# Speech Repairs, Intonational Phrases, and Discourse Markers: Modeling Speakers' Utterances in Spoken Dialogue

Peter A. Heeman*
Oregon Graduate Institute

James F. Allen†
University of Rochester

*Interactive spoken dialogue provides many new challenges for natural language understanding systems. One of the most critical challenges is simply determining the speaker's intended utterances: both segmenting a speaker's turn into utterances and determining the intended words in each utterance. Even assuming perfect word recognition, the latter problem is complicated by the occurrence of speech repairs, which occur where speakers go back and change (or repeat) something they just said. The words that are replaced or repeated are no longer part of the intended utterance, and so need to be identified. Segmenting turns and resolving repairs are strongly intertwined with a third task: identifying discourse markers. Because of the interactions, and interactions with POS tagging and speech recognition, we need to address these tasks together and early on in the processing stream. This paper presents a statistical language model in which we redefine the speech recognition problem so that it includes the identification of POS tags, discourse markers, speech repairs, and intonational phrases. By solving these simultaneously, we obtain better results on each task than addressing them separately. Our model is able to identify 72% of turn-internal intonational boundaries with a precision of 71%, 97% of discourse markers with 96% precision, and detect and correct 66% of repairs with 74% precision.*

## 1. Introduction

Consider the following example from the Trains corpus (Heeman and Allen 1995).

**Example 1** (d93-13.3 utt63)

um it'll be there it'll get to Dansville at three a.m. and then you wanna do you take tho- want to take those back to Elmira so engine E two with three boxcars will be back in Elmira at six a.m. is that what you wanna do

In order to understand what the speaker was trying to say, the reader probably segmented the above into a number of sentence-like segments, **utterances**, as follows.

**Example 1 Revisited**

um it'll be there it'll get to Dansville at three a.m.
and then you wanna do you take tho- want to take those back to Elmira
so engine E two with three boxcars will be back in Elmira at six a.m.
is that what you wanna do

* Computer Science and Engineering, P.O. Box 91000, Portland, OR 97291. E-mail: heeman@cse.ogi.edu
† Department of Computer Science, Rochester, NY 14627. E-mail: james@cs.rochester.edu

Even this does not fully capture what the speaker was intending to convey. The first and second utterances contain **speech repairs**, where the speaker goes back and changes (or repeats) something she just said. In the first, the speaker changed *it'll be there* to *it'll get to*; in the second, she changed *you wanna* to *do you take tho-*, which she then further revised. The speaker's intended utterances are thus as follows:[1]

**Example 1 Revisited Again**

um it'll get to Dansville at three a.m.
and then do you want to take those back to Elmira
so engine E two with three boxcars will be back in Elmira at six a.m.
is that what you wanna do

The tasks of segmenting speakers' turns into utterance units and resolving speech repairs are strongly intertwined with a third task: identifying whether words, such as *so*, *well*, and *right*, are part of the sentential content or are being used as discourse markers to relate the current speech to the preceding context. In the example above, the second and third utterances begin with discourse markers.

### 1.1 Utterance Units and Intonational Phrases

As illustrated above, understanding a speaker's turn necessitates segmenting it into individual utterance units. However, there is no consensus as to how to define an utterance unit (Traum and Heeman 1997). The manner in which speakers break their speech into intonational phrases undoubtedly plays a major role in its definition. Intonational phrase endings are signaled through variations in the pitch contour, segmental lengthening, and pauses. Beach (1991) demonstrated that hearers can use intonational information early on in sentence processing to help resolve syntactic ambiguities. Bear and Price (1990) showed that a parser can use automatically extracted intonational phrasing to reduce ambiguity and improve efficiency. Ostendorf, Wightman, and Veilleux (1993) used hand-labeled intonational phrasing to do syntactic disambiguation and achieved performance comparable to that of human listeners. Due to their significance, we will focus on the task of detecting intonational phrase boundaries.

### 1.2 Speech Repairs

The on-line nature of spoken dialogue forces conversants to sometimes start speaking before they are sure of what they want to say. Hence, the speaker might need to go back and repeat or modify what she just said. Of course there are many different reasons why speakers make repairs; but whatever the reason, speech repairs are a normal occurrence in spoken dialogue. In the Trains corpus, 23% of speaker turns contain at least one repair and 54% of turns with at least 10 words contain a repair.

Fortunately for the hearer, speech repairs tend to have a standard form. As illustrated by the following example, they can be divided into three intervals, or stretches of speech: the **reparandum**, **editing term**, and **alteration**.[2]

1 The speech that was revised cannot simply be thrown out since it might contain information, such as the identity of an anaphoric reference as the following example shows: *Peter was well he was fired.*

**Example 2** (d92a-2.1 utt29)

that's the one with the bananas I mean that's taking the bananas
*reparandum* *ip* *editing terms* *alteration*

The reparandum is the stretch of speech that the speaker is replacing, and can end with a **word fragment**, where the speaker interrupts herself during the middle of a word. The end of the reparandum is the **interruption point** and is often accompanied by a disruption in the intonational contour. This can be optionally followed by the editing term, which can consist of filled pauses, such as *um* or *uh* or cue phrases, such as *I mean, well,* or *let's see.* Reparanda and editing terms account for 10% of the words in the Trains corpus. The last part is the alteration, which is the speech that the speaker intends as the replacement for the reparandum. In order for the hearer to determine the intended utterance, he must detect the repair and determine the extent of the reparandum and editing term. We refer to this latter process as **correcting** the speech repair. In the example above, the speaker's intended utterance is *that's the one that's taking the bananas.*

Hearers seem to be able to effortlessly understand speech with repairs in it, even when multiple repairs occur in a row. In laboratory experiments, Martin and Strange (1968) found that attending to speech repairs and the content of an utterance are mutually inhibitory, and Bard and Lickley (1997) found that subjects have difficulty remembering the actual words in the reparandum. Listeners must be resolving repairs very early on in processing the speech. Earlier work by Lickley and colleagues (Lickley, Shillcock, and Bard 1991; Lickley and Bard 1992) strongly suggests that there are prosodic cues across the interruption point that hearers make use of in detecting repairs. However, little progress has been made in detecting speech repairs based solely on acoustical cues (cf. Bear, Dowding, and Shriberg 1992; Nakatani and Hirschberg 1994; O'Shaughnessy 1994; Shriberg, Bates, and Stolcke 1997).

**1.2.1 Classification of Speech Repairs.** Psycholinguistic work in speech repairs and in understanding the implications that they pose for theories of speech production (e.g. Levelt 1983; Blackmer and Mitton 1991; Shriberg 1994) has come up with a number of classification systems. Categories are based on how the reparandum and alteration differ, for instance whether the alteration repeats the reparandum, makes it more appropriate, or fixes an error in the reparandum. Such an analysis can shed light on where in the production system the error and its repair originated. Our concern, however, is in computationally resolving repairs. The relevant features are those that the hearer has access to and can make use of in detecting and correcting a repair. Following loosely in the footsteps of the work of Hindle (1983), we divide them into the following categories: **fresh starts**, **modification repairs**, and **abridged repairs**.

Fresh starts occur where the speaker abandons the current utterance and starts again, where the abandonment seems to be acoustically signaled either in the editing term or at the onset of the alteration. Example 3 illustrates a fresh start where the speaker abandons the partial utterance *I need to send,* and replaces it by the question *how many boxcars can one engine take.*

2 Our notation is adapted from Levelt (1983). We follow Shriberg (1994) and Nakatani and Hirschberg (1994) in using reparandum to refer to the entire interval being replaced. We use alteration in the same way.

**Example 3** (d93-14.3 utt2)

I need to send let's see how many boxcars can one engine take
*reparandum* *ip* *editing terms* *alteration*

For fresh starts, there can sometimes be little or even no correlation between the reparandum and alteration. Although it is usually easy to determine the reparandum onset, initial discourse markers and preceding intonational phrases can prove problematic.

The second type are modification repairs, which comprise the remainder of repairs with a nonempty reparandum. The example below illustrates this type of repair.

**Example 4** (d92a-1.2 utt40)

you can carry them both on tow both on the same engine
*reparandum* *ip* *alteration*

Modification repairs tend to have strong word correspondences between the reparandum and alteration, which can help the hearer determine the reparandum onset as well as signal that a repair occurred. In the example above, there are word matches on the instances of *both* and *on*, and a replacement of the verb *carry* by *tow*. Modification repairs can in fact consist solely of the reparandum being repeated by the alteration.

The third type are the abridged repairs. These repairs consist of an editing term, but with no reparandum, as the following example illustrates.

**Example 5** (d93-14.3 utt42)

we need to um manage to get the bananas to Dansville more quickly
*ip* *editing terms*

For these repairs, the hearer has to determine that an editing term occurred, which can be difficult for phrases such as *let's see* or *well* since they can also have a sentential interpretation. The hearer also has to determine that the reparandum is empty. As the example above illustrates, this is not necessarily a trivial task because of the spurious word correspondences between *need to* and *manage to*.

### 1.3 Discourse Markers

Phrases such as *so*, *now*, *firstly*, *moreover*, and *anyways* can be used as discourse markers (Schiffrin 1987). Discourse markers are conjectured to give the hearer information about the discourse structure, and so aid the hearer in understanding how the new speech or text relates to what was previously said and for resolving anaphoric references (Hirschberg and Litman 1993). Although discourse markers, such as *firstly*, and *moreover*, are not commonly used in spoken dialogue (Brown and Yule 1983), a lot of other markers are employed. These markers are used to achieve a variety of effects: such as signal an acknowledgment or acceptance, hold a turn, stall for time, signal a speech repair, or signal an interruption in the discourse structure or the return from one.

