitive feel for the discourse grammar, we used our bigram grammar to generate a 'random conversation' by randomly generating a sequence of dialog acts according to their discourse grammar probability, and then selecting a random instance of each dialog act from our corpus. This is seen in Table 8.

4.3.4 DA perplexities

As a general item of interest, as well as to compare different discourse grammars, we computed the perplexities of the DA label sequences. DA perplexity measures how many choices, on average, the discourse grammar allows for the next DA label. It is defined as

$$\text{Perplexity}(U) = P(U)^{-\frac{1}{n}}$$

Table 8: *A randomly generated conversation.*

Spkr	Dialog Act	Utterance
A	**Backchannel**	Uh-huh.
A	**Statement**	And, it was so good.
B	**Backchannel-Q**	*Really.*
B	**Statement**	*But when we found it, it was, we knew it.*
A	**Statement**	and, as luck had it, uh, she had puppies the same week my daughter was born,
A	**Backchannel**	Uh-huh.
A	**Statement**	I hear that from, my brother-in-law lives in Plano,
B	**Backchannel-Q**	*Right.*
B	**Statement**	*And I like Demi Moore.*
B	**Statement**	*I just still have need for the four bedrooms just having having company from time to time.*
A	**Appreciation**	you're not kidding.
A	**Agree**	Absolutely.
A	**Agree**	Absolutely.

where $P(U)$ is probability of a test corpus, and n is the length of the test corpus.

First, we consider the perplexity of just the DA labels, i.e, without taking turn information into account. We trained a DA-trigram model on the WS97-TRAIN set (197,564 utterances) and tested its perplexity on the WS97-DEV set (4,190 utterances). Table 9 shows the perplexity of unigram, bigram, and trigram versions of this model. The first row gives the perplexity in the absence of a discourse model, i.e., if all DAs had equal probability.

Table 9: *Perplexity when guessing just DA with no turn information.*

Discourse Grammar	Perplexity
None	42
Unigram	11.0
Bigram	7.9
Trigram	7.5

In the next step we added turn information to the discourse model. As discussed above, we did so by augmenting the DA labels with speaker labels (A and B). Results are shown in Table 10.

Table 10: *Perplexity when guessing both the DA and turn information.*

Discourse Grammar	Perplexity
None	84
Unigram	18.5
Bigram	10.4
Trigram	9.8

Note that the number of possible events is twice that of the original, DA-only model, explaining the higher perplexities. However, the perplexities are less than double that of the original model, indicating that speaker changes are partly predictable from the DA sequences. This raises another question of general interest: What is the perplexity of the speaker label sequence alone, when conditioning on the previous speakers *and* DA labels? The answer, using a trigram model, is 1.97, showing that turn changes are still quite unpredictable, even given preceding DAs.

The perplexity of the speaker label stream *without* knowledge of the discourse acts was slightly higher (including end-of-conversation tags:

Table 11: *Perplexity of speaker/turn stream only.*

Discourse Grammar	Perplexity
None	3
Unigram	2.1
Bigram	2.0
Trigram	2.0

Finally, we are interested in the perplexity of DAs given that the speaker of each utterance is known. This is relevant for Switchboard and other corpora where, due to multi-channel recording or speaker identification methods, the turn changes can be inferred from the timing of the speech signal. In other words, we wish to compute the perplexity of the conditional distribution $P(U|T)$. Using

$$P(U|T) = \frac{P(U,T)}{P(T)}$$

we can estimate the perplexity of the conditional model by dividing the perplexity of the joint DA-speaker label model above by the perplexity of just the turn labels. The results are shown in Table 12.

Table 12: *DA Perplexity conditioned on turn information.*

Discourse Grammar	Perplexity
None	42
Unigram	9.0
Bigram	5.1
Trigam	4.8

4.3.5 Using turn information during decoding

It is straightforward to use known turn information (speaker identities) in the HMM framework described in Section 4.2. Instead of one HMM state per DA label we have one state for each DA-speaker pair. During decoding we set the likelihoods to zero for those states that are inconsistent with the known speaker identity. The state transitions are given by a joint DA-speaker N-gram model as described above.

4.3.6 Modeling alternatives for Discourse Grammars

The use of a relatively simple backoff model for LVCSR word N-gram grammars (like the Witten-Bell smoothing we used for our word N-grams) is often dictated for three reasons: other models are too hard to estimate, the implicit ordering of the constraints (trigram, bigram, unigram) is fairly well motivated, and the time and space requirements for other models could be prohibitive. However this does not seem to hold for the discourse grammars discussed here: Compared to the number of words, our utterance-based training corpus is fairly small (197,489 training utterances vs. 1.4 million training words), the number of types is small (42 dialog act types vs. 30,000 words in a typical SWBD LVCSR system) and the structuring of the context is not necessarily hierarchical (the conversation is carried out on two channels and the dialogue acts are overlapping).

We therefore decided to investigate more sophisticated models for discourse grammar than simple backoff N-grams. In particular, we wanted to try models that we thought could incorporate the following two kinds of knowledge:

Non-hierarchical constraints: Standard N-gram backoff models have a simple notion of constraint hierarchy (trigram, bigram, unigram). But for dialog acts it is not as clear a priori how we should model the history. We

Table 13: *Perplexities of maximum entropy discourse grammars.*

Features								Perplexity
Unigram	Speaker-change	Anychannel			Samechannel Bigram	Otherchannel Bigram	Trigger	
		Bigram	Trigram	Skip1-Bigram				
•		•						5.55
•		•	•					5.25
•		•	•				•	5.30
•		•	•	•				5.24
•	•	•			•	•		5.30
•	•	•	•		•	•		5.30
•					•			6.65
•						•		7.06

explored the use of maximum entropy (ME) models; these allow a more flexible notion of context than backoff models.

Long distance knowledge: Dialog models could be more sensitive to long distance dependencies. For example, we might expect that particular dialog patterns between two speakers could stay constant over the conversation. We attempted to model this with standard cache and maximum entropy trigger models.

Maximum Entropy Models We implemented an improved Generalized Iterative Scaling (GIS) algorithm and trained a maximum entropy model for dialog acts with it (Della Pietra *et al.* 1997; Rosenfeld 1996). The maximum entropy model predicts the following dialog act using the following features:

- the last discourse act (on the same channel or on the other channel
- the last discourse act on any channel + the information whether the last discourse act was on the same or other channel
- like the previous feature but for the last two discourse acts
- like the previous feature but only conditioned on the discourse act + channel information before the last discourse act
- was discourse act X seen in the last n discourse acts?
- was the last discourse act on the same or other channel?

To compare the performance of our maximum entropy (ME) model with a backoff model, we trained both types of model on a corpus of DA labels without speaker labels, using uni- and bigrams. The DA perplexities obtained were 6.87 for the ME model and 6.98 for the bigram model. Next, we tried adding various types of constraints, including triggers, to the ME model, as shown in Table 13. Perplexity results indicate that it is not beneficial to add constraints other than those already represented in the N-gram backoff model. Our preliminary conclusion is therefore that ME and backoff models produce roughly equivalent results when used as discourse grammars.

Cache models We used a standard unigram and bigram cache model (Kuhn and de Mori 1990) and interpolated it with a backoff model that was generated as described above. This technique typical achieves a significant perplexity win on standard SWBD language models for words and we had hoped to see a similar improvement if not more for the Discourse Grammar. But we found no improvement in the perplexity of the interpolated model over the standard model alone.

Conclusions for Discourse Grammar We conclude that using current technology it is hard to do better than a simple backoff model for Discourse Grammars, as long as one encodes the turn information. Long distance modeling does not seem to have any significant impact and the standard hierarchical encoding of the history for the backoff model seems to be appropriate. These are strong indications that most dialog act selection decisions in SWBD are made very locally and global effects on it are minimal. Patterns that might render themselves as being global might be mostly corollaries of the local patterning rules. We should however place a strong caveat on this result: The speakers in SWBD (ideally) don't know each other, they take similar social roles in all conversations and they have no common history they can built on. We would assume this result to change significantly on a corpus like CallHome or CallFriend, where the speakers are acquainted with each other and the variability of discourse behavior is greater.

4.4 Dialog act detection using words

The first, and most straightforward source of evidence we examined are the true (hand-transcribed) word sequences found in different discourse acts. To compute word-based likelihoods $P(W|U)$ we built trigram language models for each of the 42 dialog acts.[1] All DAs of a particular type found in the WS97 training corpus were pooled and a DA-specific trigram model was built using standard techniques (Katz-backoff with Witten-Bell discounting).

The 42 models were applied to each of the utterances in the WS97 devtest set and the likelihoods obtained were combined with the N-gram discourse grammar described in Section 4.3. Results for discourse grammars of various orders are summarized in Table 14.

Table 14: *Results for DA detection from words.*

Discourse Grammar	Accuracy (%)	
	Full conv.	5 mins.
None	53.9	54.3
Unigram	69.0	68.1
Bigram	71.5	70.6
Trigram	72.0	71.9

Since the devtest conversations had been truncated to five minutes for recognition purposes, and a few utterances had been lost as a result of the segmentation process, we ran this experiment both on the full conversation transcript, and on the 5-minute subset of utterances used in later recognition experiments. As shown in Table 14, the accuracies on the full conversations were slightly better, probably because the discourse grammar was somewhat mismatched to the truncated conversations. The fact that the differences were minor is important, however, since we will later use the same discourse grammars in combination with recognizer outputs.

An interesting detail concerning the DA-specific language models is that they were *not* optimized for perplexity. As described later, it is desirable for use in word recognition to smooth the DA-specific models by interpolating them with a general all-corpus trigram model. However, this would decrease the discrimination between DAs. With smoothed DA-specific LMs and no discourse grammar, the detection accuracy drops to 37.6% (from 53.9% without smoothing), and results with discourse grammar deteriorate by about 1%.

The above results were obtained without optimizing the discourse grammar and likelihood weights (both were set to unity). Post hoc experiments showed that the optimal δ and μ were in fact very close to unity. This is not surprising since both discourse grammar and likelihoods stem from similar types of (i.e., N-gram) models.

4.5 Dialog act detection using recognized words

For fully automatic DA detection, the previous approach is only a partial solution, since we are not yet able to recognize words in spontaneous speech with high enough accuracy. A suboptimal approach is to trust the recognized words nevertheless and use them as input to the word-based detector described above. Using the standard WS97 recognizer and a bigram discourse grammar, this gives an accuracy of 61.3%, as compared to 70.6% when the true words are

[1]Note we are now talking about word-based N-gram models, not discourse grammars, whose tokens correspond to utterances.

available. This could probably be improved by retraining the DA-specific language models with recognizer output; however, this would a very time-consuming endeavor.

A more practical approach is obtained by deriving the DA likelihoods based on recognizer acoustics A. We compute $P(A|U)$ by decomposition into an acoustic likelihood $P(A|W)$ and a word-based likelihood $P(W|U)$, and summing over all word sequences:

$$\begin{aligned} P(A|U) &= \sum_W P(A|W,U)P(W|U) \\ &= \sum_W P(A|W)P(W|U) \end{aligned}$$

The second line is justified under the assumption that the recognizer acoustics (typically, cepstral coefficients) are invariant to DA type once the words are fixed. This is questionable; for example, a word pronunciation may change as a result of different emphasis placed on a word.

The acoustic likelihoods $P(A|W)$ are the acoustic scores from the recognizer, and have to be scaled (in the log domain) by the inverse of the recognizer language model weight to be compatible with $P(W|U)$. The word-based likelihoods are obtained from DA-specific language models as before. The summation over all W has to be approximated; we did so by summing over the 2500 best hypotheses output by the recognizer.[2]

Table 15 summarizes results using the N-best approach combined with discourse grammars of various orders. We observe about a 7% reduction absolute reduction in accuracy compared to using the true words.

Table 15: *Results for DA detection from N-best lists.*

Discourse Grammar	Accuracy (%)
None	42.8
Unigram	61.9
Bigram	64.6
Trigram	64.9

We also compared the 2500-best summation to an even grosser 1-best approximation. Here, only the single best hypothesis (according the DA-specific language model) is used in computing $P(A|U)$. The result is a drop in accuracy from 64.6% to 63.4%.

4.6 Dialog act detection using prosody

Our experiments on the use of prosodic knowledge for DA detection had a slightly different focus than our other experiments. Results from preliminary experiments revealed that the DA detection was driven largely by priors (encoded as unigram frequencies in the dialog grammar) because of an extreme skew in the distribution of DAs in the corpus. In order to understand whether prosodic properties of the utterances themselves can be used to predict DAs, we eliminate additional knowledge sources that could confound our results. Analyses are conducted in the "no-priors" domain (all DAs are made equally likely). We also exclude contextual information from the dialog grammar (such as the DA of the previous utterance). In this way, we hope to gain a better understanding of the prosodic properties of the different DAs, which can in turn be applied in building better integrated models for natural speech corpora in general.

Our approach builds on recent methodology that has achieved good success on conversational speech for a different task (Shriberg *et al.* 1997). The method involves construction of a large database of automatically extracted acoustic-prosodic features. In training, decision tree classifiers are inferred from the features; the trees are then applied to an unseen set of data to evaluate performance.

We apply the trees to four DA-classification tasks. We begin with a task involving all-way classification of the DAs in our corpus. We then examine three subtasks found to be problematic for word-based classification: question classification, agreement classification, and the classification of incomplete utterances. For each task, we build subtrees with various feature sets to gain an understanding of the relative importance of different prosodic features. In addition,

[2]The N-best list were generated using a standard, DA-unspecific trigram language model.