Although Schiffrin defines discourse markers as bracketing units of speech, she explicitly avoids defining what the unit is. We feel that utterance units are the building

blocks of spoken dialogue and that discourse markers operate at this level to relate the current utterance to the discourse context or to signal a repair in an utterance. In the following example, *and then* helps signal that the upcoming speech is adding new information, while *so* helps indicate a summary is about to be made.

**Example 6** (d92a-1.2 utt47)

and then while at Dansville take the three boxcars
so that's total of five

### 1.4 Interactions

The tasks of identifying intonational phrases and discourse markers and detecting and correcting speech repairs are highly intertwined, and the solution to each task depends on the solution for the others.

**1.4.1 Intonational Phrases and Speech Repairs.** Phrase boundaries and interruption points of speech repairs share a number of features that can be used to identify them: there is often a pause at these events as well as lengthening of the final syllable before them. Even correspondences, traditionally associated with speech repairs, can cross phrase boundaries (indicated with "%"), as the following example shows.

**Example 7** (d93-8.3 utt73)

that's all you need %
you only need one boxcar %

Second, the reparandum onset for repairs, especially fresh starts, often occurs at the onset of an intonational phrase, and reparanda usually do not span phrase boundaries. Third, deciding if filled pauses and cue phrases should be treated as abridged repairs can only be done by taking into account whether they are midutterance or not (cf. Shriberg and Lickley 1993), which is associated with intonational phrasing.

**1.4.2 Intonational Phases and Discourse Markers.** Discourse markers tend to be used at utterance boundaries, and hence have strong interactions with intonational phrasing. In fact, Hirschberg and Litman (1993) found that discourse markers tend to occur at the beginning of intonational phrases, while sentential usages tend to occur midphrase. Example 8 below illustrates *so* being used midutterance as a subordinating conjunction, not as a discourse marker.

**Example 8** (d93-15.2 utt9)

it takes an hour to load them %
just so you know %

Now consider the third turn of the following example in which the system is not using *no* as a quantifier to mean that there are not any oranges available, but as a discourse marker in signaling that the user misrecognized *oranges* as *orange juice*.

**Table 1**
Frequency of discourse markers in the editing term of speech repairs and as the alteration onset.

| | Abridged | Modification | Fresh Starts |
|---|---|---|---|
| Number of repairs | 423 | 1,301 | 671 |
| DM in editing term | 36 | 60 | 155 |
| DM as alteration onset | 8 | 126 | 147 |
| Either | 41 | 179 | 269 |

**Example 9** (d93-11.1 utt109-111)

**system:** so so we have three boxcars of oranges at Corning
**user:** three boxcars of orange juice at Corning
**system:** no um oranges

The discourse marker interpretation is facilitated by the phrase boundary between *no* and *oranges*, especially since the determiner reading of *no* would be very unlikely to have a phrase boundary separating it from the noun it modifies. Likewise, the recognition of *no* as a discourse marker makes it more likely that there will be a phrase boundary following it.

**1.4.3 Speech Repairs and Discourse Markers.** Discourse markers are often used in the editing term to help signal that a repair occurred, and can be used to help determine if it is a fresh start (cf. Hindle 1983; Levelt 1983), as the following example illustrates.

**Example 10** (d92a-1.3 utt75)

we have the orange juice in two ↑ oh how many did we need
*reparandum* *ip* *et*

Realizing that *oh* is being used as a discourse marker helps facilitate the detection of the repair, and vice versus. This holds even if the discourse marker is not part of the editing term, but is the first word of the alteration. Table 1 shows the frequency with which discourse markers co-occur with speech repairs. We see that a discourse marker is either part of the editing term or is the alteration onset for 40% of fresh starts and 14% of modification repairs. Discourse markers also play a role in determining the onset for fresh starts, since they are often utterance initial.

**1.5 Interactions with POS Tagging and Speech Recognition**

Not only are the tasks of identifying intonational phrases and discourse markers and resolving speech repairs intertwined, but these tasks are also intertwined with identifying the lexical category or part of speech (POS) of each word, and the speech recognition problem of predicting the next word given the previous context.

Just as POS taggers for text take advantage of sentence boundaries, it is natural to assume that tagging spontaneous speech would benefit from modeling intonational phrases and speech repairs. This is especially true for repairs, since their occurrence disrupts the local context that is needed to determine the POS tags (Hindle 1983). In

the example below, both instances of *load* are being used as verbs; however, since the second instance follows a preposition, it could easily be mistaken for a noun.

**Example 11** (d93-12.4 utt44)

by the time we load in load the bananas
*reparandum* ↑ *ip*

However, by realizing that the second instance of *load* is being used in a repair and corresponds to the first instance of *load*, its POS tag becomes obvious. Conversely, since repairs disrupt the local syntactic context, this disruption, as captured by the POS tags, can be used as evidence that a repair occurred, as shown by the following example.

**Example 12** (d93-13.1 utt90)

I can run trains on the in the opposite direction
*reparandum* ↑ *ip* *alteration*

Here we have a preposition following a determiner, an event that only happens across the interruption point of a speech repair.

Just as there are interactions with POS tagging, the same holds for the speech recognition problem of predicting the next word given the previous context. For the lexical context *can run trains on the*, it would be very unlikely that the word *in* would be next. It is only by modeling the occurrence of repairs and their word correspondences that we can account for the speaker's words.

There are also interactions with intonational phrasing. In the example below, after asking the question *what time do we have to get done by*, the speaker refines this to be whether they have to be done by two p.m. The result, however, is that there is a repetition of the word *by*, but separated by a phrase boundary.

**Example 13** (d93-18.1 utt58)

what time do we have to get done by %
by two p.m. %

By modeling the intonational phrases, POS taggers and speech recognition language models would be expecting a POS tag and word that can introduce a new phrase.

### 1.6 Modeling Speakers' Utterances

In this paper, we address the problem of modeling speakers' utterances in spoken dialogue, which involves identifying intonational phrases and discourse markers and detecting and correcting speech repairs. We propose that these tasks can be done using local context and early in the processing stream. Hearers are able to resolve speech repairs and intonational phrase boundaries very early on, and hence there must be enough cues in the local context to make this feasible. We redefine the speech recognition problem so that it includes the resolution of speech repairs and identification of intonational phrases, discourse markers, and POS tags, which results in a statistical language model that is sensitive to speakers' utterances. Since all tasks are being resolved in the same model, we can account for the interactions between the tasks in a

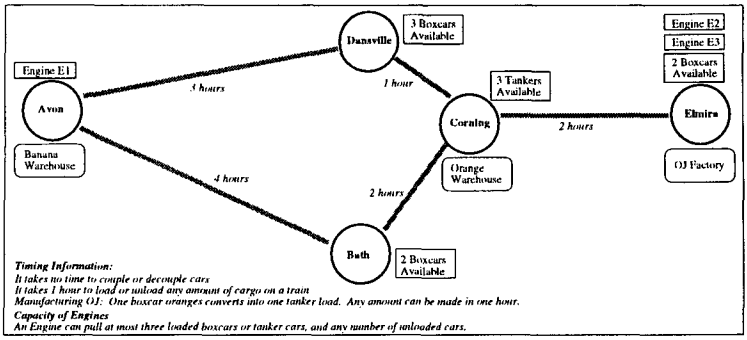


**Figure 1**
Map used by the system in collecting the Trains corpus.

framework that can compare alternative hypotheses for the speaker's turn. Not only does this allow us to model the speaker's utterance, but it also results in an improved language model, evidenced by both improved POS tagging and in better estimating the probability of the next word. Furthermore, speech repairs and phrase boundaries have acoustic correlates, such as pauses between words. By resolving speech repairs and identifying intonational phrases during speech recognition, these acoustic cues, which otherwise would be treated as noise, can give evidence as to the occurrence of these events, and further improve speech recognition results.

Resolving the speaker's utterances early on will not only help a speech recognizer determine what was said, but it will also help later processing, such as syntactic and semantic analysis. The literature (e.g., Bear and Price 1990; Ostendorf, Wightman, and Veilleus 1993) already indicates the usefulness of intonational information for syntactic processing. Resolving speech repairs will further simplify syntactic and semantic understanding of spontaneous speech, since it will remove the apparent ill-formedness that speech repairs cause. This will also make it easier for these processes to cope with the added syntactic and semantic variance that spoken dialogue seems to license.

### 1.7 Overview of the Paper

We next describe the Trains corpus and the annotation of speech repairs, intonational phrases, discourse markers, and POS tags. We then introduce a language model that incorporates POS tagging and discourse marker identification. We then augment it with speech repair and intonational phrase detection, repair correction, and silence information, and give a sample run of the model. We then evaluate the model by analyzing the effects that each component of the model has on the other components. Finally, we compare our work with previous work and present the conclusions and future work.

## 2. The Trains Corpus

One of the goals that we are pursuing at the University of Rochester is the development of a conversationally proficient planning assistant, which assists a user in constructing a plan to achieve some task involving the manufacture and shipment of goods in a railroad freight system (Allen et al. 1995). In order to do this, we need to know what kinds of phenomena occur in such dialogue. To this end, we have collected a corpus of human-human dialogues (Heeman and Allen 1995). The person playing the role of the system was given the map in Figure 1. The user was also given a map, but lacking

**Table 2**
Size of the Trains corpus.

| | | | |
|---|---|---|---|
| Dialogues | 98 | Intonational Phrases | 10,947 |
| Speaker Turns | 6,163 | Turn-Internal Phrase Boundaries | 5,535 |
| Words | 58,298 | Abridged Repairs | 423 |
| Fragments | 756 | Modification Repairs | 1,302 |
| Filled Pauses | 1,498 | Fresh Starts | 671 |
| Discourse Markers | 8,278 | Editing Terms | 1,128 |

the distances and timing information. The collection procedure was designed to make the setting as close to human-computer interaction as possible, but was not a Wizard of Oz scenario, where one person pretends to be a computer; rather, both participants know that they are speaking to a real person. Thus these dialogues provide a snapshot into an ideal human-computer interface that is able to engage in fluent conversation. Table 2 gives details about the Trains corpus. The corpus consists of six and a half hours of speech produced by 34 different speakers solving 20 different problems.

The Trains corpus provides natural examples of dialogue usage that spoken dialogue systems need to handle in order to carry on a dialogue with a user. For instance, the corpus contains instances of overlapping speech, back-channel responses, and turn taking: phenomena that do not occur in collections of single speaker utterances, such as ATIS (MADCOW 1992). The Trains corpus also differs from the Switchboard corpus (Godfrey, Holliman, and McDaniel 1992) in that it is task oriented and has a limited domain, making it a more realistic domain for studying the types of conversations that people would want to have with a computer.

### 2.1 Word Transcription

Table 3 gives a dialogue from the Trains corpus. Overlapping speech is indicated by the "+" markings. Each word was transcribed using its orthographic spelling, unless it was mispronounced and the speaker subsequently repairs the mispronunciation. Contractions, including words such as *wanna*, were transcribed as single words. Word fragments were annotated by spelling as much of the word as can be heard followed by a dash. If it was clear what word the speaker was saying, then the rest of the word was enclosed in parentheses before the dash.

### 2.2 POS and Discourse Marker Annotations

Our POS tagset is based on the Penn tagset (Marcus, Santorini, and Marcinkiewicz 1993), but modified to include tags for discourse markers and end-of-turns, and to provide richer syntactic information (Heeman 1997). Table 4 lists our tagset with differences from the Penn tagset marked in bold. Contractions are annotated using "∧" to conjoin the tag for each part; for instance, *can't* is annotated as **MD∧RB**.

Discourse marker usage is captured by the POS tags. The tag **AC** marks single word acknowledgments, such as *okay, right, mm-hm,* and *no*. The tag **CC_D** marks discourse conjuncts, such as *and, so,* and *but*. The tag **RB_D** marks discourse adverbials, such as *then, now, actually, first,* and *anyway*. Finally, **UH_D** marks interjections, such as *oh, well, hm,* and *mm*. Verbs used as discourse markers, such as *wait,* and *see,* are not given special markers, but are annotated as **VB**. No attempt has been made at analyzing multiword discourse markers, such as *by the way* and *you know*; however, phrases such as *oh really* and *and then* are treated as two individual discourse markers.

**Table 3**
Transcription of dialogue d93-12.2.

**Problem 1-B**
*Transport 2 boxcars of bananas to Corning by 11 AM. It is now midnight.*

turn1 s: hello can I help you
turn2 u: I need to take two boxcars of bananas um from Avon to Corning by eleven a.m.
turn3 s: so two boxcars of what
turn4 u: bananas
turn5 s: bananas to where
turn6 u: Corning
turn7 s: to Corning okay
turn8 u: um so the first thing we need to do is to get the uh boxcars to uh Avon
turn9 s: okay so there's boxcars in Dansville and there's boxcars in Bath
turn10 u: okay is Dansville the shortest route
turn11 s: yep
turn12 u: okay how long will it take from to to have the oh I need it ooh how long will it take to get from Avon to Dansville
turn13 s: three hours
turn14 u: okay so I'll need to go from Avon to Dansville with the engine to pick up two boxcars
turn15 s: okay so we'll g- we'll get to Dansville at three a.m.
turn16 u: okay I need to return to Avon to load the boxcars
turn17 s: okay so we'll get back to Avon at six a.m. and we'll load them which takes an hour so that'll be done by seven a.m.
turn18 u: and then we need to travel to uh Corning
turn19 s: okay so the quickest way to Corning is through Dansville which will take four hours so we'll get there at + eleven a.m. +
turn20 u: + eleven + a.m. okay it's doable
turn21 s: great

**Table 4**
Part-of-speech tags used in annotating the Trains corpus.

| | | | | | |
|---|---|---|---|---|---|
| **AC** | Acknowledgement | **HAVEP** | Present tense of *have* | **RB_D** | Discourse adverbial |
| **BE** | Base form of *be* | **HAVEZ** | 3rd person sing. present | RP | Reduced particle |
| **BED** | Past tense of *be* | JJ | Adjective | **SC** | Subordinating conjunct |
| **BEG** | Present participle of *be* | JJR | Relative Adjective | TO | *To*-infinitive |
| **BEN** | Past participle of *be* | JJS | Superlative Adjective | **TURN** | Turn marker |
| **BEP** | Present tense of *be* | MD | Modal | **UH_D** | Discourse interjection |
| **BEZ** | 3rd person sing. present | NN | Noun | **UH_FP** | Filled pause |
| CC | Coordinating conjunct | NNS | Plural noun | VB | Base form of verb (other than *do*, *be*, or *have*) |
| **CC_D** | Discourse connective | NNP | Proper Noun | | |
| CD | Cardinal number | NNPS | Plural proper Noun | VBD | Past tense |
| **DO** | Base form of *do* | PDT | Pre-determiner | VBG | Present participle |
| **DOD** | Past tense of *do* | POS | Possessive | VBN | Past participle |
| **DOP** | Present tense of *do* | **PPREP** | Pre-preposition | VBP | Present tense |
| **DOZ** | 3rd person sing. present | **PREP** | Preposition | VBZ | 3rd person sing. present |
| **DP** | Pro-form | PRP | Personal pronoun | WDT | *Wh*-determiner |
| DT | Determiner | PRP$ | Possessive pronoun | WP | *Wh*-pronoun |
| EX | Existential *there* | RB | Adverb | WRB | *Wh*-adverb |
| **HAVE** | Base form of *have* | RBR | Relative Adverb | WP$ | Possessive *Wh*-pronoun |
| **HAVED** | Past tense of *have* | RBS | Superlative Adverb | | |

### 2.3 Speech Repair Annotations

Our repair annotation scheme, defined in Table 5, is based on the one proposed by Bear et al. (1993), but extended to better deal with ambiguous and overlapping re-

**Table 5**
Labels used for annotating speech repairs.

| | |
|---|---|
| **ip***r* | Interruption point of a speech repair. Index *r* is used to distinguish between multiple repairs. Indices are in multiples of 10 and all word correspondence for the repair are given a unique index between the repair index and the next highest repair index. Repair indices of 0 are not marked, as Example 14 illustrates. |
| **ip***r***:mod** | The **mod** suffix indicates a modification repair; **mod+** indicates uncertainty as to the type of repair. |
| **ip***r***:can** | The **can** suffix indicates a fresh start (or *cancel*); **can+** marks ambiguous repairs. |
| **ip***r***:abr** | The **abr** suffix indicates an abridged repair. |
| **sr***r*< | Denotes the onset of the reparandum of a fresh start. |
| **m***i* | Used to label word correspondences in which the two words are identical. The index *i* is used both to coindex the two words and to associate them with the repair index. |
| **r***i* | Used to label word correspondences in which one word replaces another. |
| *x**r*** | Word deletion or insertion. It is indexed by the repair index. |
| **p***i* | Multiword correspondence, such as replacement of a pronoun by a longer description. |
| **et** | Used to label the editing term that follows the interruption point. |

pairs (Heeman 1997). Like their scheme, ours allows the annotator to capture word correspondences between the reparandum and alteration. Below, we give a repair annotation.

**Example 14** (d93-15.2 utt42)

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| engine | two | from | Elmi(ra)- | | or | engine | three | from | Elmira |
| **m1** | **r2** | **m3** | **m4** | ↑ | **et** | **m1** | **r2** | **m3** | **m4** |
| | | | **ip:mod+** | | | | | | |

In this example, the reparandum is *engine two from Elmi(ra)-*, the editing term is *or*, and the alteration is *engine three from Elmira*. The word matches on *engine* and *from* are annotated with **m** and the replacement of *two* by *three* is annotated with **r**. As with the POS tags, "∧" can be used in annotating contracted words.[3]

### 2.4 Intonation Annotations

For our intonation annotation, we have annotated the intonational phrase boundaries, using the ToBI (Tones and Break Indices) definition (Silverman et al. 1992). Intonational phrases are determined by both the pitch contour and the perceived juncture between each pair of words, where the perceived juncture takes into account both interword pauses and preboundary lengthening (normalized duration of the final consonants). Labeling with the full ToBI annotation scheme is very time-consuming; hence, we labeled only the intonational phrase boundaries.

## 3. POS-based Language Model

In this section, we present a speech recognition language model that incorporates POS tagging. Here, POS tags are viewed as part of the output of the speech recognizer rather than as intermediate objects. Not only is this syntactic information needed for

3 Shriberg (1994) also extends the scheme of Bear et al. (1993) to deal with overlapping repairs.

modeling the occurrence of speech repairs and intonational phrases, but it will also be useful for higher-level syntactic and semantic processes. Incorporating POS tagging can also be seen as a first step in tightening the coupling between speech recognition and natural language processing so as to be able to make use of richer knowledge of natural language than simple word-based language models provide.