Table 16: *Duration features.*

Feature Name	Description
Duration	
ling_dur	duration of utterance (linguistically-segmented)
Duration-pause	
ling_dur_minus_min10pause	ling_dur minus sum of duration of all pauses of at least 100 msec
cont_speech_frames	number of frames in continuous speech regions (> 1 sec, ignoring pauses < 100msec)
Duration-correlated F0-based counts	
f0_num_utt	number of frames with F0 values in utterance (prob_voicing=1)
f0_num_good_utt	number of F0 values above f0_min (f0_min = .75*f0_mode)
regr_dur	duration of F0 regression line (from start to end point, includes voiceless frames)
regr_num_frames	number of points used in fitting F0 regression line (excludes voiceless frames)
numacc_utt	number of accents in utterance from event recognizer
numbound_utt	number of boundaries in utterance from event recognizer

we integrated tree models with DA-specific language models to explore the role of prosody when word information is also available.

4.7 Prosodic Features

In order to train our decision trees, we built a database of prosodic features for each utterance in our training and test sets. The prosodic database included a variety of features that could be computed automatically, without reference to word information. In particular we attempted to have good coverage for features and feature extraction regions that expected to play a role in the three focussed analyses mentioned in the Introduction: classification of questions, agreements, and incomplete utterances. Based on the literature on question intonation, we expected questions to show rising F0 at the end of the utterance, particularly for declarative and yes-no questions. Thus, F0 should be a helpful cue for distinguishing questions from other long DAs such as statements. Many incomplete utterances give the impression of being cut off prematurely, so the prosodic behavior at the end of such an utterance may be similar to that of the middle of a normal utterance. Specifically, energy can be expected to be higher at the end of an abandoned utterance compared to a completed one. In addition, unlike most completed utterances, the F0 contour at the end of the utterance is neither rising nor falling. For these reasons RMS energy and F0 were calculated additionally within regions near the end of the utterance. We expected backchannels to differ from agreements by the amount of effort used in speaking. Backchannels function to acknowledge another speaker's contributions without taking the floor, whereas agreements assert an opinion. We therefore expected agreements to have higher energy, greater F0 movement, and a higher likelihood of accents and boundary tones.

4.7.1 Duration and pause features

Duration was expected to be a good cue for discriminating statements and questions from DAs functioning to manage the dialog (e.g. backchannels), although this difference is also encoded to some extent in the language model. In addition to the duration of the utterance in seconds, we included features correlated with utterance duration but based on frame counts conditioned on the value of other feature types, as shown in Table 16.

The duration-pause set of features computes duration ignoring pause regions. Such features may be useful if pauses are unrelated to DA-classification. (If pauses are relevant however, this should be captured by the pause features described in the next section). The F0-based count features reflect either the number of frames or recognized intonational events (accents or boundaries) based on F0 information (see F0 features, below). The first four of these features capture time in speaking using knowledge about the presence and location of voiced frames, which may be more robust for our data than relying on pause locations from the alignments. The last two features are intended to

Table 17: *Pause features.*

Feature Name	Description
min10pause_count_n_ldur	number of pauses of at least 10 frames in utterance, normalized by duration of utterance
total_min10pause_dur_n_ldur	sum of duration of all pauses of at least 100msec in utterance, normalized by duration of utterance
mean_min10pause_dur_utt	mean pause duration for pauses of at least 10 frames in utterance
mean_min10pause_dur_ncv	mean pause duration for pauses of at least 10 frames in utterance, normalized by same in convside
cont_speech_frames_n	number of frames in continuous speech regions ($>$ 1 sec, ignoring pauses $<$ 10 frames) normalized by duration of utterance

capture the amount of information in the utterance, by counting accents and phrase boundaries. Duration-normalized versions of many of these features are included under their respective feature type in the following sections.

4.7.2 Pause features

To address the possibility that hesitation could provide a cue to the type of DA, we included features intended to reflect the degree of pausing, as shown in Table 17. To obtain pause locations we used information available from forced-alignments; however this was only for convenience (the alignment information was in our database for other purposes). In principle pause locations can be detected by current recognizers with high accuracy without knowledge of the words. Pauses with durations below 100 milliseconds (10 frames) were excluded since they are more likely to reflect segmental information than hesitation. Features were normalized to remove the inherent correlation with utterance duration. The last feature was included to provide a more global constraint; it counts only those speech frames occurring in regions of at least one second of continuous speaking.

4.7.3 F0 features

F0 features, shown in Table 18, included both raw and regression values based on frame-level F0 values from ESPS/Waves+. To capture overall pitch range, mean F0 values were calculated over all voiced frames in an utterance. To normalize differences in F0 range over speakers, particularly across genders, utterance-level values were normalized with respect to the mean and standard deviation for F0 values measured over the whole conversation side. F0 difference values were normalized on a log scale. The standard deviation in F0 over an utterance was computed as a possible measure of expressiveness over the utterance. Minimum and maximum F0 values, calculated after median smoothing to eliminate spurious values, were also included for this purpose.

We also included parallel measures that used only "good" F0 values, or values above a threshold ("f0_min") estimated as the bottom of a speaker's natural F0 range. The f0_min can be calculated in two ways. For both methods, a smoothed histogram of all the calculated F0 values for a conversation side is used to find the F0 mode. The true f0_min comes from the minimum F0 value to the left of this mode. Because the histogram can be flat or not sufficiently smoothed, our algorithm may be fooled into choosing a value greater than the true minimum. A simpler way to estimate f0_min takes advantage of the fact that values below the minimum typically result from pitch halving. Thus, a good estimate of f0_min is to take the point at 0.75 times the F0 value at the mode of the histogram. This measure closely approximates the true f0_min, and is more robust for use with the Switchboard data.[3]. The percentage of "good" F0 values was also included to measure (inversely) the degree of creaky-voice or vocal fry.

The rising/falling behavior of contours is a good cue to their utterance type. We investigated a number of ways to measure this behaviour. To measure overall slope, we calculated the gradient of a least-squares fit regression line for the F0 contour. While this gives an adequate measure for the overall gradient of the utterance, it is not always a good indicator of the type of rising/falling behavior we are most interested in. Rises at the end can be swamped by the declination of the part of the contour preceding this, and hence the overall gradient for a contour can be falling. We therefore marked two special regions at the end of the contour, corresponding to the last 200ms ("end" region) and the previous 200ms to that ("penultimate" region). For each of these regions we measured the mean F0 and gradient,

[3] We thank David Talkin for suggesting this method.

Table 18: *F0 features.*

Feature Name	Description
f0_mean_good_utt	mean of F0 values included in f0_num_good_utt
f0_mean_n	difference between mean F0 of utterance and mean F0 of convside for F0 values > f0_min
f0_mean_ratio	ratio of F0 mean in utterance to F0 mean in convside
f0_mean_zcv	mean of good F0 values in utterance normalized by mean and st dev of good F0 values in convside
f0_sd_good_utt	st dev of F0 values included in f0_num_good_utt
f0_sd_n	log ratio of st dev of F0 values in utterance and in convside
f0_max_n	log ratio of max F0 values in utterance and in convside
f0_max_utt	maximum F0 value in utterance (no smoothing)
max_f0_smooth	maximum F0 in utterance after median smoothing of F0 contour
f0_min_utt	minimum F0 value in utterance (no smoothing); can be below f0_min
f0_percent_good_utt	ratio of number of good F0 values to number of F0 values in utterance
utt_grad	least-squares all-points regression over utterance
pen_grad	least-squares all-points regression over penultimate region
end_grad	least-squares all-points regression over end region
end_f0_mean	mean F0 in end region
pen_f0_mean	mean F0 in penultimate region
abs_f0_diff	difference between mean F0 of end and penultimate regions
rel_f0_diff	ratio of F0 of end and penultimate regions
norm_end_f0_mean	mean F0 in end region normalized by mean and st dev of F0 from convside
norm_pen_f0_mean	mean F0 in penultimate region normalized by mean and st dev from convside
norm_f0_diff	difference between mean F0 of end and penultimate regions, normalized by mean and st dev of F0 from convside
regr_start_f0	first F0 value of contour, determined by regression line analysis
finalb_amp	amplitude of final boundary (if present), from event recognizer
finalb_label	label of final boundary (if present), from event recognizer
finalb_tilt	tilt of final boundary (if present), from event recognizer
numacc_n_ldur	number of accents in utterance from event recognizer, normalized by duration of utterance
numacc_n_rdur	number of accents in utterance from event recognizer, normalized by duration of F0 regression line
numbound_n_ldur	number of boundaries in utterance from event recognizer, normalized by duration of utterance
numbound_n_rdur	number of boundaries in utterance from event recognizer, normalized by duration of F0 regression line

and used the differences between these as features. The starting value in the regression line was also included as a potential cue to F0 register (the actual first value is prone to F0 measurement error).

4.7.4 HMM-based Event Detection

In addition to these F0 features, we also included intonational-event features, or features intended to capture local pitch accents and phrase boundaries. The event features obtained using the event recognizer described in (Taylor *et al.* 1997). This detector relies on the intuition that different utterance types are characterized by different intonational 'tunes' (Kowtko 1996), and has been applied successfully to the detection of move types in the Maptask corpus (Bard *et al.* 1995). The system detects sequences of distinctive pitch patterns which characterize particular utterance types, by training one continuous density HMM for each DA to be detected (Taylor *et al.* 1997).

Taylor *et al.*'s (1997) algorithm actually proceeds in 3 steps.

Step 1: Find Intonational Event Loci

Step 2: Compute Pitch Patterns for each Locus

Step 3: Detect Characteristic Pitch Pattern Sequences

For our database of prosodic features, we actually only used Step 1 of the algorithm. In preliminary experiments we had also tried using the complete Taylor *et al.* (1997) algorithm, but we decided to concentrate on a single algorithm, and settled on the decision-tree methodology. Step 1 uses a simple set of 5 HMMs (completely separate "event-HMMs" not to be confused with the DA-detector HMMs) to detect areas where intonational events are likely to occur. This locus-finding stage consists of 5 simple event-detector HMMs. Each of them takes as input (observations) F0, energy, delta-f0 and delta-energy. Each is trained to detect one of the following 5 event types:

[a:]	pitch accent
[b:]	boundary
[c:]	connection
[sil:]	silence
[ab:]	accent+boundary

The **a** and **b** labels are called intonational *events* and represent the linguistically significant portion of the intonation contour. **c** is used simply to fill in parts of the contour which are not an event or silence. The compound label **ab** is used for when an accent and boundary are so close they overlap and form a single intonational event. Figure 6 shows a sample 'event-detector HMM'.

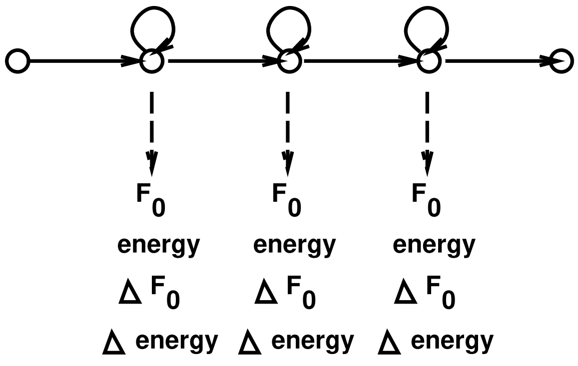

Figure 6: Sample event-detector HMM for label **a**.

Following is a sample pitch locus labeling:

```
i just  i i i don't see it
a       a     a      a   fb
```

Table 19: *Energy features.*

Feature Name	Description
utt_nrg_mean	mean RMS energy in utterance
abs_nrg_diff	difference between mean RMS energy of end and penultimate regions
end_nrg_mean	mean RMS energy in end region
norm_nrg_diff	normalized difference between mean RMS energy of end and penultimate regions
rel_nrg_diff	ratio of mean RMS energy of end and penultimate regions
snr_mean_utt	mean signal-to-noise ratio (CDF value) in utterance
snr_sd_utt	st dev of signal-to-noise ratio values (CDF values) in utterance
snr_diff_utt	difference between maximum SNR and minimum SNR in utterance
snr_min_utt	st dev of signal-to-noise ratio values (CDF values) in utterance
snr_max_utt	maximum signal-to-noise ratio values (CDF values) in utterance

```
victoria holt is that right
a                a    rb
```

4.7.5 Energy features

We included two types of energy features, as shown in Table 19. The first set of features was computed based on standard root mean square (rms) energy. Because our data were recorded from telephone handsets with various noise sources (background noise as well as channel noise) we also included a signal-to-noise ratio (SNR) feature to estimate the energy from the speaker. SNR values were calculated using the SRI recognizer with a Switchboard-adapted front-end (Neumeyer and Weintraub 1994; Neumeyer and Weintraub 1995). Values were calculated over the entire conversation side, and those extracted from regions of speech were used to find a cumulative distribution function (CDF) for the conversation. The frame-level SNR values were then represented by their CDF value in order to normalize the SNR values across speakers and conversations.