### 3.1 Word-based Language Models

The goal of speech recognition is to find the most probable sequence of words $\hat{W}$ given the acoustic signal $A$ (Jelinek 1985).

$$\hat{W} = \arg\max_{W} \Pr(W|A) \tag{1}$$

Using Bayes' rule, we rewrite the above equation in the following manner.

$$\hat{W} = \arg\max_{W} \frac{\Pr(A|W)\Pr(W)}{\Pr(A)} \tag{2}$$

Since $\Pr(A)$ is independent of the choice of $W$, we can simplify the above as follows.

$$\hat{W} = \arg\max_{W} \Pr(A|W)\Pr(W) \tag{3}$$

The first term, $\Pr(A|W)$, is the **acoustic model** and the second term, $\Pr(W)$, is the **language model**. We can rewrite $W$ explicitly as the sequence of words $W_1W_2W_3\ldots W_N$, where $N$ is the number of words in the sequence. For expository ease, we use $W_{i,j}$ to refer to $W_i\ldots W_j$. We now use the definition of conditional probabilities to rewrite $\Pr(W_{1,N})$ as follows.

$$\Pr(W_{1,N}) = \prod_{i=1}^{N} \Pr(W_i|W_{1,i-1}) \tag{4}$$

The above equation gives us the probability of the word sequence as the product of the probability of each word given its previous lexical context. This probability distribution must be estimated. The simplest approach to estimating the probability of an event given a context is to use a training corpus to compute the relative frequency of the event given the context. However, no matter how large the corpus is, there will always be event-context pairs that have not been seen, or have been seen too rarely to accurately estimate the probability. To alleviate this problem, one must partition the contexts into equivalence classes and use these to compute the relative frequencies. A common technique is to partition the context based on the last $n-1$ words, $W_{i-n+1,i-1}$, which is referred to as an $n$-gram language model. One can also mix in smaller-size language models to use when there is not enough data to support the larger context. Two common approaches for doing this are interpolated estimation (Jelinek and Mercer 1980) and the backoff approach (Katz 1987).

### 3.2 Incorporating POS Tags and Discourse Marker Identification

Previous attempts to incorporate POS tags into a language model view the POS tags as intermediate objects and sum over all POS possibilities (Jelinek 1985).

$$\begin{aligned}\Pr(W_{1,N}) &= \sum_{P_{1,N}} \Pr(W_{1,N}P_{1,N}) \\ &= \sum_{P_{1,N}} \prod_{i=1}^{N} \Pr(W_i|W_{1,i-1}P_{1,i})\Pr(P_i|W_{1,i-1}P_{1,i-1})\end{aligned} \tag{5}$$

However, this throws away valuable information that is needed by later processing. Instead, we redefine the speech recognition problem so as to include finding the best POS and discourse marker sequence along with the best word sequence. For the word sequence $W$, let $D$ be a POS sequence that can include discourse marker tags. The goal of the speech recognition process is to now solve the following:

$$\begin{aligned} \hat{WD} &= \arg\max_{WD} \Pr(WD|A) \\ &= \arg\max_{WD} \Pr(A|WD)\Pr(WD) \end{aligned} \tag{6}$$

The first term $\Pr(A|WD)$ is the acoustic model, which can be approximated by $\Pr(A|W)$. The second term $\Pr(WD)$ is the POS-based language model and accounts for both the sequence of words and their POS assignment. We rewrite this term as follows:

$$\begin{aligned} \Pr(W_{1,N}D_{1,N}) &= \prod_{i=1}^{N} \Pr(W_iD_i|W_{1,i-1}D_{1,i-1}) \\ &= \prod_{i=1}^{N} \Pr(W_i|W_{1,i-1}D_{1,i})\Pr(D_i|W_{1,i-1}D_{1,i-1}) \end{aligned} \tag{7}$$

Equation 7 involves two probability distributions that need to be estimated. These are the same distributions that are needed by previous POS-based language models (Equation 5) and POS taggers (Church 1988; Charniak et al. 1993). However, these approaches simplify the context so that the lexical probability is just conditioned on the POS category of the word, and the POS probability is conditioned on just the preceding POS tags, which leads to the following two approximations.

$$\Pr(W_i|W_{1,i-1}D_{1,i}) \approx \Pr(W_i|D_i) \tag{8}$$

$$\Pr(D_i|W_{1,i-1}D_{1,i-1}) \approx \Pr(D_i|D_{1,i-1}) \tag{9}$$

However, to successfully incorporate POS information, we need to account for the full richness of the probability distributions, as will be demonstrated in Section 3.4.4.

### 3.3 Estimating the Probabilities

To estimate the probability distributions, we follow the approach of Bahl et al. (1989) and use a decision tree learning algorithm (Breiman et al. 1984) to partition the context into equivalence classes. The algorithm starts with a single node. It then finds a question to ask about the node in order to partition the node into two **leaves**, each more informative as to which event occurred than the parent node. Information-theoretic metrics, such as minimizing entropy, are used to decide which question to propose. The proposed question is then verified using held-out data: if the split does not lead to a decrease in entropy according to the held-out data, the split is rejected and the node is not further explored. This process continues with the new leaves and results in a hierarchical partitioning of the context. After the tree is grown, relative frequencies are calculated for each node, and these probabilities are then interpolated with their parent node's probabilities using a second held-out dataset.

Using the decision tree algorithm to estimate probabilities is attractive since the algorithm can choose which parts of the context are relevant, and in what order. Hence, this approach lends itself more readily to allowing extra contextual information to be

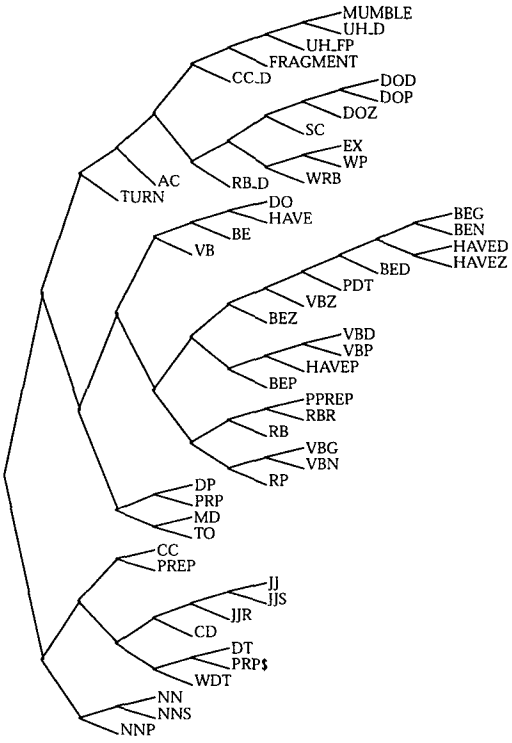


**Figure 2**
Binary classification tree that encodes the POS tags for the decision tree algorithm.

included, such as both the word identities and POS tags, and even hierarchical clusterings of them. If the extra information is not relevant, it will not be used. The approach of using decision trees will become even more critical in the next two sections, where the probability distributions will be conditioned on even richer context.

**3.3.1 Simple Questions.** One of the most important aspects of using a decision tree algorithm is the form of the questions that it is allowed to ask. We allow two basic types of information to be used as part of the context: numeric and categorical. For a numeric variable $N$, the decision tree searches for questions of the form "is $N >= n$", where $n$ is a numeric constant. For a categorical variable $C$, it searches over questions of the form: "is $C \in S$", where $S$ is a subset of the possible values of $C$. We also allow composite questions (Bahl et al. 1989), which are Boolean combinations of elementary questions.

**3.3.2 Questions about POS Tags.** The context that we use for estimating the probabilities includes both word identities and POS tags. To make effective use of this information, we allow the decision tree algorithm to generalize between words and POS tags that behave similarly. To learn which ones behave similarly, Black et al. (1992) and Magerman (1994) used the clustering algorithm of Brown et al. (1992) to build a hierarchical classification tree. Figure 2 gives the tree that we built for the POS tags. The algorithm starts with each POS tag in a separate class and iteratively finds two classes to merge that results in the smallest loss of information about POS adjacency. This continues until only a single class remains. The order in which classes were merged, however, gives a binary tree with the root corresponding to the entire

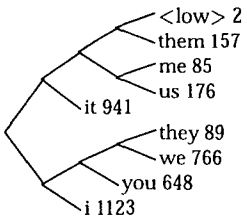


**Figure 3**
Binary classification tree that encodes the personal pronouns (**PRP**).

tagset, each leaf to a single POS tag, and intermediate nodes to groupings of the tags that are statistically similar. The path from the root to a tag gives the binary encoding for the tag. For instance, the binary encoding of **VBG** in Figure 2 is 01011100. The decision tree algorithm can ask which partition a tag belongs to by asking questions about its binary encoding.

**3.3.3 Questions about Word Identities.** For handling word identities, one could follow the approach used for handling the POS tags (e.g., Black et al. 1992; Magerman 1994) and view the POS tags and word identities as two separate sources of information. Instead, we view the word identities as a further refinement of the POS tags. We start the clustering algorithm with a separate class for each word and each tag that it takes on. Classes are only merged if the tags are the same. The result is a word classification tree for each tag. This approach means that the trees will not be polluted by words that are ambiguous as to their tag, as exemplified by the word *loads*, which is used in the corpus as a third-person present tense verb **VBZ** and as a plural noun **NNS**. Furthermore, this approach simplifies the clustering task because the hand annotations of the POS tags resolve a lot of the difficulty that the algorithm would otherwise have to learn. Hence, effective trees can be built even when only a small amount of data is available.