4.7.6 Speaking rate ("enrate") features

We were also interested in overall speaking rate. However we needed a measure that could be run directly on the signal. For this purpose, we experimented with a signal processing measure, "enrate", which is currently under development at ICSI by Nelson Morgan, Nikki Mirghafori and Eric Fosler. This measure runs directly on the speech signal, and has been shown to correlate moderately with lexical measures of speaking rate (Morgan *et al.* 1997). The measure can be run over the entire signal, but because it uses a large window, values are less meaningful if significant pause time is included in the window. Since our speakers were recorded continuously, we had long pause regions in our data (mainly times during which the other speaker was talking). Based on advice from the ICSI team, we applied the measure to certain stretches of speech of minimum duration without excessive pauses.

We calculated frame-level values over a two second speech interval. The enrate value was calculated for a 25ms frame window with a window step hop of 0.1 second. Output values were calculated for 10ms frames to correspond to other measurements. We included pauses of less than 1 second and ignored speech regions of less than one second, where pause locations were determined as described earlier.

If the end of a speech segment was approaching, meaning that the 2 second window could not be filled, no values were written out. The enrate values corresponding to particular utterances were then extracted from the conversation side values. This way, if utterances were adjacent, information from surrounding speech regions could be used to get enrate values for the beginnings and ends of utterances which normally would not fill the 2 second speech window. Features computed for use in tree-building are listed in Table 20.

4.7.7 Gender features

We also included gender features. This was not a main focus of our study, however it was a good idea to include it as a check on our F0 normalizations, we included the gender of the speaker. It is also possible, however, that features

Table 20: *Speaking rate features.*

Feature Name	Description
mean_enr_utt	mean of enrate values in utterance
mean_enr_utt_norm	mean_enr_utt normalized by mean enrate in conversation-side
stdev_enr_utt	st dev of enrate values in utterance
min_enr_utt	minimum enrate value in utterance
max_enr_utt	maximum enrate value in utterance

could be used differently by men and women, even after appropriate normalization for pitch range differences. We also included the gender of the listener to check for a possible sociolinguistic interaction between the ways in which speakers employ different prosodic features and the conversational dyad

4.7.8 Decision Tree Classifiers

For our prosodic classifiers, we used CART-style decision trees (Breiman *et al.* 1983). Decision trees allow combination of discrete and continuous features, and can be inspected to gain an understanding of the role of different features and feature combinations.

We downsampled our data to obtain an equal number of datapoints in each class. Although a drawback to downsampling is a loss of power in the analyses due to fewer datapoints, downsampling was warranted for two important reasons. First, as noted earlier, the distribution of frequencies for our DA classes was severely skewed. Because decision trees split according to an entropy criterion, large differences in class sizes wash out any effect of the features themselves, causing the tree not to split. By downsampling to equal class priors we assure maximum sensitivity to the features. A second motivation for downsampling was that by training our classifiers on a uniform distribution of DAs, we facilitated integration with other knowledge sources (see section on Integration).

After finishing expanding the tree with questions, the tree-growing algorithm used a ten-fold cross-validation procedure to avoid overfitting the training data. Leaf nodes were successively pruned if they failed to reduce the entropy in the cross-validation procedure.

We report tree performance using two metrics, accuracy and efficiency. Accuracy is the number of correct classifications divided by the total number of samples. Accuracy is based on hard decisions; the classification is that class with the highest probability. Because we downsample to equal class priors, the chance performance for any tree with N classes is 100/N%. For any particular accuracy level, there is a trade-off between recall and false alarms. In the real world there may well be different costs to a false positive versus a false negative in detecting a particular utterance type. In the absence of any model of how such costs would be assigned for our data, we report results assuming equal costs to these errors for our downsampled trees.

Efficiency measures the relative reduction in entropy from the root node to the final tree, taking into account the probability distributions produced by the tree. Two trees may have the same classification accuracy, but the tree which more closely approximates the probability distributions of the data (even if there is no effect on decisions) has higher efficiency (lower entropy). Although accuracy and efficiency are typically correlated, the relationship between the measures is not strictly monotonic since efficiency looks at probability distributions and accuracy looks only at decisions.

4.7.9 Prosodic DA Detection: Results and Discussion

We first examine results of the prosodic model for a seven-way classification involving all DAs. We then look to results from a words-only analysis, to discover potential subtasks for which prosody could be particularly helpful. The analysis reveals that even if correct words are available, certain DAs tend to be misclassified. We examine the potential role of prosody for three such subtasks: (1) the classification of questions; (2) the classification of agreements; and (3) the classification of incomplete utterances. In all analyses we seek to understand the relative importance of different features and feature types, as well as to determine whether integrating prosodic information with a language model can improve classification performance overall.

Table 21: *Feature type usage in seven-way classification.*

Feature Type	Usage (%)
Dur	0.554
F0	0.126
Pau	0.121
Nrg	0.104
Enr	0.094

4.7.10 Seven-Way Classification

We applied the prosodic model first to a seven-way classification task for the full set of DAs: Statements, Questions, Incomplete utterances, Backchannels, Agreements, Appreciations, and Other. Note that "Other" is a catch-all class representing a large number of heterogeneous DAs that occurred infrequently in our data. Therefore we do not expect this class to have consistent features, but rather to be distinguished to some extent based on feature consistencies within the other six classes. As described in the Method section, data were downsampled to equal class sizes to avoid confounding results information from prior frequencies of each class. section). Because there are seven classes, chance accuracy for this task is 100/7% or 14.3%. For simplicity, we assumed equal cost to all decision errors, i.e. to all possible confusions among the seven classes.

A tree built using the full database of features described earlier allows a classification accuracy of 41.15%. This gain in accuracy is highly significant by a binomial test; $p < .0001$. If we are interested in probability distributions rather than decisions, we can look at the efficiency of the tree, or the relative reduction in entropy from the root node of the tree. By using the all-features prosodic tree for this seven-way classification, we reduce the number of bits necessary to describe the class of each datapoint by 16.8%.

The all-features tree is large (52 leaves), making it difficult to interpret the tree directly. It is helpful however to summarize the tree. We do this by reporting a measure of "feature usage". The usage measure reflects the number of times a feature was queried in classifying the datapoints. The measure thus accounts for the position of the feature in the tree; features further up in the tree have higher counts than those lower in the since there are more datapoints at the higher nodes. The measure is normalized to sum to 1.0 for each tree. Table 21 lists usage by feature type.

Table 21 indicates that when all features are available, a duration-related feature is used in more than half of the queries. Gender features are not used at all; this supports the earlier hypothesis that, as long as features are appropriately normalized, it is reasonable to create gender-independent prosodic models for these data. Individual feature usage, as shown in Table 22 reveals that the raw duration feature (ling_dur), which is a "free" measure in our work since we assumed locations of utterance boundaries—accounted for only 14% of the queries in the tree; the remaining portion of the 55% accounted for by duration features were those associated with the computation of F0- and pause-related information. Thus the power of duration for the seven-way classification comes largely from measures involving computation of other prosodic features. The most-queried feature, regr_num_frames (the number of frames used in computing the F0 regression line) may be better than other duration measures at capturing actual speech portions (as opposed to silence or nonspeech sounds), and may be better than other F0-constrained duration measures (e.g. f0_num_good_utt) due to a more robust smoothing algorithm. We can also note that the high overall rate of F0 feature given in Table 22 represents a summation over many different individual features.

Since we were also interested in feature importance, a number of individual trees were built using the leave-one-out method, in which the feature list is systematically modified and a new tree is built for each subset of allowable features. It was not feasible to leave out individual features because of the large set of features used; we therefore left out groups of features corresponding to the feature types as defined in Table 22. We also included a matched set of "leave-one-in" trees for each of the feature types (i.e. trees for which all *other* feature types were removed), and a single leave-two-in tree, built *post hoc* which made available on the two most feature types with highest accuracy from the leave-one-in analyses. Note that the defined feature lists specify the features *available* for use in building a particular prosodic model; whether or not features are *actually* used requires inspection of the resulting tree. Figure 7 shows results for the set of leave-one-out and leave-one-in trees, with the all-features tree provided for comparison purposes. The upper graph indicates accuracy values; the lower graph shows efficiency values. Each bar indicates a

Table 22: *Feature usage for seven-way (all DAs) classification.*

Feature Type	Feature	Usage (%)
Dur	regr_num_frames	0.180
Dur	ling_dur	0.141
Pau	total_count_enr_utt_n	0.121
Enr	stdev_enr_utt	0.081
Enr	ling_dur_minus_min10pause	0.077
Enr	total_count_enr_utt	0.073
Nrg	snr_max_utt	0.049
Nrg	snr_mean_utt	0.043
Dur	regr_dur	0.041
F0	f0_mean_zcv	0.036
F0	f0_mean_n	0.027
Dur	f0_num_good_utt	0.021
Dur	f0_num_utt	0.019
F0	norm_end_f0_mean	0.017
F0	numacc_n_rdur	0.016
F0	f0_sd_good_utt	0.015
Enr	mean_enr_utt	0.009
F0	f0_max_n	0.006
Nrg	snr_sd_utt	0.006
Nrg	rel_nrg_diff	0.005
Enr	mean_enr_utt_norm	0.004
F0	regr_start_f0	0.003
F0	finalb_amp	0.003

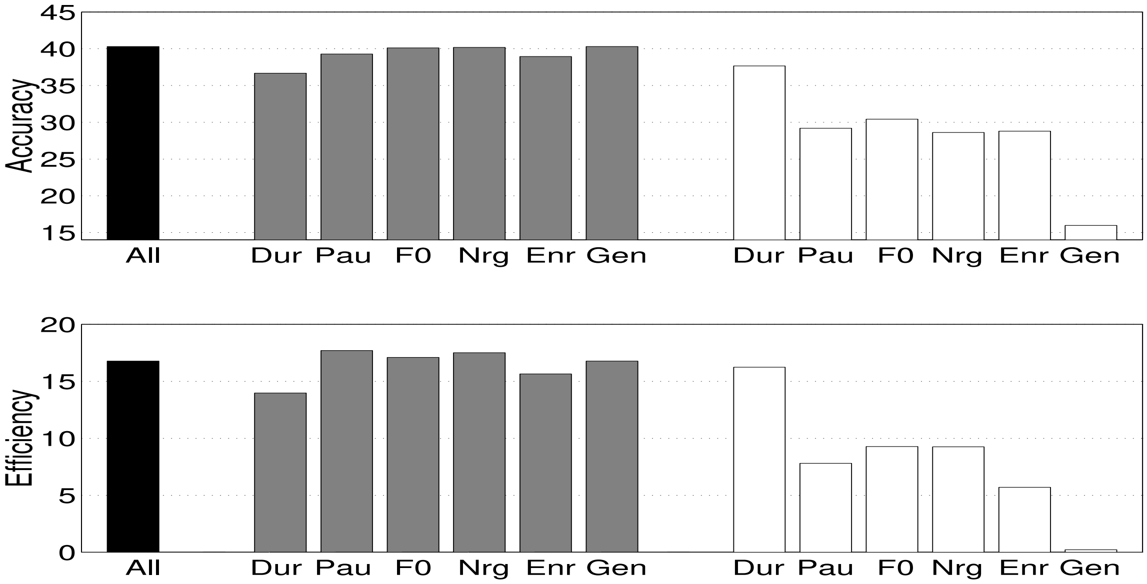

Figure 7: Performance of prosodic trees using different feature sets for the classification of all seven DAs (Statements, Questions, Incomplete Utterances, Backchannels, Agreements, Appreciations, Other). N for each class=391. Chance accuracy = 14.3%. Gray bars=exclude feature type; white bars=include only feature type. Dur=Duration, Pau=Pause, F0=Fundamental frequency, Nrg=Energy, Enr=Enrate (speaking rate), Gen=Gender features.

separate tree.

We first tested whether there was any significant loss in leaving out a feature type, by doing pairwise comparisons between the all-features tree and each of the leave-one-out trees.[4] Although trees with more features to choose from typically perform better than those with fewer features, additional features can hurt performance. The greedy algorithm used cannot look ahead to determine the optimal overall model, but rather seeks to maximize entropy reduction locally at each split. This limitation of decision trees is another motivation for conducting the leave-one-out analyses. Since cannot predict the direction of change for different feature sets, comparison on tree results are conducted using two-tailed tests.

Results showed that the only two feature types whose removal caused a significant reduction in accuracy were duration ($p < 0.0001$) and enrate ($p < 0.05$). The enrate-only tree however yields accuracies on par with other feature types whose removal did not affect overall performance; this suggests that the contribution of enrate in the overall tree may be through interactions with other features. All of the leave-one-in trees were significantly less accurate than the all-features tree; although the tree using only duration achieved an accuracy of, it was still significantly less accurate than the all-features tree by a Sign test ($p < 0.01$). Adding F0 features (the next best feature set in the leave-one-in trees) did not significantly improve accuracy over the duration-only tree alone, suggesting that for this task the two feature types are highly correlated. Nevertheless, each of the leave-one-in trees, all feature types except gender yielded accuracies significantly above chance by a binomial test ($p < .0001$ for the first five trees). The gender-only tree was slightly better than chance by either a one- or a two-tailed test,[5] however this was most likely due to a difference in gender representation across classes.

Taken together, these results suggest that there is considerable redundancy in the features for DA classification, since removing one feature type at a time (other than duration) makes little difference to accuracy. Results also suggest however that features are not perfectly correlated; there must be considerable interaction among features in classifying DAs, because trees using only individual feature types are significantly less accurate than the all-features tree.

[4]To test whether one tree (A) was significantly better than another (B), we counted the number of test instances on which A and B differed, and on how many A was correct but B was not; we then applied a Sign test to these counts.

[5]It is not clear here whether a one- or two-tailed test is more appropriate. Trees typically should not do worse than chance; however because they minimize entropy, and not accuracy, the accuracy can fall slightly below chance.

Table 23: *Accuracy for individual and combined models for seven-way classification.*

Knowledge Source	HLD Set true words	DEV Set true words	DEV Set N-best output
samples	2737	287	287
chance (%)	14.29	14.29	14.29
tree (%)	41.15	38.03	38.03
words (%)	67.61	70.3	58.77
words+tree (%)	69.98	71.14	60.12

Finally, duration is clearly of primary importance to this classification. This is not surprising, as the task involves a seven-way classification including longer utterances (such as statements) and very brief ones (such as backchannels like "uh-huh"). Two questions of further interest regarding duration, however, are: (1) will a prosody model that uses mostly duration add anything to a language model (in which duration is implicitly encoded); and (2) is duration useful for other tasks involving classification of DAs similar in length. Both questions are address in the following sections.

As just discussed, the all-features tree (as well as others including only subsets of feature types) provide significant information for the seven-way classification task. Thus if one were only to use prosodic information (no words or context), this is the level of performance resulting for the case of equal class frequencies. To explore whether the prosodic information could be of use when lexical information is also available, we integrated the tree probabilities with likelihoods from our DA-specific trigram language models built from the same data. For simplicity, integration results are reported only for the all-features tree in this and all further analyses, although as noted earlier this is not guaranteed to be the optimal tree.

Since our trees were trained with uniform class priors, we combined tree probabilities $P(U|F)$ with the word-based likelihoods $P(W|U)$ linearly, as described in sections 4.8 and 4.9, using a weighting factor found by optimizing on held out data. The integration was performed separately for each of our two test sets (HLD and DEV), and within the DEV set for both transcribed and recognized words. Results are shown in Table 23. Classification performance is shown for each of the individual classifiers, as well as for the optimized combined classifier.

As shown, for all three analyses, adding information from the tree to the words model improved classification accuracy. Although the gain appears modest in absolute terms, for the HLD test set was highly significant by a Sign test,[6] $p < .001$. For the smaller DEV test set, the improvements did not reach significance; however the pattern of results suggests that this is likely to be due to a lack of power due to the small sample size. It is also the case that the tree model does not perform as well for the DEV as the HLD set; this is not attributable to small sample size, but rather to a mismatch between the DEV set and the training data involving how data were segmented, as noted in the Method section. The mismatch in particular affects duration features, which were important in this analyses as discussed earlier. Nevertheless, word-model results are lower for N-best than for true words in the DEV data while by definition the tree results stay the same. We see that accordingly, integration provides a larger win for the recognized than the true words. Thus we would expect results for recognized words for the HLD set (if they could be obtained) should show an even larger win than the win observed for the true words in that set.

These results provide an answer to one of the questions posed in the previous section: does prosody provide an advantage over words if the prosody model uses mainly duration? The results indicate that the answer is yes. Although the number of words in an utterance is highly correlated with duration, and word counts are represented implicitly by the probability of the end-of-utterance marker in a language model, a duration-based tree model still provides added benefit over words alone. One reason may be that duration (reflected by the various features we included) is simply a better predictor of DA than is word count. Another independent possibility is that the advantage from the tree model comes from its ability to directly and iteratively threshold the data.

4.7.11 DA Confusions Based on Word Information

Next we explored additional tasks for which prosody could aid DA classification. Since our trees allow N-ary classification, the logical search space of possible tasks was too large to explore systematically. We therefore looked to

[6] One-tailed, because model integration assures no loss in accuracy.

the language model to guide us in identifying particular tasks of interest. Specifically, we were interested in DAs that tended to be misclassified even given knowledge of the true words. We therefore examined the pattern of confusions made when our seven DAs were classified using the language model alone. Results are shown in Figure 8. Each subplot represents the data for one actual DA.[7] Bars reflect the normalized rate at which the actual DA was classified as each of the seven possible DAs, in each of the the three test conditions (HLD, DEV-true, and DEV-Nbest).

As shown, classification is excellent for the statement class, with few misclassifications even when only the recognized words are used.[8] For the remaining DAs however, misclassifications occur at considerable rates. Classification of questions is a case in point: even using true words, questions are often misclassified as statements (but not vice versa), and this pattern is exaggerated when testing on recognized as opposed to true words. The asymmetry is partially attributable to the presence of declarative questions. The effect associated with recognized words appears to reflect a high rate of missed initial "do" in our recognition output, as discovered in independent error analyses (see §7). For both statements and questions however, there is little misclassification involving the remaining classes. This probably reflects the length distinction as well as the fact that most of the propositional content in our corpus occurred in statements and questions, whereas other DAs generally served to manage the communication—a distinction likely to be reflected in the words. Thus, our first subtask will be to examine the role of prosody in the classification of statements and questions.

A second problem visible in Figure 8 is the classification of incomplete utterances. Even using true words, classification of these DAs is at only 75.0% accuracy. Knowing whether or not a DA is complete would be particularly useful for both language modeling and understanding. Since the misclassifications are distributed over the set of DAs, and since logically any DA can have an incomplete counterpart, our second subtask will be to classify a DA as either incomplete or not-incomplete (all other DAs).

A third notable pattern of confusions involves backchannels and explicit agreements. This is not surprising, since the two classes share words such as "yeah" and "right". In this case, the confusions are considerable in both directions, but more marked for the case of agreements. As mentioned in the method section, some of these cases may involve utterances that were mislabeled because labelers used only the transcripts. However, for any mislabeled cases we would expect no improvement by adding prosody, since we would also need to match the (incorrect) transcriber labels. Thus any gain from prosody would be likely to reflect a contribution for correctly labeled cases; we will therefore examine backchannels and agreements as our third classification subtask.

4.7.12 Subtasks

In the three subtasks, we applied a similar analysis, looking at feature usage by selectively removing features and assessing the loss in performance. Results are described in detail in Shriberg *et al.* (1998); we briefly the general findings below.

Feature analyses revealed that the relative importance of different features depended critically on the task. For the classification of statements and questions we found that duration, pause, and F0 features were heavily used. Furthermore, when we broke our question class down into yes-no questions, wh-questions and declarative questions, and ran a four-way classification along with statements, results showed primary importance of F0 information. Results, as shown in Figure 9, were in good accord with the literature on question intonation, which predicts strongest final rises for yes-no and declarative questions and an absence of a final rise for wh-questions, as well as a higher overall F0 for questions than statements. Final rises are captured in our tree by the features end_grad, norm_f0_diff, and utt_grad; overall or average F0 is captured by f0_mean_zcv.

For incomplete utterances on the other hand, energy features, which were not useful for the seven-way or the question classification, were of particular importance. Typically, utterances fall to a low energy value when close to completion. However when speakers stop mid-stream, this fall has not yet occurred, and thus the energy stays unusually high. Our prosodic trees pick up on this feature, as well as others such as duration, to classify utterances as either finished or incomplete. Finally, for the classification of backchannels and agreements, we found that duration, pause, and energy features played a strong role. Thus, although DAs are redundantly signaled by prosodic features, the degree to which different features are important depends on which DAs one is classifying. For this reason, for optimal coverage it is best to include a variety of features in the prosodic model.

[7] Due to the heterogeneous makeup of the "other" DA class *per se*, we were not particularly interested in its pattern of confusions and hence the graph for that data is not shown.

[8] The high classification rate for statements by word information was a prime motivation for downsampling our data in order to examine the inherent contribution of prosody, since as noted in the Method section, statements make up most of the data in this corpus.

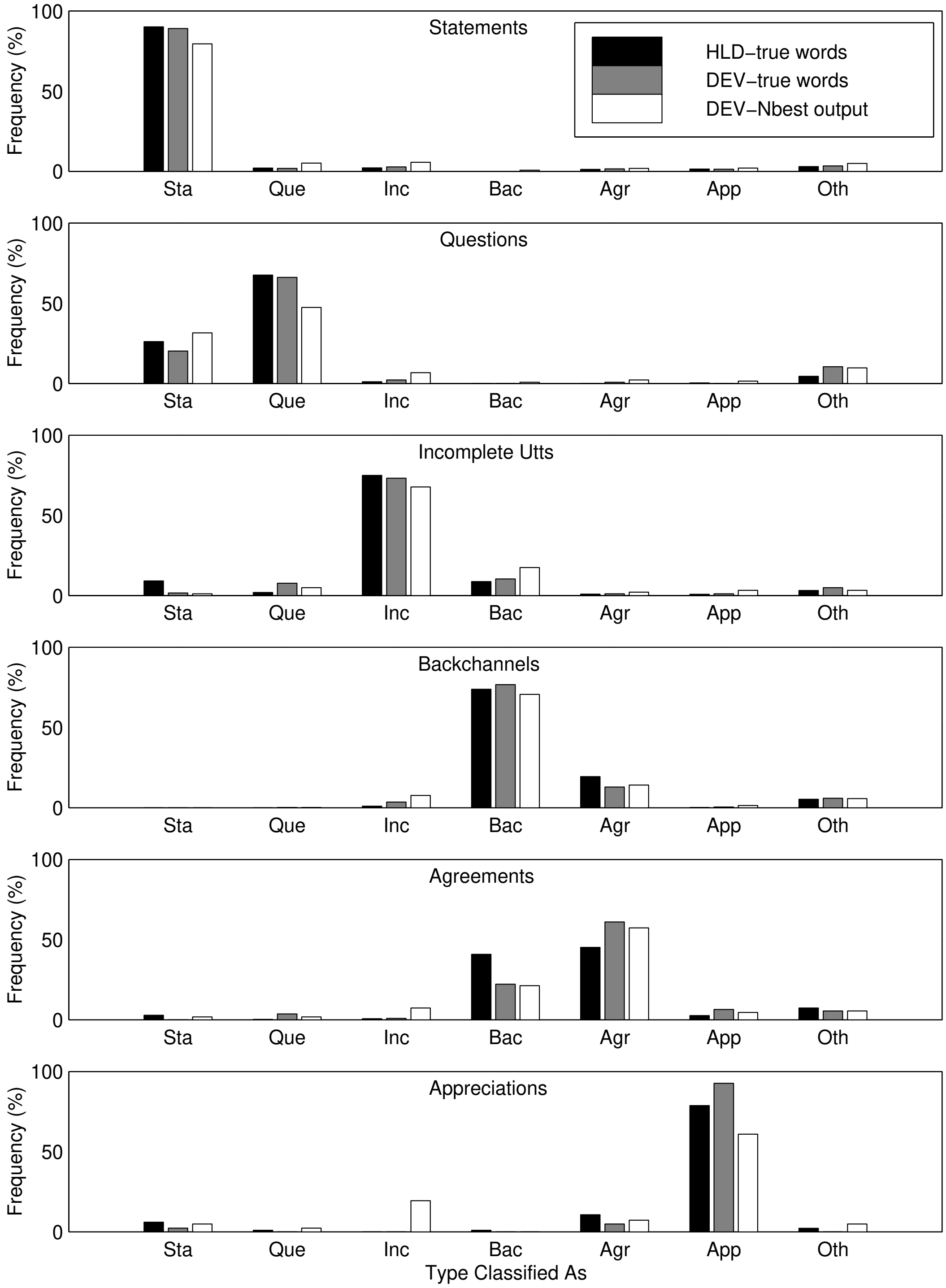

Figure 8: Classification of DAs based on word trigrams only, using three different test sets.

Figure 9: Decision tree for the classification of statements (S), yes-no questions (QY), wh-questions (QW), and declarative questions (QD).

Second, for each subtask we integrated the prosodic model with our DA-specific language model. The results are shown in Table 24.

As shown, the prosody alone did quite well, significantly better than chance in all cases ($p < .0001$ by a binomial test for all trees). For the question detection task, the tree actually does better than the recognized words; for the agreement task it does better than the true words in the HLD set. Integration yielded consistent improvement over the words alone, i.e. for all three tasks and all three test sets, accuracy improved by adding prosody. For the incomplete utterance task, the gain was not significant using true words, but nearly reached significance for recognized words in the much smaller DEV set. Additional analysis revealed that better performance can be achieved if the incomplete class is split into two classes based on turn information (which is available from our dialog grammar). The incomplete class combines both self-cutoffs and other-interruptions, which a separate tree showed are distinguishable prosodically at high accuracy (81%, where chance is 50%). For the question and agreement tasks, Sign tests run for the larger HLD set showed a highly significant gain in accuracy by adding prosody, $p < .001$ and $p < .00001$, respectively.

4.8 Combining decision trees with discourse grammars

We now turn to detection results based on prosodic features that incorporate discourse grammars, analogous to those reported for word-based detection. The HMM framework requires that we compute prosodic likelihoods of the form $P(F_i|U_i)$ for each utterance U_i and associated prosodic features values F_i. We have the apparent difficulty that decision trees give estimates for the posterior probabilities, $P(U_i|F_i)$.

The problem can be overcome by applying Bayes' Rule locally:

$$P(F_i|U_i) = P(F_i)\frac{P(U_i|F_i)}{P(U_i)} \propto \frac{P(U_i|F_i)}{P(U_i)}$$

A quantity proportional to the required likelihood can therefore be obtained by either dividing the posterior tree probability by the prior $P(U_i)$, or by training the tree on a uniform prior distribution of DA types. We chose the second approach.