Figure 3 shows the classification tree for the personal pronouns (**PRP**). For reference, we also list the number of occurrences of each word for the POS tag. In the figure, we see that the algorithm distinguished between the subjective pronouns *I*, *we*, and *they*, and the objective pronouns *me*, *us*, and *them*. The pronouns *you* and *it* can take both cases and were probably clustered according to their most common usage in the corpus. The class **low** is used to group singleton words, which do not have enough training data to allow effective clustering. In using the word identities with the decision tree algorithm, we restrict the algorithm from asking word questions when the POS tag for the word is not uniquely determined by previous questions.

**3.3.4 Example Decision Tree.** Figure 4 illustrates the top part of the tree that was grown for estimating the probability distribution of the POS tag of the current word. The question on the root node "is $D^1_{i-1} = 0 \vee D^2_{i-1} = 1$" is asking whether the POS tag of the previous word has a 0 as the first bit or a 1 as the second bit of its binary encoding. If the answer is *no* then the bottom branch is followed, which corresponds to the following partition.

$$D_{i-1} \in \{\textbf{CC}, \textbf{PREP}, \textbf{JJ}, \textbf{JJS}, \textbf{JJR}, \textbf{CD}, \textbf{DT}, \textbf{PRP\$}, \textbf{WDT}\}$$

Following the bottom branch of the decision tree, we see that the next question is "is $D^3_{i-1} = 1$", which gives a true partition of $D_{i-1} \in \{\textbf{JJ}, \textbf{JJS}, \textbf{JJR}, \textbf{CD}, \textbf{DT}, \textbf{PRP\$}, \textbf{WDT}\}$. Following the top branch, we see that the next question is "is $D^4_{i-1} = 1$", whose true

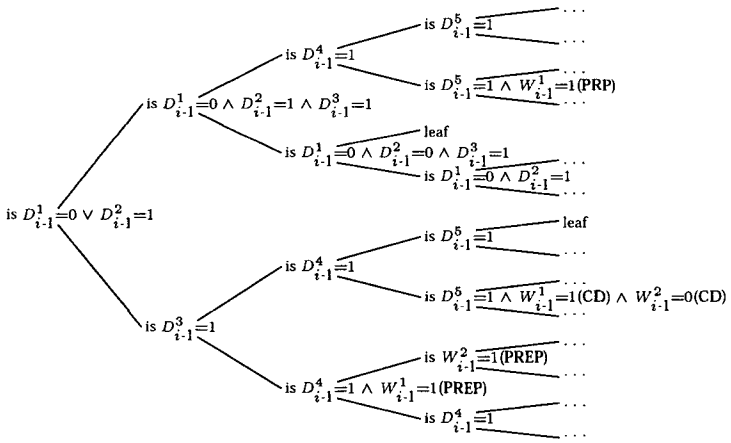


**Figure 4**
The top part of the decision tree used for estimating the POS probability distribution.

partition is $D_{i-1} \in \{\mathbf{DT}, \mathbf{PRP\$}, \mathbf{WDT}\}$. The next question along the top branch is "is $D^5_{i-1}=1$", which gives a true partition of $D_{i-1}=\mathbf{WDT}$. As indicated in the figure, this is a leaf node, and so no suitable question was found to ask of this context.

### 3.4 Results

To test our POS-based language model, we ran two experiments. The first set examines the effect of using richer contexts for estimating the word and POS probability distributions. The second set measures whether modeling discourse usage leads to better language modeling. Before we give the results, we explain the methodology that we use throughout the experiments.

**3.4.1 Experimental Setup.** In order to make the best use of our limited data, we tested our model using a sixfold cross-validation procedure. We divided the dialogues into six partitions and tested each partition with a model built from the other partitions. We divided the dialogues for each pair of speakers as evenly between the six partitions as possible. Changes in speaker are marked in the word transcription with the special token <**turn**>. The end-of-turn marker is not included in the POS results, but is included in the perplexity results. We treat contractions, such as *that'll* and *gonna*, as separate words, treating them as *that* and *'ll* for the first example, and *going* and *ta* for the second. We also changed all word fragments into a common token <**fragment**>.

Since current speech recognition rates for spontaneous speech are quite low, we have run the experiments on the hand-collected transcripts. In searching for the best sequence of POS tags for the transcribed words, we follow the technique proposed by Chow and Schwartz (1989) and only keep a small number of alternative paths by pruning the low probability paths after processing each word.

**3.4.2 Perplexity.** A way to measure the effectiveness of the language model is to measure the **perplexity** that it assigns to a test corpus (Bahl et al. 1977). Perplexity is an estimate of how well the language model is able to predict the next word of a test corpus in terms of the number of alternatives that need to be considered at each point. For word-based language models, with estimated probability distribution of $\hat{\Pr}(w_i|w_{1,i-1})$, the perplexity of a test set $w_{1,N}$ is calculated as $2^H$, where $H$ is the entropy, which is defined as $H = -\frac{1}{N}\sum_{i=1}^{N} \log_2 \hat{\Pr}(w_i|w_{1,i-1})$.

**Table 6**
Using richer histories to estimate probabilities.

| | $\Pr(W_i\|D_i)$ $\Pr(D_i\|D_{i-2,i-1})$ | $\Pr(W_i\|D_{i-2,i}W_{i-2,i-1})$ $\Pr(D_i\|D_{i-2,i-1}W_{i-2,i-1})$ |
|---|---|---|
| POS Errors | 1,778 | 1,711 |
| POS Error Rate | 3.04 | 2.93 |
| DM Errors | 690 | 630 |
| DM Error Rate | 8.33 | 7.61 |
| DM Recall | 95.86 | 96.75 |
| DM Precision | 95.79 | 95.68 |
| Word Perplexity | 43.22 | 24.04 |
| Branching Perplexity | 47.25 | 26.35 |

*Branching Perplexity.* Our POS-based model is not only predicting the next word, but its POS tag as well. To estimate the branching factor, and thus the size of the search space, we use the following formula for the entropy, where $d_i$ is the POS tag for word $w_i$.

$$H = -\frac{1}{N}\sum_{i=1}^{N} \log_2 \hat{\Pr}(w_i|w_{1,i-1}d_{1,i})\hat{\Pr}(d_i|w_{1,i-1}d_{1,i-1}) \tag{10}$$

*Word Perplexity.* In order to compare a POS-based model against a word-based language model, we should not penalize the POS-based model for incorrect POS tags. Hence, we should ignore them when defining the perplexity and base the perplexity measure on $\hat{\Pr}(w_i|w_{1,i-1})$. However, for our model, this probability is not estimated. Hence, we must rewrite it in terms of the probabilities that we do estimate. To do this, our only recourse is to sum over all possible POS sequences.

$$H = -\frac{1}{N}\sum_{i=1}^{N} \log_2 \frac{\sum_{D_{1,i}} \hat{\Pr}(w_iD_i|w_{1,i-1}D_{1,i-1})\hat{\Pr}(w_{1,i-1}D_{1,i-1})}{\sum_{D_{1,i-1}} \hat{\Pr}(w_{1,i-1}D_{1,i-1})} \tag{11}$$

**3.4.3 Recall and Precision.** We report results on identifying discourse markers in terms of *recall, precision* and *error rate.* The recall rate is the number of times that the algorithm correctly identifies an event over the total number of times that it actually occurred. The precision rate is the number of times the algorithm correctly identifies it over the total number of times it identifies it. The error rate is the number of errors in identifying an event over the number of times that the event occurred.

**3.4.4 Using Richer Histories.** Table 6 shows the effect of varying the richness of the information that the decision tree algorithm is allowed to use in estimating the POS and word probabilities. The second column uses the approximations given in Equation 8 and 9 and the third column uses the full context. The results show that adding the extra context has the biggest effect on the perplexity measures, decreasing the word perplexity by 44.4% from 43.22 to 24.04. The effect on POS tagging is less pronounced, but still gives a reduction of 3.8%. We also see a 8.7% improvement in identifying discourse markers. Hence, in order to use POS tags in a speech recognition language model, we need to use a richer context for estimating the probabilities than what is typically used. In other work (Heeman 1999), we show that our POS-based model results in lower perplexity and word error rate than a word-based model.

**Table 7**
Effect of modeling discourse markers with special POS tags.

| | WP | WD |
|---|---|---|
| POS Errors | 1,219 | 1,189 |
| POS Error Rate | 2.09 | 2.04 |
| Word Perplexity | 24.20 | 24.04 |
| Branching Perplexity | 26.08 | 26.35 |

**3.4.5 Modeling Discourse Markers.** Table 7 shows the effect of modeling discourse markers by using special POS tags. In column two, we give the results of a model in which we use a POS tagset that does not distinguish discourse marker usage (**P**). The discourse conjuncts **CC_D** are collapsed into **CC**, discourse adverbials **RB_D** into **RB**, and acknowledgments **AC** and discourse interjections **UH_D** into **UH_FP**. The third column gives the results of the model in which we use our tagset that does distinguish discourse marker usage (**D**). To ensure a fair comparison, we do not penalize POS errors that result from a confusion between discourse and sentential usages. We see that modeling discourse markers results in a perplexity reduction from 24.20 to 24.04 and reduces the number of POS errors from 1,219 to 1,189, giving a 2.5% error rate reduction. Although the improvements in perplexity and POS tagging are small, they indicate that there are interactions, and hence discourse markers should be resolved at the same time as POS tagging and speech recognition word prediction.