Table 24: *Accuracy for individual and combined models for three subtasks.*

Knowledge Source	HLD Set true words	DEV Set true words	DEV Set N-best output
Questions and Statements			
samples	1852	266	266
chance (%)	50.00	50.00	50.00
tree (%)	74.21	75.97	75.97
words (%)	83.65	85.85	75.43
words+tree (%)	85.64	87.58	79.76
Incomplete and Completed			
samples	2646	366	366
chance (%)	50.00	50.00	50.00
tree (%)	72.16	72.01	72.01
words (%)	88.44	89.91	82.38
words+tree (%)	88.74	90.49	84.56
Agreements and Backchannels			
samples	2520	214	214
chance (%)	50.00	50.00	50.00
tree (%)	68.77	72.88	72.88
words (%)	68.63	80.99	78.22
words+tree (%)	76.90	84.74	81.70

Table 25: *DA detection using prosody.*

Discourse Grammar	Accuracy (%)
None	38.9
Unigram	48.3
Bigram	50.2

Our experiments in this area were limited, and results should be considered preliminary. A single tree was trained to discriminate among the five most frequent dialog acts (Statements, Questions, Backchannels, Agreements, Abandoned), with all others lumped together in an "Other" category. The probability in the "Other" category was split uniformly among all the types in that category. Results are shown in Table 25

4.9 Dialog act detection using multiple knowledge sources

Finally, we wanted to combine evidence from recognized words and prosody for DA detection, since, as noted earlier, we can expect them to give partially complementary information, and prosody might help alleviate the effects of unreliable word recognition.

Combining recognized words and prosody amounts to estimating a combined likelihood $P(A_i, F_i|U_i)$ for each utterance. The simplest approach is to assume the two types of acoustic observations (recognizer acoustics and prosodic features) are approximately conditionally independent once U_i is given:

$$\begin{aligned} P(A_i, F_i|U_i) &= P(A_i|U_i)P(F_i|A_i, U_i) \\ &\approx P(A_i|U_i)P(F_i|U_i) \end{aligned}$$

Since the recognizer acoustics are modeled through their dependence on words, it is particularly important to avoid using prosodic features that are directly correlated with word-identities, or features that are also modeled by the discourse grammars, such as utterance position relative to turn changes.

For the one experiment we conducted using this approach, we combined the acoustic n-best likelihoods from Section 4.5 with the Top-5 tree classifier from Section 4.8. Since these represent very different types of models we had to optimize both the relative weighting between the likelihood models, as well as the weighting of the likelihoods overall against the discourse grammar. Results are summarized in Table 26.

Table 26: *Combined utterance detection accuracies.*

Discourse Grammar	Accuracy (%)		
	Prosody only	Recog. Words only	Combined
None	38.9	42.8	56.5
Unigram	48.3	61.9	62.6
Bigram	50.2	64.6	**65.0**

As shown, the combined classifier presents an improvement over the recognizer-based classifier. The experiment without discourse grammar indicates that the combined evidence is considerably stronger than either knowledge source alone, yet this improvement seems to be made largely redundant by the use of the discourse grammar. As noted earlier, these results are preliminary in that no significant tuning of the decision tree component has been done.

5 SWBD RECOGNITION

We applied our utterance detection algorithm to the Switchboard word-recognition task by using a mixture of the 42 dialog-act-specific language models to rescore each test-set utterance, and using the combined detector to set the mixture weights. As a result, we had a promising but not statistically significant movement in word error from 41.2% to 40.9%.

Before describing our experiments we first describe the cheating experiment that we ran at the beginning of the project.

5.1 Cheating Experiment 1: Perplexity

How much can we expect discourse information to reduce word error rate on Switchboard? In order to give an upper bound for our discourse LM experiments, we performed two **cheating** experiments. In these we asked the question: "Suppose we had an oracle give us perfect knowledge of an utterance's correct dialog act tag, could we use this to reduce WER?".

In our first experiment, we built 42 separate trigram language models, one for each DA. Each was based on a standard backoff trigram with Witten-Bell discounting (Witten and Bell 1991). We built two versions of each language model; one was trained solely on the examples of each DA (the 'DLM'); the other was trained on each DA and also interpolated with an LM trained on the entire 166,000-utterance training set (the 'baseline' LM or BLM), using deleted interpolation.

The baseline language model was a standard trigram trained on 1.8 million words from Switchboard. Since earlier WS95 work had shown that N-gram LMs based on complete utterance units give lower perplexity than those trained on acoustic segments (Rosenfeld *et al.* 1996), this language model included a token marking utterance boundaries. However only 1.4 million of the 1.8 million SWBD words (i.e. the WS97-TRAIN 'linguistically segmented' corpus) had hand-coded utterance boundaries. Stolcke (1997) trained an automatic linguistic segmenter (Stolcke and Shriberg 1996) on the hand-segmented 1.4 million words, and used it to segment the remaining training data. The hand-segmented and the automatically-segmented training data were pooled, and our baseline language model (BLM) was trained on these 1.8 million words.

The deleted interpolation algorithm trains a separate interpolation coefficient λ_j for each of the 42 DA_j's. Computing the prior probability of the word sequence W for an utterance j with dialog act U_j for the interpolated model:

$$P(W|U_j) = \prod_i (\lambda_j P_{ULM_j}(w_i|w_{i-2}, w_{i-1}) + (1-\lambda_j) P_{BLM}(w_i|w_{i-2}, w_{i-1}))$$

The first column of Table 31 shows the λ values for the 42 dialog acts. Recall that the λ value is the weight for the DA itself; (1-λ) is the weight for the other utterances. Thus a very high λ indicated that the LM for the DA was significantly different from the rest of the utterances. A very low lambda probably indicates that the DA looks a lot like the average utterance but just has insufficient data to train on. Thus dialog acts like **nn (Answer-No)**, **b (Backchannel)**, **ny (Answer-Yes)**, **bk (Response-Acknowledgement)**, **bh (Question-Backchannel)**, and **ft (Thanks)** have high lambdas indicating LMs which are quite distinct from the other dialog acts.

We compared these 2 LMs with the BLM by computing the perplexity on the 31,000-utterance held-out set.

Table 27: *Using Oracle to select 42 language models: Perplexity on WS97 DevTest.*

LM	Perplexity
Baseline (all data)	76.8
DA-specific, (interpolated)	66.8

The three perplexity columns of Table 28 shows that training 42 separate trigrams, one on each DA, produced a slight perplexity reduction over homogeneous training on the entire training set Interpolating each of these DA-specific word-LMs with the entire training set resulted in a significant decrease in perplexity (to 67). The actual perplexity decrease was different for different dialog acts; see the perplexity columns of Table 31 shows the individual DA differences.

5.2 Cheating Experiment 2: Word Error Rate

In our second experiment, we extended these perplexity results to actual word error. We used the same 42 separate backoff-trigram LMs trained on the 197,000 utterance WS97-TRAIN train + held-out set, each interpolated with the BLM, and tested on the 4,000 utterance dev set. We then rescored the lattices, which had a baseline word error of 41.2%. Table 28 shows that the overall reduction in word error was small, 0.9% absolute. The reduction was statistically highly significant under a matched-pairs Sign test ($p < .0001$). The word error reduction due to specific DAs is shown in the "Word Accuracy" columns of Table 31.

Table 28: *Using Oracle to select 42 language models: WER on WS97 DevTest.*

LM	WER (%)
Baseline (all data)	41.2
DA-specific, with interpolation	40.3

Why was the word error reduction so small? Figure 10 shows that while about 50% of the utterances are Statements, a full 83% of the *words* in the DevTest were in Statements.

Our error reduction was mostly in various kinds of questions and backchannels, which among them don't contain a very large proportion of the SWBD words. The final column of Table 31 shows the actual number of word errors that were reduced by using the interpolated utterance-specific LMs.

5.3 Conclusions from Cheating Experiments

The results of the cheating experiments are mixed. On the positive side, switching among the 42 DA-specific LMs produced a significant perplexity decrease, from 76 to 67. But this perplexity decrease is not matched by an equivalent decrease in word error; the word error only decrease from 41.2% to 40.3%.

The cheating experiment shows that even perfect knowledge of the dialog acts can only be expected to give about a 1 percent reduction in word error. This is mainly due to the fact mentioned above that **Statement** (non-opinion and opinion combined) account for 83% of the words in our corpus (since e.g. backchannels and answers tend to be short). Table 29, summarized from our Cheating Experiments described above, however, shows that using utterance-specific

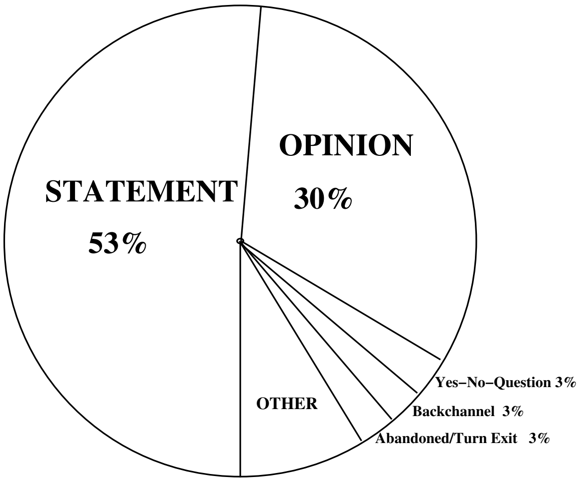

Figure 10: Number of words in various DevTest DAs.

Table 29: *Cheating error rates on specific dialog acts.*

Dialog Act	WER	Oracle WER	Improvement with Oracle(%)
Answer No	29.4	11.8	-17.6
Backchannel	25.9	18.6	-7.3
Backchannel Questions	15.2	9.1	-6.1
Abandoned & Turn-Exit	48.9	45.2	-3.7
Wh-Questions	38.4	34.9	-3.5
Yes-No-Questions	55.5	52.3	-3.2
Statement	42.0	41.5	-0.5

language models can significantly improve WER for some dialog acts, and hence this approach could prove useful for tasks with a different distribution of dialog acts.

Using the utterance-specific models does have a strong effect on the dialog acts which differ the most from statements (i.e. those whose λ values are close to 1.0; see the first column of Table 31). Thus word error was significantly reduced on many kinds of backchannels, yes/no answers, and questions.

These results suggested two directions that we followed in our research. First, that we focus our attention on detection and LM-improvement of these dialog acts (in particular yes-no questions and backchannels). We predicted at this point that this would not not result in a significant decrease in SWBD word error (since SWBD has very few questions, and while it has many backchannels, they tend to be one or two words). And indeed it did not. But focusing on questions and backchannels should have a number of other benefits. Questions will occur in other corpora and tasks (like the Map Task, or various SLU database retrieval tasks) in much higher percentage than in SWBD. Being able to distinguish a question from a non-question ('Question Detection') is an important and relevant task independent of its impact on word-error. And better detection of backchannels may help improve our automatic turn segmentation algorithms.

Second, since we now know that our model has very little effect on the Statement (sd and sv) dialog acts which comprise 83% of the words in the devtest set, we clearly need to find ways to divide the statements into subclasses. §6.2 summarizes our experiments which check if Statements pattern differently if they are embedded in a series of statements by the same speaker, versus being surrounded by turn changes. While our results suggest that the statements in single-statement-turns are much shorter (have fewer words) than the statements in multi-statement turns, we found no significant improvement in utterance detection accuracy by splitting the statements in this way.

5.4 Baseline Error Analysis (DA-specific error with Baseline LM)

We trained a standard language model on the entire 197,000 utterance training set, ran recognition with this LM on the 4,000 utterance WS97-dev-test set, and computed the error rate for each discourse tag. The following chart shows the word accuracy for each discourse tag that occurs with frequency 7 or more in the dev-test.

Table 30: *Word accuracy on background (non-DA-specific) recognizer reported by dialog act.*

Tag	Count	Word Accuracy (%)
br	8	32.0
qh	7	40.6
bf	8	42.6
qy	61	42.6
ba	40	44.1
^2	9	46.4
b^m	13	46.7
qy^d	24	47.1
o	12	48.4
ad	19	52.4
sd	871	57.0
fc	9	57.1
sv	425	59.1
qw	36	59.4
qo	14	62.5
nn	12	64.7
bk	17	66.7
na	7	66.7
bh	16	68.6
ny	57	70.4
b	420	73.4
h	15	75.4
aa	106	76.6

The average error for the discourse tags in Table 30 is 55%. Many kinds of questions were significantly worse. For example yes-no questions, declarative yes-no-questions, and rhetorical questions were all between 42% and 47% word error. This suggested that questions would be worth focusing on.

5.5 Non-Cheating SWBD Recognition Experiments

Although the word error reduction in the cheating experiment was very small, we wanted to verify how much of that reduction could be realized in a fair recognition experiment, i.e., using automatic DA detection.

The experiment consisted of two phases. In the first phase, posterior DA probabilities $P(U_i|A)$ were computed, using the N-best based DA detector (Section 4.5). These probabilities were then used in a sentence-level mixture (Iyer *et al.* 1994) of the 42 DA-specific (smoothed) language models, such that the mixture weight for each LM corresponded to the posterior probability of the corresponding DA. The 2500-best lists were thus rescored with a combined language model of the following form:[9]

$$P(W_i) = \sum_{U_i} P(U_i|A)P(P(W_i|U_i)$$

Note that while the N-best-based DA detector had an accuracy of 64.6%, the mixture model should be relatively robust to detection errors as long as the correct DA is among those with highest posterior probability.