## 4. Identifying Speech Repairs and Intonational Phrases

In the previous section, we presented a POS-based language model that uses special tags to denote discourse markers. However, this model does not account for the occurrence of speech repairs and intonational phrases. Ignoring these events when building a statistical language model will lead to probabilistic estimates for the words and POS tags that are less precise, since they mix contexts that cross intonational boundaries and interruption points of speech repairs with fluent stretches of speech. However, there is not a reliable signal for detecting the interruption point of speech repairs (Bear, Dowding, and Shriberg 1992) nor the occurrence of intonational phrases. Rather, there are a number of different sources of information that give evidence as to the occurrence of these events. These sources include the presence of pauses, filled pauses, cue phrases, discourse markers, word fragments, word correspondences, and syntactic anomalies. Table 8 gives the number of occurrences for some of these features for each word in the corpus that is not turn-final nor part of the editing term of a speech repair. Each word is classified by whether it immediately precedes the interruption point of a fresh start, modification, or abridged repair, or ends an intonational phrase. All other words are categorized as fluent. The first row gives the number of occurrences of these events. The second row reports whether the word is a fragment. The third and fourth give the number of times the word is followed by a filled pause or discourse marker, respectively. The fifth and sixth rows report whether the word is followed by a pause that is less than or greater than 0.5 seconds, respectively. Pause durations were computed automatically with a speech recognizer constrained to the word transcription (Entropic Research Laboratory, Inc. 1994). The next row reports whether there is a word match that crosses the word with at most two intervening words, and the next row, those with at most five intervening words.

**Table 8**
Occurrence of features that signal speech repairs and intonational boundaries.

| Feature | Fluent Speech | Abridged Repairs | Modification Repairs | Fresh Starts | Intonational Boundaries |
|---|---|---|---|---|---|
| All | 43,439 | 423 | 1,301 | 671 | 5,211 |
| Fragments | 7 | 0 | 481 | 150 | 0 |
| Filled Pauses | 97 | 374 | 114 | 71 | 358 |
| Short Pauses | 4,415 | 146 | 711 | 313 | 1,710 |
| Long Pauses | 1,537 | 121 | 176 | 186 | 1,622 |
| Matching (2) | 2,629 | 27 | 869 | 197 | 373 |
| Matching (5) | 11,479 | 94 | 1,517 | 575 | 1,375 |

From the table, it is clear that none of the cues on their own is a reliable indicator of speech repairs or intonational boundaries. For instance, 44.5% (1,622/3,642) of all long pauses occur after an intonational boundary and 13.3% occur after the interruption point of a speech repair. Conversely, 31.1% (1,622/5,211) of intonational boundaries are followed by a pause while 20.2% of all repairs are followed by a long pause. Hence, pauses alone do not give a complete picture of whether a speech repair or intonational boundary occurred. The same holds for filled pauses, which can occur both after the interruption point of a speech repair and in fluent speech, namely between utterances or after utterance-initial discourse markers. Word matchings can also be spurious, as evidenced by the 27 word matches with at most two intervening words across abridged repairs, as well as the matchings across intonational boundaries and fluent speech. Even syntactic ill-formedness at the interruption point is not always guaranteed, as the following example illustrates.

**Example 15** (d93-13.2 utt53)

load two $\underbrace{\text{boxes of}}_{\textit{reparandum}}$ $\underset{\textit{ip}}{\uparrow}$ boxcars with oranges

Hence using parser failures to find repairs (cf. Dowding et al. 1993) will not be robust.

In this section, we augment our POS-based language model so that it also detects intonational boundaries and speech repairs, along with their editing terms. Although not all speech repairs have obvious syntactic anomalies, the probability distributions for words and POS tags are going to be different depending on whether they follow the interruption point of a speech repair, an intonational boundary, or fluent speech. So, it makes sense to take the speech repairs and intonational boundaries into account by directly modeling them when building the language model, which automatically gives us a means of detecting these events and better prediction of the speech that follows. To model the occurrence of intonational boundaries and speech repairs, we introduce three extra variables into the language model. The **repair** tag $R_i$, the **editing term** tag $E_i$ and the **intonation tag** $I_i$. These **utterance tags** capture the discontinuities in the speaker's turn, and we use these discontinuities to better model the speech that follows.

### 4.1 Speech Repairs

The repair tag indicates the occurrence of speech repairs. However, we not only want to know whether a repair occurred, but also the type of repair: whether it is a modification

repair, a fresh start, or an abridged repair. The type of repair is important since the strategy that a hearer uses to correct the repair depends on the type of repair. For fresh starts, the hearer must determine the beginning of the current utterance. For modification repairs, the hearer can make use of the correspondences between the reparandum and alteration to determine the reparandum onset. For abridged repairs, there is no reparandum, and so simply knowing that it is abridged gives the correction.

For repairs that do not have an editing term, the interruption point is where the local context is disrupted, and hence is the logical place to tag such repairs. For repairs with an editing term, there are two choices for marking the speech repair: either directly following the end of the reparandum, or directly preceding the onset of the alteration. The following example illustrates these two choices, marking them with **Mod?**.

**Example 16** (d92a-5.2 utt34)

so we'll pick up a tank of **Mod?** uh **Mod?** the tanker of oranges

The editing term by itself does not completely determine the type of repair. The alteration also helps to disambiguate the repair. Hence, we delay hypothesizing about the repair type until the end of the editing term, which should keep our search-space smaller, since we do not need to keep alternative repair type interpretations while processing the editing term. This leads to the following definition of the repair variable $R_i$ for the transition between word $W_{i-1}$ and $W_i$:

$$R_i = \begin{cases} \textbf{Mod} & \text{if } W_i \text{ is the alteration onset of a modification repair} \\ \textbf{Can} & \text{if } W_i \text{ is the alteration onset of a fresh start (or } \textit{cancel}) \\ \textbf{Abr} & \text{if } W_i \text{ is the alteration onset of an abridged repair} \\ \textbf{null} & \text{otherwise} \end{cases}$$

### 4.2 Editing Terms

Editing terms are problematic for tagging speech repairs since they separate the end of the reparandum from the alteration onset, thus separating the discontinuity that gives evidence that a fresh start or modification repair occurred. For abridged repairs, they separate the word that follows the editing term from the context that is needed to determine the identity of the word and its POS tag. If editing terms could be identified without having to consider the context, we could skip over them, but still use them as part of the context for deciding the repair tag (cf. Heeman and Allen 1994). However, this assumption is not valid for words that are ambiguous as to whether they are an editing term, such as *let me see*. Even filled pauses are problematic since they are not necessarily part of the editing term of a repair. To model editing terms, we use the variable $E_i$ to indicate the type of editing term transition between word $W_{i-1}$ and $W_i$.

$$E_i = \begin{cases} \textbf{Push} & \text{if } W_{i-1} \text{ is not part of an editing term but } W_i \text{ is} \\ \textbf{ET} & \text{if } W_{i-1} \text{ and } W_i \text{ are both part of an editing term} \\ \textbf{Pop} & \text{if } W_{i-1} \text{ is part of an editing term but } W_i \text{ is not} \\ \textbf{null} & \text{if neither } W_{i-1} \text{ nor } W_i \text{ are part of an editing term} \end{cases}$$

Below, we give an example and show all non-null editing term and repair tags.

**Example 17** (d93-10.4 utt30)

that'll get there at four a.m. **Push** oh **ET** sorry **Pop Mod** at eleven a.m.

### 4.3 Intonational Phrases

The final variable is $I_i$, which marks the occurrence of intonational phrase boundaries.

$$I_i = \begin{cases} \% & \text{if } W_{i-1} \text{ ends an intonational phrase} \\ \textbf{null} & \text{otherwise} \end{cases}$$

The intonation variable is separate from the editing term and repair variables since it is not restricted by the value of the other two. For instance, an editing term could end an intonational phrase, especially on the end of a cue phrase such as *let's see*, as can the reparandum, as Example 18 below demonstrates.

**Example 18** (d92a-2.1 utt29)

that's the one with the bananas % **Push I ET** mean **Pop Mod** that's taking the bananas

### 4.4 Redefining the Speech Recognition Problem

We now redefine the speech recognition problem so that its goal is to find the sequence of words and the corresponding POS, intonation, editing term, and repair tags that is most probable given the acoustic signal.

$$\begin{aligned} \hat{W}\hat{D}\hat{R}\hat{E}\hat{I} &= \arg\max_{WDREI} \Pr(WDREI|A) \\ &= \arg\max_{WDREI} \Pr(A|WDREI)\Pr(WDREI) \end{aligned} \tag{12}$$

The second term is the language model probability, and can be rewritten as follows.

$$\begin{aligned} \Pr(W_{1,N}D_{1,N}R_{1,N}E_{1,N}I_{1,N}) &= \prod_{i=1}^{N} \Pr(W_iD_iR_iE_iI_i|W_{1,i-1}D_{1,i-1}R_{1,i-1}E_{1,i-1}I_{1,i-1}) \\ &= \prod_{i=1}^{N} \Pr(I_i|W_{1,i-1}D_{1,i-1}R_{1,i-1}E_{1,i-1}I_{1,i-1}) \\ &\quad \Pr(E_i|W_{1,i-1}D_{1,i-1}R_{1,i-1}E_{1,i-1}I_{1,i}) \\ &\quad \Pr(R_i|W_{1,i-1}D_{1,i-1}R_{1,i-1}E_{1,i}I_{1,i}) \\ &\quad \Pr(D_i|W_{1,i-1}D_{1,i-1}R_{1,i}E_{1,i}I_{1,i}) \\ &\quad \Pr(W_i|W_{1,i-1}D_{1,i}R_{1,i}E_{1,i}I_{1,i}) \end{aligned} \tag{13}$$

### 4.5 Representing the Context

Equation 13 requires five probability distributions to be estimated. The context for each includes all of the words, POS, intonation, repair, and editing term tags that have been hypothesized, each as a separate piece of information. In principal, we could give this to the decision tree algorithm and let it decide what information to use in constructing equivalence classes. However, repairs, editing terms, and even intonation phrases do not occur in the same abundance as fluent speech and are not as constrained. Hence, it will be difficult to model the discontinuities that they introduce into the context.