[9] Rescoring was done on N-best lists, not lattices, since the computation of the mixture LMs requires keeping full word histories.

Table 31: *Summary table: Error analysis by dialog act.*

Tag	Lambda	Perplexity			Word Accuracy			#errors
		baseline	cheating	cheating win	baseline	cheating	cheating win	corrected in cheating
b	0.981	4.9	2.8	-42.9	25.9	18.6	-7.3	34
bk	0.978	9.7	5.1	-47.5	33.3	25.0	-8.3	2
nn	0.970	11.8	1.5	-87.1	29.4	11.8	-17.6	3
ny	0.946	4.5	2.1	-54.7	26.5	25.0	-1.5	1
bh	0.934	11.5	2.7	-76.5	15.2	9.1	-6.1	2
fe.ba	0.869	31.3	11.5	-63.2	45.5	43.8	-1.7	1
aa	0.826	13.7	8.6	-36.9	21.6	23.0	1.4	-4
%	0.816	27.3	17.4	-36.4	48.9	45.2	-3.7	17
ft	0.751	15.2	1.7	-88.5	0.0	0.0	0	0
h	0.747	11.6	7.4	-36.5	16.1	12.9	-3.2	2
fp	0.729	91.4	28.0	-69.3	90.0	70.0	-20.0	1
qo	0.665	23.2	11.1	-52.0	35.9	32.8	-3.1	3
fc	0.662	107.2	64.1	-40.2	38.1	38.1	0	0
qw	0.651	65.5	43.3	-33.8	38.4	34.9	-3.5	9
sd	0.635	103.1	98.8	-4.2	42.0	41.5	-0.5	48
ˆh	0.598	28.5	25.9	-9.1	53.2	48.9	-4.3	3
fx.sv	0.557	95.7	88.6	-7.5	40.8	40.4	-0.4	11
br	0.524	44.4	21.0	-52.5	60.0	48.0	-12.0	3
fa	0.510	95.3	11.0	-88.4	80.0	80.0	0	0
qr.qy	0.470	80.3	60.0	-25.3	55.5	52.3	-3.2	17
ad	0.444	138.5	117.4	-15.2	48.7	50.8	2.1	-2
ar	0.443	103.0	69.0	-33.0	20.0	20.0	0	0
bˆm	0.413	125.5	94.7	-24.5	56.7	50.0	-6.7	3
nnˆe.ng	0.376	83.0	62.6	-24.5	29.2	29.2	0	-1
nyˆe.na	0.316	56.8	52.9	-6.8	30.0	30.0	0	0
ˆ2	0.301	305.6	292.3	-4.3	67.9	53.6	-14.3	4
qh	0.296	59.2	46.8	-20.9	54.7	56.3	1.6	0
qyˆd	0.285	118.6	104.9	-11.5	45.3	43.1	-2.2	5
no	0.256	55.6	50.4	-9.4	20.0	20.0	0	0
ˆq	0.245	88.4	82.8	-6.3	41.5	36.6	-4.9	2
bf	0.226	145.1	127.7	-12.0	57.4	63.8	6.4	-3
aap.am	0.226	71.0	44.2	-37.8	36.4	36.4	0	1
arp.nd	0.178	373.2	261.7	-29.9	63.6	81.8	18.2	-2

Table 32 shows both word error and perplexities obtained for the DA-conditioned mixture LM. Also shown are the results for the baseline LM, the cheating LM conditioned on the true DA labels, and rescoring with just the LM corresponding to the most likely DA (1-best LM).

Table 32: *Non-significant reduction in SWBD word error.*

Model	WER (%)	Perplexity
Baseline	41.2	76.8
1-best LM	41.0	69.3
Mixture LM	40.9	66.9
Cheating LM	40.3	66.8

In this fair comparison, WER is reduced by only 0.3% over the baseline, a non-significant change ($0.3 > p > 0.2$). What is encouraging is that the perplexity of the DA-conditioned mixture model is virtually the same as that of the cheating LM. The results for the 1-best approximation to the mixture LM are slightly worse.

Overall, while the WER improvement is not statistically significant, we find these results promising in that they indicate that we can approach the ideal (cheating) performance of a DA-conditioned LM using automatic DA detection techniques. Naturally, improving these techniques is expected to improve recognition results as well.

6 Other Experiments: What to do with Statements?

The cardinality and constitution of our 42 tag clusters were determined by our own intuition. In order to see if we could produce a better set of clusters with automatic or semi-automatic means, we ran a number of different clustering experiments. Since statements constituted the bulk of our data, we were particularly concerned with ways to revise the statements. The first experiment tested whether the sd/sv (Statement/Opinion) distinction was a helpful one. We then studied whether we should split the Statement category depending on its dialog act context (i.e. between backchannels, between other statements, etc). Finally, we tried splitting individual statement utterances into two ‘utterance pieces’ at the verb (creating a “utterance before the verb” and “utterance after the verb” piece), training separate language models on each piece.

6.1 Experiment: Are sv and sd different?

One important question we asked early on was how difficult would it be to discriminate between the statement (sd) and opinion-statement (sv) classes, since together they comprised 49% of the DA tokens, and were hard for humans to discriminate. A good measure of the difference between the two types is their cross-entropies. Two language models were trained, one only on utterances that were **Statements (sd)** and the other on utterances that were only **Statement-Opinion (fx.sv)**. Each model was presented with a data-set of only Statements, and a set of only Opinions. In Table 34 we see that there is a significant increase in entropy when a model is tested on data it wasn’t trained on, suggesting that there is indeed a difference in these utterances.

We did find that despite the differences in the Statement and Opinion data, there were enough similarities that it helped reduce perplexity to combine the training data of the two types. We assume this is because the extra data helped alleviate the data sparseness problem. Table 33 shows how perplexity was reduced by several attempts to find the ideal combination of training data to minimize test-set perplexity on Statements. The ideal result is an interpolation between the Statement data (0.6) and all other data (0.4), giving weight to the saying, “There’s no data like more data.”

6.2 Extensions: Above and Below the Statement

In the work described so far, the unit over which the algorithms operated is the sentence, or ”linguistic segment” as defined by the annotation of the SWBD corpus done by the LDC in 1996 (Meteer, et al 1995), and ”utterances” in the terminology of this document. However, especially in the case of statements which are frequent of often quite long, it is worth looking at the structure of the larger units, such as turns and conversations, and at smaller units, such as the

Table 33: *How well do sv and sd LM's model sv test data.*

Language Model Training	Test Set	Perplexity
sv's	sv's	101
sv's	sd's	163
sd's	sd's	102
sd's	sv's	114

Table 34: *What's the ideal set of training data for an sd model?*

Language Model Training	Test Set	Perplexity
sv's	sv's	101
all training-set	sv's	97
sv's and sd's	sv's	94
.6 * training-set sv's + .4 * ALL	sv's	89

beginning and end of the sentence. In the following sections we describe the work done at these two different levels of granularity.

6.2.1 Conversational Analysis

The labeling of such of large corpus of spontaneous conversational speech provides an opportunity to look at the structure of whole conversations and their subparts, for example, to begin work on characterizing what differentiates a narrative from an argument. The first step towards understanding these genres is to look at individual statements in their context. We began this effort by attempting to understand the relationship between the function of a statement, its place in the conversation and/or turn, and other features such as the length of the statement. Table xx shows the average length of statements based on what precedes or follows that statements.

The shortest statements are those that are between the other speaker's statements. These are essentially functioning as backchannels, where a speaker adds a short comment. The next group are answers to questions, which are on average 8 word statements. Note that since "yes" and "no" are labeled explicitly, these are only full statements that function as answers to questions. The next group, which are around 10 words on average, are in the middle of a turn and are followed by another statement or a backchannel by the other speaker. The longest statements are those between backchannels by the other speaker or before or after self statements of viewpoint.

Table 35: *Length of statements when surrounded by different types of utterances.*

Before		After	Length
Other statement	(s)	Other statement	4.33
Other question	(s)	Anything	8.06
Other backchannel	(s)	Self statement	10.42
Other statement	(s)	Other backchannel	10.50
Other backchannel	(s)	Self unfinished	10.57
Other backchannel	(s)	Other backchannel	12.94
Self unfinished	(s)	Other backchannel	13.07

6.2.2 Internal structure of statements

Statements, questions, and other utterance types that are full sentences (the is, have a subject and verb) are not uniform from beginning to end in the kinds of words and syntactic structures used. In discourse analysis, a common way to divide sentences is into two parts, ”given” and ”new” or ”theme” and ”rheme. Meteer and Iyer (1995) showed that dividing the sentence into two parts, before and after the main verb, and analyzing these two corpora separately showed a significant difference in the vocabulary used and the types and frequencies of disfluencies, and even the perplexity in a language model for speech recognition. The ”pivot point” dividing the sentence is the first strong verb (excluding verbs such as ”do”, ”be”, ”have” and ”seem”) or the last weak verb if there is not strong verb, as shown in the examples below:

```
- i've [PVT] voted in every major election
           since i turned twenty-one
- oh that's [PVT] great
```

Using this division, one could consider a more complex ”conversational” language model that is a finite state grammar reflecting larger discourse structure in the conversation and this internal structure of sentences, as shown in Figure Figure 11.

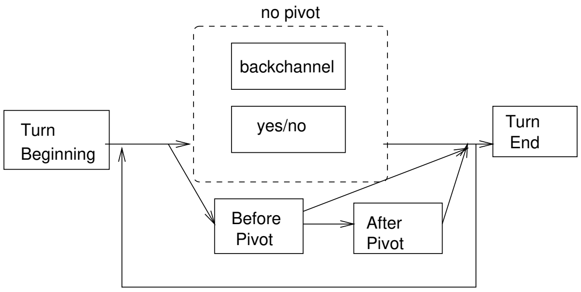

Figure 11: Given-new structure of utterances.

By using a part of speech marked corpus and a list of stop words for ”weak” verbs, we were able to mark most of the switchboard training corpus with pivot points. Note that all backchannels and other utterance types without verbs and sentences that were aborted before the verb cannot be marked and are considered separately.

There are many implications for this model on processing language, for example that the information extracted from early parts of the sentence are more likely to carry redundant information already communicated in the conversation, whereas later parts of the statement may constrain more new information.

The focus of our work with this data at the Summer Workshop was looking at how creating separate language models for before and after the pivot of a sentence effected the word error rate for speech recognition. We began by marking the pivot in the training corpus by applying an algorithm that used the part of speech marked data to find the verbs and a list of weak verbs to determine which is the first strong verb. This is the same algorithm that is reported in Meteer and Iyer (1995). The next step was to run a cheating experiment which used a test corpus where the pivot point was known. As shown in Table 36, word error rate improved slightly over the baseline. (Note that the baseline is slightly worse in performance here than previously reported since not all of the data had both part of speech and discourse labels marked, so the training set was smaller.)

In the next three results shown in the table, the pivot point is determined automatically in the test set. In the first, we used the training marked by the algorithm directly. In the second, we used that training to create a pivot model which could determine the pivot automatically and then remarked the training with that model, and in the third, we used the automatically marked data and interpolated that with the full SWBD language model. This last was the best performing, showing a full point improvement over the baseline.

One interesting result of these experiments is that the models do not improve evenly across discourse types. As we might expect, these models improve on statements and hurt performance on backchannels. However, somewhat more surprising is that they also hurt performance on opinions. Further analysis is needed to determine how these two discourse types differ in their given and new structure such that they perform differently on the pivot language model.

Table 36: *Pivot models improve WER on some DA's.*

			Improvement in WACC from Baseline			
Train	WER	PPL	Statement	Opinion	BackChannel	Abandonment
Baseline	41.2	74.81				
Cheating	40.8	70.78	+0.71	+0.44	+7.72	+8.34
Pivot Algorithm	42.0	85	-0.22	-2.60	-1.11	-10.08
Pivot Automatic	41.7	80	+0.17	-1.16	-0.83	-7.89
Pivot Auto Interp	41.0	76	+1.00	-0.39	0.00	-5.25

7 Looking at the Data – Looking for LVCSR Error Sources

In speech recognition research there are two traditions for identifying error sources. One approach we call the engineering or statistical approach focuses the attention on a single metric, optimally on a single number such as word accuracy or perplexity, that can be calculated automatically. On the other end of the spectrum the comparison of machine with human transcripts is used to determine properties of errors the LVCSR system is producing. We call this the "language experts approach".

Both approaches have disadvantages. The engineering or statistical approach is hard to interpret and laborious to implement. Additionally it can often only verify a hypothesis and rarely generates new hypotheses. The language experts approach takes a long time to carry out and reading the transcripts and assigning error sources can be tedious and confusing.

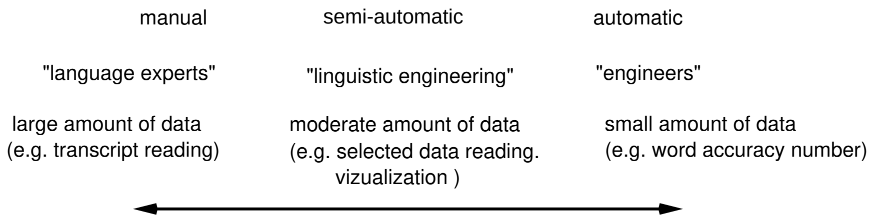

Figure 12: Viewing error analysis as a continuum: In addition to matrices like word accuracy and to unsupported transcript reading an intermediate view on the data is important.