Consider the following example of a speech repair without an editing term.

**Example 19** (d92-1 utt53)

engine E two picks **Mod** takes the two boxcars

When predicting the first word of the alteration *takes*, it is inappropriate to ask about the preceding words, such as *picks*, without realizing that there is a modification repair in between. The same also holds for intonational boundaries and editing term pushes and pops. In the example below, a question should only be asked about *is* in the realization that it ends an intonational phrase.

**Example 20** (d92a-1.2 utt3)

you'll have to tell me what the problem is % I don't have their labels

Although the intonation, repair, and editing term tags are part of the context and so can be used in partitioning it, the question is whether this will happen. The problem is that null intonation, repair, and editing term tags dominate the training examples. So, we are bound to run into contexts in which there are not enough intonational phrases and repairs for the algorithm to learn the importance of using this information, and instead might blindly subdivide the context based on some subdivision of the POS tags. The solution is analogous to what is done in POS tagging of written text: we give a view of the words and POS tags with the non-null repair, non-null intonation, and editing term push and pop tags inserted. By inserting these tags into the word and POS sequence, it will be more difficult for the learning algorithm to ignore them. It also allows these tags to be grouped with other tags that behave in a similar way, such as change in speaker turn, and discourse markers.

Now consider the following examples, which both start with *so we need to*.

**Example 21** (d92a-2.2 utt6)

so we need to **Push** um **Pop Abr** get a tanker of OJ to Avon

**Example 22** (d93-11.1 utt46)

so we need to get the three tankers

This is then followed by the verb *get*, except the first has an editing term in between. However, in predicting this word, the editing term hinders the decision tree algorithm from generalizing with nonabridged examples. The same thing happens with fresh starts and modification repairs. To allow generalizations between repairs with an editing term and those without, we need a view of the context with completed editing terms removed (cf. Stolcke and Shriberg 1996b).

Part of the context given to the decision tree is the words and POS tags with the non-null utterance tags inserted (i.e., %) and completed editing terms removed. We refer to this as the **utterance context**, since it incorporates the utterance information that has been hypothesized. Consider the following example.

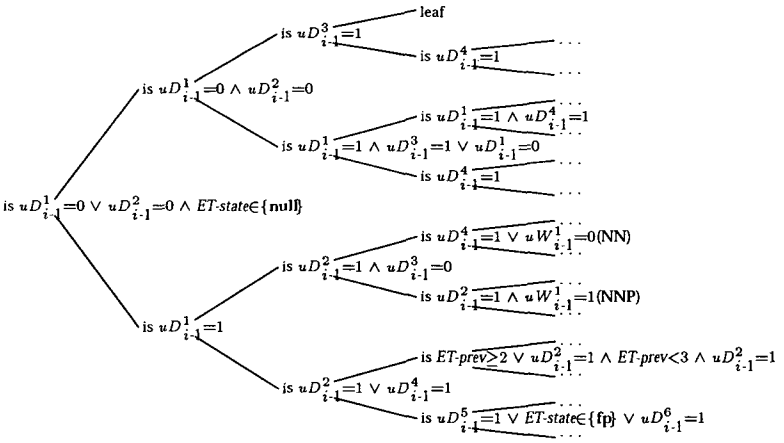


**Figure 5**
Top part of the decision tree used for estimating the probability distribution of the intonation tag.

**Example 23** (d93-18.1 utt47)

it takes one **Push** you **ET** know **Pop Mod** two hours %

The utterance context for the POS tag of *you* is "it/**PRP** takes/**VBP** one/**CD Push**." The context for the editing term **Pop** is "it/**PRP** takes/**VBP** one/**CD Push** you/**PRP** know/**VBP**." The utterance context for the repair tag has the editing term cleaned up: "it/**PRP** takes/**VBP** one/**CD**" (we also give it the context with the editing term not cleaned up). The context for the POS tag of *two* is "it/**PRP** takes/**VBP** one/**CD Mod**."

We also include two variables that indicate whether we are processing an editing term without forcing it to look for an editing term **Push** in the utterance context: **ET-state** indicates whether we are processing an editing term and whether a cue phrase was seen; and **ET-prev** indicates the number of editing term words seen so far. Figure 5 gives the top part of the decision tree that was grown for the intonation tag, where $uW$ and $uD$ are the utterance context.

## 5. Correcting Speech Repairs

The previous section focused on the detection of speech repairs, editing terms, and intonational phrases. But for repairs, we have only addressed half of the problem; the other half is determining the extent of the reparandum. Hindle (1983) and Kikui and Morimoto (1994) both separate the task of correcting a repair from detecting it by assuming that there is an acoustic editing signal that marks the interruption point of speech repairs (as well as access to the POS tags and utterance boundaries). Although the model of the previous section detects repairs, this model is not effective enough. In fact, we feel that one of its crucial shortcomings is that it does not take into consideration the task of correcting repairs (Heeman, Loken-Kim, and Allen 1996). Since hearers are often unaware of speech repairs (Martin and Strange 1968), they must be able to correct them as the utterance is unfolding and as an indistinguishable event from detecting them and recognizing the words involved.

Bear, Dowding, and Shriberg (1992) proposed that multiple information sources need to be combined in order to detect and correct speech repairs. One of these sources

**Table 9**
Occurrences of common repair structures.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| x. | 362 | mmr.mmr | 10 | mm.mxm | 4 | mxmx.mm | 2 |
| m.m | 249 | m.xm | 10 | xmmm.mmm | 3 | mmm.mmxm | 2 |
| r.r | 136 | mxx.m | 8 | mrx.mr | 3 | mm.xxmm | 2 |
| mm.mm | 85 | mmmx.mmm | 8 | mrr.mrr | 3 | mm.mxxm | 2 |
| mx.m | 76 | m.xxm | 8 | mrmx.mrm | 3 | xr.r | 2 |
| mmx.mm | 35 | mrm.mrm | 7 | mmmmm.mmmmm | 3 | xmx.m | 2 |
| mr.mr | 29 | mx.xm | 6 | mm.xmm | 3 | xmmx.mm | 2 |
| mmm.mmm | 22 | xm.m | 5 | mmmmx.mmmm | 2 | rr.rr | 2 |
| rx.r | 20 | mmmmr.mmmmr | 5 | mrmm.mrmm | 2 | rm.rxm | 2 |
| rm.rm | 20 | rmm.rmm | 4 | mmmxx.mmm | 2 | r.xr | 2 |
| xx. | 12 | mmxx.mm | 4 | mmm.xxxmmm | 2 | | |
| mmmm.mmmm | 12 | mmmr.mmmr | 4 | mmm.mxmm | 2 | | |

includes a pattern-matching routine that looks for simple cases of word correspondences that could indicate a repair. However, pattern matching is too limited to capture the variety of word correspondence patterns that speech repairs exhibit (Heeman and Allen 1994). For example, the 1,302 modification repairs in the Trains corpus take on 160 different repair structures, even when we exclude word fragments and editing terms. Of these, only 47 occurred at least twice, and these are listed in Table 9. Each word in the repair is represented by its correspondence type: **m** for word match, **r** for replacement, and **x** for deletions and insertions. A period "." marks the interruption point. For example, the structure of the repair given in Example 14 (*engine two from Elmi(ra)- or engine three from Elmira*) would be **mrm.mrm**.

To remedy the limitation of Bear, Dowding, and Shriberg (1992), we proposed that the word correspondences between the reparandum and alteration could be found by a set of well-formedness rules (Heeman and Allen 1994; Heeman, Loken-Kim, and Allen 1996). Potential repairs found by the rules were passed to a statistical language model (a predecessor of the model of Section 4), which pruned out false positives. Part of the context for the statistical model was the proposed repair structure found by the well-formedness rules. However, the alteration of a repair, which makes up half of the repair structure, occurs after the interruption point and hence should not be used to predict the occurrence of a repair. Hence this model was of limited use for integration into a speech recognizer.

Recently, Stolcke and Shriberg (1996) presented a word-based model for speech recognition that models simple word deletion and repetition patterns. They used the prediction of the repair to clean up the context and to help predict what word will occur next. Although their model is limited to simple types of repairs, it provides a starting point for incorporating speech repair correction into a statistical language model.

### 5.1 Sources of Information

There are several sources of information that give evidence as to the extent of the reparandum of speech repairs. Probably the most widely used is the presence of word correspondences between the reparandum and alteration, both at the word level and at the level of syntactic constituents (Levelt 1983; Hindle 1983; Bear, Dowding, and Shriberg 1992; Heeman and Allen 1994; Kikui and Morimoto 1994). Second, there tends to be a fluent transition from the speech that precedes the onset of the reparandum to the alteration (Kikui and Morimoto 1994). This source is very important for repairs that do not have initial retracing, and is the mainstay of the "parser-first" approach (e.g.,

Dowding et al. 1993)—keep trying alternative corrections until one of them parses. Third, there are certain regularities for where speakers restart. Reparandum onsets tend to be at constituent boundaries (Nooteboom 1980), and in particular, at boundaries where a coordinated constituent can be placed (Levelt 1983). Hence, reparandum onsets can be partially predicted without even looking at the alteration.