We needed tools that would allow us to view speech transcription system output from a third perspective, one that could be situated in the middle of the continuum. We call this the "linguistically engineered" viewpoint. Since we were interested in determining the effectiveness of discourse models of dialog in improving automatic transcription accuracy, we needed discourse (linguistic) views of the human and system transcripts. The main characteristic of this approach is that it uses automatic means to provide an effective visualization of the data for the human. We found all three views of the human and system transcription necessary in order to analyze system error and to build language models that would improve system accuracy.

We found the linguistically engineered tools useful for many reasons. We particularly liked the fact that we could easily discover trends in the data that we had not previously been aware of. It was the partitioning of the Yes-No Question dialog acts into their own subcorpus that revealed that the word "do" is consistently placed at the beginning of the Question and is also frequently mis-recognized (Figure 13). We would never have observed this clear trend in the data if we had looked at all the possible dialog acts at once.

Below we present a number of example analyses. Our error analysis tool combines automatic with manual methods: It uses alignments, dialog act information and the utterance order in the dialog to display the data selectively, group it and infer additional metrics. The analysis of content dependent statements also shows how an analysis can be made both with manual and automatic means. The salience analysis and error matrix exemplifies how a good representation of the most important facts can help to understand the data. The mimic and repetition analyses also shows that a good representation is important to understand the importance of a new concept.

```
REF:  DO YOU HAVE   SMOG LIKE   THEY DO     in  california
HYP:  ** *** YOU'RE NOT  SMOKE  I    COULD  in  california
REF:  do you have SMOG LIKE THEY   DO      in california
HYP:  do you have **** **** SMOKE  ACTIVE  in california


REF: -DO YOU REALLY think CARS CONTRIBUTE A    LOT NOW  THAT
HYR: *** AND THEY   think **** ARE        GOOD TO  KNOW ABOUT
REF: -do you REALLY think CARS CONTRIBUTE A    LOT NOW  THAT
HYP: do  you ****** think **** OUR        KIDS YOU KNOW ABOUT

REF: DO YOU
HYP: ** OKAY
REF: do you
HYP: do you
```

Figure 13: Looking at selected data: When we decided to look for the improvements the dialog act specific language model could achieve and selected Questions. The question initial word *do* was very prominent. The figure shows four utterances; the top reference/hypothesis pair uses the baseline language model, while the bottom pair uses a Question-specific language model.

7.1 Error Analysis Tool

One of the traditional tools for error analysis is the alignment of the LVCSR system output with the reference produced by human transcribers (Figure 13). The dialog act categories allowed us to segment the data into smaller pieces and make subanalyses of these individual portions. Additionally, we wanted to compare different outputs of the LVCSR system using the different language models.

The error analysis tool allowed us to specify complex groupings of utterances (e.g., specific dialog act types) and to calculate additional measurements from the alignments. To display the results, the tool offered the option to display only a summary for each defined grouping or alternatively, also a user-defined output for each utterance within that group. As mentioned earlier, the tool is able to handle multiple alignments. Therefore it takes just one command to produce a comparative output of all questions aligned with the standard language model and the cheating language model. With the same tool and in one command, we produced a rundown of the word accuracies of statements of different lengths.

The tool also calculated a lot of information available from the alignment output. For example, the utterance initial analysis that became interesting after we have found the frequent errors on initial *do*'s in Questions (Table 37) needed a separate word error calculation for the initial portion of the utterance.

Table 37: *Turn initial improvements from dialog act knowledge: We compare the error rate over the whole turn (all) with the error rate in the first three words (initial) in the baseline system and contrast that with the relative improvement we gain from using dialog act specific language models (cheating). The overall trend is, that the dialog act specific model corrects errors at the beginning of the utterance.*

Dialog Act	Word Accuracy Baseline all	Word Accuracy Baseline initial	Improvement Cheating all	Improvement Cheating initial
Statement	58.1	58.9	0.71	1.20
Backchannel	78.2	78.2	7.72	7.53
Opinion	59.6	57.0	0.47	1.07
Abandoned	52.3	50.8	7.79	11.64
Agree/Accept	81.1	83.6	-1.22	0.69
SWBD	58.6	57.4	-1.49	3.52

7.2 Context Dependent Statements

After the initial error analysis we found that we achieved a reduced word error rate in the cheating experiments on a number of dialog acts. However, about 83% of the words occur in statements (sd and sv) and this class showed little to no improvement using dialog act-specific language models. This resulted in a fairly small overall word error reduction. We therefore concluded that statements needed further subcategorization so that we could make predictions about their content from context and prosody. We explicitly did not distinguish between Statements and Answers, since Answers are Statements following Questions. We tried to look for subcategorizations of Statements that had a reasonably different surface form.

One way of partitioning Statements (sd and sv) is to make them context-dependent on the last and following dialog act. The dialog acts were grouped into 6 categories: sd, sv, backchannel, question, unfinished and other. We also noted whether the dialog act occurred on the same or on the other channel. This split resulted in $(6 \cdot 2)^2 = 144$ different classes of statements. Manual inspection of these classes seemed to show that they are fairly different, especially in their average length. We were able to distinguish the following groups and gave them intuitive names according to their properties as seen in Table 38.

Using this and other groupings we tried to build language models for these Statement subtypes and compare them to the standard Statement model. Even though we used linear interpolation to counteract data fragmentation, we achieved no significant reduction in perplexity. We concluded that this initial attempt was too simplistic and more sophisticated models and clustering techniques are necessary. Nevertheless, we gained insight into the relation between Statement context and Statement length. We hope to use the outcome of this investigation in future experiments.

Table 38: *Utterance length by group.*

Answers	Short
Following another person's Question Marked sd^e (extended answers)	Following Another person's Statement Following own backchannel
Long	**Medium**
Preceding another person's Backchannel Preceding or following own sv	Following own sd Other

7.3 Salience Analysis and Error Matrix

Our integrated model had two goals:1) to detect speech acts and 2) to constrain the language model to the dialog act-specific model. We asked the following questions:

- does the LVCSR system detect the words that discriminate between dialog acts?
- which dialog acts are discriminated?
- which words frequently discriminate and do they correlate with the dialog act type?
- are there dialog act-specific frequent words that are often wrongly recognized?

If we were working with higher order N-gram models, a manual analysis of the dialog act detection model would be infeasible. We therefore used mainly unigrams enriched with approximately 190 multiwords like **YOU_KNOW** that have been used in LVCSR systems in the recent past. We used the frequency of words per dialog act, their salience [10] per dialog act and their word error rate as measures and blended this into one visualization (see Table 39).

[10] The definition of salience according to Gorin (1995) is

$$\text{salience}(v) = \sum_k P(c_k|v) \cdot I(v, c_k) = \sum_k P(c_k|v) \cdot \log \frac{p(v, c_k)}{p(v) \cdot p(c_k)}$$

where c_k is the category (e.g. the speech act type) and v is the word. To calculate the salience for a single speech act **S** we calculated the salience for the two categories **S** and **OTHER**.

We found this useful especially since we saw that most of the highly salient words were fairly well represented in the domain. We could therefore hope to see enough examples of words triggering decisions for one dialog act vs. another and that the effects of constraining a dialog act could be strong. Additionally, we observed that some of the highly salient and frequent words are short and have a high word error rate associated with them. These recognition errors could either throw off the dialog act detector or they could be improved by the dialog act-dependent language models.

Table 39: *Error Matrix: The error matrix shows that the salient words are often frequent and vice versa. Some of the salient words are really short and are often misrecognized. The representation combines many different analyses into one scheme – it is a compact representation of our experiments.*

Word ranked by frequency					Words ranked
Word	Frequency	Salience	Rank in Salience	Word Error	by salience
Statement-Non-opinion					
and	18116	0.31816	2	64%	THE
the	17570	0.34334	1	27%	AND
I	14600	0.22314	8	50%	UH
uh	14250	0.30139	3	0%	YEAH
that	13538	0.29036	5	65%	THAT
Acknowledge (Backchannel)					
uh_huh	14446	2.17960	1	66%	UH_HUH
yeah	13776	2.12916	2	33%	YEAH
right	3583	0.73160	3	85%	RIGHT
oh	2543	0.60609	4	12%	OH
okay	770	0.34320	8	65%	UH
Statement-Opinion; Explicit Performative					
the	8088	0.37340	1	65%	THE
that	7232	0.32480	3	56%	AND
and	5870	0.32701	2	35%	THAT
to	5399	0.26613	5	93%	UH
uh	5234	0.30052	4	53%	TO
Abandoned/Turn-Exit; Uninterpretable					
uh	2202	0.78549	1	32%	UH
so	2120	0.71340	2	52%	SO
but	1635	0.62159	4	50%	AND
and	1490	0.67716	3	27%	BUT
I	1120	0.57710	5	0%	I
Agree/Accept					
yeah	3757	1.60606	1	55%	YEAH
right	1878	1.01612	2	67%	RIGHT
that's	1333	0.87490	3	23%	THAT'S
yes	752	0.68088	4	0%	YES
true	641	0.64361	8	0%	OH

Word ranked by frequency					Words ranked
Word	Frequency	Salience	Rank in Salience	Word Error	by salience
Appreciation; Exclamation					
oh	1557	1.27145	1	9%	OH
that's	1510	1.25348	2	88%	THAT'S
good	620	0.88074	3	0%	GOOD
well	487	0.85942	5	0%	THAT
wow	386	0.77154	7	0%	WELL
Yes-No-Question and Alternative 'or' Question					
you	1005	0.50757	4	41%	THE
do_you	948	0.45656	9	0%	UH
the	788	0.56225	1	50%	THAT
that	701	0.52502	3	33%	YOU
uh	695	0.52868	2	56%	IT
Non Speech					
uh_huh	15	5.50831	1	0%	UH_HUH
yeah	13	5.46058	2	33%	YEAH
right	4	5.14351	4	—	UH
uh	3	5.14905	3	—	RIGHT
well	3	5.11823	5	—	WELL
Yes Answers					
yeah	1865	3.71046	1	56%	YEAH
yes	624	2.13051	2	0%	YES
uh_huh	409	1.93224	3	67%	UH_HUH
oh	179	1.62500	4	9%	OH
uh	106	1.59558	5	0%	UH
Conventional-Closing					
bye	932	1.33504	1	—	BYE
you	359	1.06271	3	0%	WELL
well	358	1.06483	2	—	YOU
talking	325	1.00178	4	0%	TALKING
to_you	305	0.98986	5	—	TO_YOU

7.4 Mimic and Repetition Analysis

Besides constraining an utterance to a dialog act-specific language model, we looked into one more option that would facilitate the integration of discourse knowledge into an LVCSR system. The knowledge sources we use should be easily and reliably obtained from recognizer output. Our intuition was that within short dialog segments, lexical choices are repeated. We assume that a name introduced by one person is often repeated by the other person and that there are specific Question/Answer patterns where words are repeated. The purpose of repeating in spontaneous dialog could be to ensure that the dialog partners are referring to the same entity.

Both mimics and repetitions were labeled by the dialog act labelers. They distinguished between repeating words from the other speaker (mimic) and the same speaker (self-repeat). These occurrences could easily be extracted using our error analysis tool.

This analysis showed two things: Mimics are relatively rare and they are probably not worth focusing on. In self-repeats, however, that either both are correct or both are incorrect by far outweighs the cases where one is correct and the other is incorrect. This indicates that we cannot make good use of this constraint in the language model to improve word accuracy. We can only correct an error, if we have one correct case and can use that to correct an incorrect one.

Table 40: *Recognition performance for Self Repeats and Mimics.*

First Mention	Second Mention: Mimics correct	Mimics incorrect	Self-Repeat correct	Self-Repeat incorrect
correct	3	1	43	8
incorrect	3	9	5	14

7.5 Implications of Discourse Model for Pronunciation Modeling: Locations of Errors

Is a given word w pronounced differently depending on where it occurs in an utterance? We suspected that the answer was yes, that words which were utterance-initial tend to have reduced pronunciations, based on our observations of a high number of initial-word deletions in the recognizer outputs (i.e. of initial words which were completely missed by the recognizer).

Upon investigation of the error alignments from the WS97-baseline recognizer run on the WS97-DevTest (Percents are percents of the REF words) we found that:

1. The first word in a WS97devtest utterance (linguistic segment) has almost twice the deletion rate of the average non-initial word (15% versus 8%).
2. But the overall error rate for the first word (36%) is actually less than the average other word (40%), because the substitution rate is much lower (16% versus 28%).

Table 41: *WS97-DevTest error rates in Utterance-Initial versus Non-initial words with baseline LM.*

Type	WER (%) Initial	WER (%) Non-Initial
Correct	64	60
Substitutions	16	28
Deletions	15	8
Insertions	4	3

We hypothesize that the high initial deletion rate is caused by reduced utterance-initial pronunciation, but this requires further investigation. We suspect the reduced overall error rate for the initial word has to do with the stronger LM predictability of the initial word; the utterance-initial word always has the correct context (<s>) for its bigram prediction.

8 CONCLUSIONS

We have described a new approach for statistical modeling and detection of discourse structure for natural conversational speech. Our algorithm has possibilities for reducing word error in speech recognition. Although the skewed dialog act distribution limited our maximum word error improvement for SWBD, improvements for WER of individual dialog acts suggests that the algorithm has potential to improve recognition on other tasks (like conversational agents) where questions and other non-statements are more common. Furthermore, by combining our 3 knowledge sources, we achieved significant improvements in our ability to automatically detect dialog acts, which will help address tasks like understanding spontaneous dialog and building human-computer dialog systems.

Acknowledgments

This project was supported by the generosity of many: the 1997 Workshop on Innovative Techniques in LVCSR, the Center for Speech and Language Processing at Johns Hopkins University, and the NSF (via NSF IRI-9619921 and IRI-9314967 to Liz Shriberg, and IRI-970406 to Dan Jurafsky). Thanks to the students who did the labeling: (1) the discourse labelers at Boulder: Debra Biasca (who managed the Boulder labelers), Marion Bond, Traci Curl, Anu Erringer, Michelle Gregory, Lori Heintzelman, Taimi Metzler, and Amma Oduro and (2) the intonation labelers at Edinburgh: Helen Wright, Kurt Dusterhoff, Rob Clark, Cassie Mayo and Matthew Bull. Many thanks to Susann LuperFoy, Nigel Ward, James Allen, Julia Hirschberg, and Marilyn Walker for advice on the design of the SWBD-DAMSL tag-set, to Mitch Weintraub and Chuck Wooters for many helpful comments, to Bill Byrne, Harriet Nock, and Joe Picone for running the baselines and providing and checking our test data and recognition environment and generally being extraordinarily helpful. And thanks to Fred Jelinek for his advice and encouragement and for the opportunity to work on this project and of course to Kimberly Shiring for everything else.

References

ALLEN, JAMES, and MARK CORE, 1997. Draft of DAMSL: Dialog act markup in several layers.

BARD, E., C. SOTILLO, A. ANDERSON, and M. TAYLOR. 1995. The DCIEM map task corpus: Spontaneous dialogues under sleep deprivation and drug. Proc. ESCA-NATO Tutorial and Workshop on Speech under Stress, Lisbon.

BREIMAN, L., J. H. FRIEDMAN, R. A. OLSHEN, and C. J. STONE. 1983. Classification and Regression Trees. Pacific Grove, California: Wadsworth & Brooks.

CARLETTA, JEAN. 1996. Assessing agreement on classification tasks: The Kappa statistic. Computational Linguistics 22.249–254.

——, AMY ISARD, STEPHEN ISARD, JACQUELINE C. KOWTKO, GWYNETH DOHERTY-SNEDDON, and ANNE H. ANDERSON. 1997. The reliability of a dialogue structure coding scheme. Computational Linguistics 23.13–32.

DELLA PIETRA, S., V. DELLA PIETRA, and J. LAFFERTY. 1997. Inducing features in random fields. IEEE Transactions on Pattern Analysis and Machine Intelligence 19.1–13.

FLAMMIA, GIOVANNI, and VICTOR ZUE. 1995. Empirical evaluation of human performance and agreement in parsing discourse constituents in spoken dialogue. EUROSPEECH-95, 1965–1968, Madrid.

GODFREY, J., E. HOLLIMAN, and J. MCDANIEL. 1992. SWITCHBOARD: Telephone speech corpus for research and development. Proceedings of ICASSP-92, 517–520, San Francisco.

GORIN, ALLEN. 1995. On automated language acquistition. Journal of the Acoustical Society of America 97.3441–3461.

GROSZ, BARBARA, and JULIA HIRSCHBERG. 1992. Some intonational characteristics of discourse structure. ICSLP-92, 429–432, Banff, Canada.

GROSZ, BARBARA J., ARAVIND K. JOSHI, and SCOTT WEINSTEIN. 1995. Centering: A framework for modeling the local coherence of discourse. Computational Linguistics 21.203–225.

HEARST, MARTI A. 1997. Texttiling: Segmenting text into multi-paragraph subtopic passages. Computational Linguistics 23.33–64.

HIRSCHBERG, JULIA, and CHRISTINE NAKATANI. 1996. A prosodic analysis of discourse segments in direction-giving monologues. Proceedings of ACL-96, 286–293.

IYER, RUKMINI, MARI OSTENDORF, and J. ROBIN ROHLICEK. 1994. Language modeling with sentence-level mixtures. ARPA Human Language Technologies Workshop, 82–86, Plainsboro, N.J.

JEFFERSON, GAIL. 1984. Notes on a systematic deployment of the acknowledgement tokens 'yeah' and 'mm hm'. Papers in Linguistics 197–216.

JURAFSKY, DANIEL, REBECCA BATES, NOAH COCCARO, RACHEL MARTIN, MARIE METEER, KLAUS RIES, ELIZABETH SHRIBERG, ANDREAS STOLCKE, PAUL TAYLOR, and CAROL VAN ESS-DYKEMA. 1997a. Automatic detection of discourse structure for speech recognition and understanding. Proceedings of the 1997 IEEE Workshop on Speech Recognition and Understanding, Santa Barbara.

JURAFSKY, DANIEL, ELIZABETH SHRIBERG, and DEBRA BIASCA, 1997b. Switchboard-DAMSL Labeling Project Coder's Manual. http://stripe.colorado.edu/˜jurafsky/ manual.august1.html.

KATZ, SLAVA M. 1987. Estimation of probabilities from sparse data for the language model component of a speech recogniser. IEEE Trans. ASSP 35.400–401.

KITA, KENJI, YOSHIKAZU FUKUI, MASAAKI NAGATA, and TSUYOSHI MORIMOTO. 1996. Automatic acquisition of probabilistic dialogue models. ICSLP-96, 196–199, Philadephia.

KOWTKO, JACQUELINE C., 1996. The Function of Intonation in Task Oriented Dialogue. University of Edinburgh dissertation.

KUHN, ROLAND, and RENATO DE MORI. 1990. A cache-based natural language model for speech recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 12.570–583.

LAVIE, ALON, DONNA GATES, NOAH COCCARO, and LORI LEVIN. 1996a. Input segmentation of spontaneous speech in janus: A speech-to-speech translation system. Proceedings of the ECAI 96, Budapest.

LAVIE, ALON, LORI LEVIN, YAN QU, ALEX WAIBEL, DONNA GATES, MARSAL GAVALDA, LAURA MAYFIELD, and MAITE TABOADA. 1996b. Dialogue processing in a conversational speech translation system. ICSLP-96, Philadephia.

LITMAN, DIANE J., and JAMES F. ALLEN. 1987. A plan recognition model for subdialogues in conversations. Cognitive Science 11.163–200.

MAST, M., R. KOMPE, ST. HARBECK, A. KIESSLING, H. NIEMANN, , and E. NÖTH. 1996. Dialog act classification with the help of prosody. ICSLP-96, 1728–1731, Philadephia.

METEER, MARIE, and OTHERS. 1995. Dysfluency Annotation Stylebook for the Switchboard Corpus. Linguistic Data Consortium. Revised June 1995 by Ann Taylor. ftp://ftp.cis.upenn.edu/pub/treebank/swbd/doc/DFL-book.ps.gz.

MORGAN, NELSON, ERIC FOSLER, and NIKKI MIRGHAFORI. 1997. Speech recognition using on-line estimation of speaking rate. EUROSPEECH-97, Rhodes, Greece.

NAGATA, MASAAKI, and TSUYOSHI MORIMOTO. 1994. First steps toward statistical modeling of dialogue to predict the speech act type of the next utterance. Speech Communication 15.193–203.

NEUMEYER, LEONARDO, and MITCH WEINTRAUB. 1994. Microphone-independent robust signal processing using probabilistic optimum filtering. ARPA HLT Workshop, 336–341, Plainsboro, NJ.

——, and ——. 1995. Robust speech recognition in noise using adaptation and mapping techniques. ICASSP-95, volume 1, 141–144, Detroit.

OSTENDORF, M., and N. VEILLEUX. 1994. A hierarchical stochastic model for automatic prediction of prosodic boundary location. Computational Linguistics 20.

PASSONNEAU, REBECCA, and DIANE LITMAN. 1993. Feasibility of automated discourse segmentation. Proceedings of the 31st ACL, 148–155.

PERRAULT, C. R., and J. F. ALLEN. 1980. A plan-based analysis of indirect speech acts. American Journal of Computational Linguistics 6.167–182.

RABINER, L. R., and B. H. JUANG. 1986. An introduction to hidden Markov models. IEEE ASSP Magazine 3.4–16.

REITHINGER, NORBERT, RALF ENGEL, MICHAEL KIPP, and MARTIN KLESEN. 1996. Predicting dialogue acts for a speech-to-speech translation system. ICSLP-96, 654–657, Philadephia.

ROSENFELD, R., R. AGARWAL, B. BYRNE, R. IYER, M. LIBERMAN, E. SHRIBERG, J. UNVERFUEHRT, D. VERGYRI, and E. VIDAL. 1996. LM95 Project Report: Language modeling of spontaneous speech. Technical Report Research Note No. 1, Center for Language and Speech Processing, Johns Hopkins University, Baltimore.

ROSENFELD, RONI. 1996. A maximum entropy approach to adaptive statistical language modeling. Computer Speech and Language 10.187–228.

SACKS, HARVEY, EMANUEL A. SCHEGLOFF, and GAIL JEFFERSON. 1974. A simplest systematics for the organization of turn-taking for conversation. Language 50.696–735.

SCHEGLOFF, EMANUAL. 1968. Sequencing in conversational openings. American Anthropologist 70.1075–1095.

SCHEGLOFF, EMANUAL A. 1982. Discourse as an interactional achievement: Some uses of 'uh huh' and other things that come between sentences. Analyzing discourse: Text and talk, edited by deborah tannen. Washington, D.C.: Georgetown University Press.

SEARLE, JOHN R. 1969. Speech Acts. Cambridge: Cambridge University Press.

SHRIBERG, ELIZABETH, REBECCA BATES, and ANDREAS STOLCKE. 1997. A prosody-only decision-tree model for disfluency detection. EUROSPEECH-97, volume 5, 2383–2386, Rhodes, Greece.

SHRIBERG, ELIZABETH, REBECCA BATES, PAUL TAYLOR, ANDREAS STOLCKE, DANIEL JURAFSKY, KLAUS RIES, NOAH COCCARO, RACHEL MARTIN, MARIE METEER, and CAROL VAN ESS-DYKEMA. 1998. Can prosody aid the automatic classification of dialog acts in conversational speech? Submitted to Language and Speech .

STOLCKE, ANDREAS. 1997. Modeling linguistic segment and turn boundaries for N-best rescoring of of spontaneous speech. EUROSPEECH-97, volume 5, 2779–2782.

——, YOCHAI KONIG, and MITCH WEINTRAUB. 1997. Explicit word error minimization in N-best list rescoring. EUROSPEECH-97, volume 1, 163–166.

——, and ELIZABETH SHRIBERG. 1996. Automatic linguistic segmentation of conversational speech. ICSLP-96, 1005–1008, Philadephia.

——, ——, REBECCA BATES, NOAH COCCARO, DANIEL JURAFSKY, RACHEL MARTIN, MARIE METEER, KLAUS RIES, PAUL TAYLOR, and CAROL VAN ESS-DYKEMA. 1998. Dialog act modeling for conversational speech. AAAI Spring Symposium on Applying Machine Learning to Discourse Processing . to appear.

SUHM, B., and A. WAIBEL. 1994. Toward better language models for spontaneous speech. ICSLP-94, 831–834.

TAYLOR, PAUL, SIMON KING, STEPHEN ISARD, and HELEN WRIGHT. 1998. Intonation and dialogue context as constraints for speech recognition. Submitted to Language and Speech .

——, ——, ——, ——, and JACQUELINE KOWTKO. 1997. Using intonation to constrain language models in speech recognition. EUROSPEECH-97, 2763–2766, Rhodes, Greece.

TAYLOR, PAUL, H. SHIMODAIRA, STEPHEN ISARD, SIMON KING, and JACQUELINE KOWTKO. 1996. Using prosodic information to constrain language models for spoken dialogue. ICSLP'96, Philadelphia.

TERRY, MARK, RANDALL SPARKS, and PATRICK OBENCHAIN. 1994. Automated query identification in English dialogue. ICSLP-94, 891–894.

WAIBEL, ALEX. 1988. Prosody and Speech Recognition. San Mateo, CA.: Morgan Kaufmann.

WALKER, MARILYN A., and ELLEN F. PRINCE. 1993. A bilateral approach to givenness: A hearer-status algorithm and a centering algorithm. Reference and referent accessibility, ed. by T. Fretheim and J. Gundel. Amsterdam: John Benjamins.

WEBER, ELIZABETH G. 1993. Varieties of Questions in English Conversation. Amsterdam: John Benjamins.

WITTEN, I. H., and T. C. BELL. 1991. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. IEEE Trans. Information Theory 37.1085–1094.

WOSZCZYNA, M., and A. WAIBEL. 1994. Inferring linguistic structure in spoken language. ICSLP-94, 847–850, Yokohama, Japan.

YAMAOKA, TAKAYUKI, and HITOSHI IIDA. 1991. Dialogue interpretation model and its application to next utterance prediction for spoken language processing. EUROSPEECH-91, 849–852, Genova, Italy.

YNGVE, VICTOR H. 1970. On getting a word in edgewise. Papers from the 6th Regional Meeting of the Chicago Linguistics Society, 567–577, Chicago.