### 5.2 Our Approach

Most previous approaches to correcting speech repairs have taken the standpoint of finding the best reparandum given the neighboring words. Instead, we view the problem as finding the reparandum that best predicts the following words. Since speech repairs are often accompanied by word correspondences (Levelt 1983; Hindle 1983; Bear, Dowding, and Shriberg 1992; Heeman and Allen 1994; Kikui and Morimoto 1994), the actual reparandum will better predict the words in the alteration of the repair. Consider the following example:

**Example 24** (d93-3.2 utt45)

which engine $\underbrace{\text{are we}}_{\textit{reparandum}}$ $\uparrow_{\textit{ip}}$ are we taking

In this example, if we predicted that a modification repair occurred and that the reparandum consists of *are we*, then the probability of *are* being the first word of the alteration would be very high, since it matches the first word of the reparandum. Conversely, if we are not predicting a modification repair with reparandum *are we*, then the probability of seeing *are* would be much lower. The same reasoning holds for predicting the next word, *we*: it is much more likely under the repair interpretation. So, as we process the words of the alteration, the repair interpretation will better account for the words that follow it, strengthening the interpretation.

When predicting the first word of the alteration, we can also make use of the second source of evidence identified in the previous section: the context provided by the words that precede the reparandum. Consider the following repair in which the first two words of the alteration are inserted.

**Example 25** (d93-16.2 utt66)

and two tankers $\underbrace{\text{to}}_{\textit{reparandum}}$ $\uparrow_{\textit{ip}}$ of OJ to Dansville

Here, if we know the reparandum is *to*, then we know that the first word of the reparandum must be a fluent continuation of the speech before the onset of the reparandum. In fact, we see that the repair interpretation (with the correct reparandum onset) provides better context for predicting the first word of the alteration than a hypothesis that predicts either the wrong reparandum onset or predicts no repair at all. Hence, by predicting the reparandum of a speech repair, we no longer need to predict the onset of the alteration on the basis of the ending of the reparandum, as we did in Section 4.5. Such predictions are based on limited amounts of training data since only examples of speech repairs can be used. Rather, by first predicting the reparandum, we can use examples of fluent transitions to help predict the first word of the alteration.

We can also make use of the third source of information. When we initially hypothesize the reparandum onset, we can take into account the a priori probability

that it will occur at that point. In the following example, the words *should* and *the* are preferred by Levelt's coordinated constituent rule (Levelt 1983), and hence should have a higher score. Exceptions to the rule, such as this one, should have a lower score.

**Example 26** (d93-10.4 utt30)

two boxcars $\underbrace{\text{of orange juice should}}_{\textit{reparandum}}\ \overset{\uparrow}{\underset{ip}{}}$ er of oranges should be made into orange juice

To incorporate correction processing into our language model, we need to add extra variables. After we predict a repair, we need to predict the reparandum onset. Knowing the reparandum onset then allows us to predict the word correspondences between the reparandum and alteration, thus allowing us to use the repair to better predict the words and their POS tags that make up the alteration.

### 5.3 Reparandum Onset

After we predict a modification repair or a fresh start, we need to predict the reparandum onset. Consider the following two examples of modification repairs.

**Example 27** (d93-16.3 utt9)

to fill $\underbrace{\text{the engine}}_{\textit{reparandum}}\ \overset{\uparrow}{\underset{ip}{}}$ the boxcars with bananas

**Example 28** (d93-25.6 utt31)

drop off $\underbrace{\text{the one tanker}}_{\textit{reparandum}}\ \overset{\uparrow}{\underset{ip}{}}$ the two tankers

Although the examples differ in the length of the reparandum, their reparanda both start at the onset of a noun phrase. This same phenomena also exists for fresh starts where reparandum onsets are likely to follow an intonational boundary, the beginning of the turn, or a discourse marker. In order to allow generalizations across different reparandum lengths, we query each potential onset to see how likely it is as the onset. For $R_i \in \{\textbf{Mod}, \textbf{Can}\}$ and $j < i$, we define $O_{ij}$ as follows:

$$O_{ij} = \begin{cases} \textbf{Onset} & W_j \text{ is the reparandum onset of repair } R_i \\ \textbf{null} & \text{otherwise} \end{cases}$$

We normalize the probabilities to ensure that $\sum_j (O_{ij} = \textbf{Onset}) = 1$.

Just as we exclude the editing terms of previous repairs from the utterance words and POS tags, so we exclude the reparanda of previous repairs. Consider the following example of **overlapping** repairs, repairs in which the reparanda and alterations cannot be separated.

**Example 29** (d93-16.3 utt4)

what's the shortest route from engine $\underset{ip}{\uparrow}$ from $\underset{ip}{\uparrow}$ for engine two at Elmira

The reparandum of the first repair is *from engine*. In predicting the reparandum of the second, we work from the cleaned up context: *what's the shortest route from*.

The context used in estimating how likely a word is as the reparandum onset includes which word we are querying. We also include the utterance words and POS tags that precede the proposed reparandum onset, thus allowing the decision tree to check if the onset is at a suitable constituent boundary. Since reparanda rarely extend over more than one utterance, we include three variables that help indicate whether an utterance boundary is being crossed. The first indicates the number of intonational phrase boundaries embedded in the proposed reparandum. The second indicates the number of discourse markers in the reparandum. Discourse markers at the beginning of the reparandum are not included, and if discourse markers appear consecutively, the group is only counted once. The third indicates the number of filled pauses in the reparandum.

Another source of information is the presence of other repairs in the turn. In the Trains corpus, 35.6% of nonabridged repairs overlap. If a repair overlaps a previous one then its reparandum onset is likely to co-occur with the alteration onset of the previous repair (Heeman 1997). Hence we include a variable that indicates whether there is a previous repair, and if there is, whether the proposed onset coincides with, precedes, or follows the alteration onset of the preceding repair.

### 5.4 The Active Repair

Determining word correspondences is complicated by the occurrence of overlapping repairs. To keep our approach simple, we allow at most one previous word to license the correspondence. Consider again Example 29. Here, one could argue that the word *for* corresponds to the word *from* from either the reparandum of the first or second repair. In either case, the correspondence to the word *engine* is from the reparandum of the first repair. Our approach is to first decide which repair the correspondence will be to and then decide which word of that repair's reparandum will license the current word. We always choose the most recent repair that has words in its reparandum that have not yet licensed a correspondence (other than a word fragment). Hence, the active repair for predicting the word *for* is the second repair, while the active repair for predicting *engine* is the first repair. For predicting the word *two*, neither the first nor second repair has any unlicensed words in its reparandum, and hence *two* will not have an active repair. In future work, we plan to choose between the reparandum of alternative speech repairs, as allowed by the annotation scheme (Heeman 1997).

### 5.5 Licensing a Correspondence

If we are in the midst of processing a repair, we can use the reparandum to help predict the current word $W_i$ and its POS tag $D_i$. In order to do this, we need to determine which word in the reparandum of the active repair will license the current word. As illustrated in Figure 6, word correspondences for speech repairs tend to exhibit a cross serial dependency (Heeman and Allen 1994); in other words, if we have a correspondence between $w_j$ in the reparandum and $w_k$ in the alteration, any correspondence with a word in the alteration after $w_k$ will be to a word that is after $w_j$.

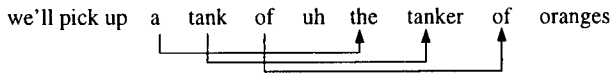


**Figure 6**
Cross serial correspondences between reparandum and alteration.

This regularity does have exceptions, as the following example illustrates; however, we currently do not support such correspondences.

**Example 30** (d93-19.4 utt37)

can we have ↑ we can have three engines in Corning at the same time
*reparandum* *ip*

Since we currently do not support such exceptions, this means that if there is already a correspondence for the repair, then the licensing word will follow the last correspondence in the reparandum.

The licensing word might need to skip over words due to deleted words in the reparandum or inserted words in the alteration. In the example below, the word *tow* is licensed by *carry*, but the word *them* must be skipped over before processing the licensing between the two instances of *both*.

**Example 31** (d92a-1.2 utt40)

you can carry them both on ↑ tow both on the same engine
*reparandum* *ip*

The next example illustrates the opposite problem: the word *two* has no correspondence with any word in the reparandum.

**Example 32** (d93-15.4 utt45)

and fill my boxcars full of oranges ↑ my two boxcars full of oranges
*reparandum* *ip*

For words that have no correspondence, we define the licensing word as the first available word in the alternation, in this case *boxcars*. We leave it to the correspondence variable to encode that there is no correspondence. This gives us the following definition for the correspondence licensor, $L_{ij}$, where $i$ is the current word and $j$ runs over all words in the reparandum of the active repair that come after the last word in the reparandum with a correspondence.

$$L_{ij} = \begin{cases} \textbf{Corr} & W_j \text{ licenses the current word} \\ \textbf{Corr} & W_i \text{ is an inserted word and } W_j \text{ is first available word in reparandum} \\ \textbf{null} & \text{otherwise} \end{cases}$$

Just as with the reparandum onset, we estimate the probability by querying each eligible word. The context for this query includes information about the proposed word, namely its POS tag, as well as the utterance POS and word context prior to the current word, the type of repair and the reparandum length. We also include

information about the repair structure that has been found so far. If the previous word was a word match, there is a good chance that the current word will involve a word match to the next word. The rest of the features are the number of words skipped in the reparandum and alteration since the last correspondence, the number of words since the onset of the reparandum and alteration, and the number of words to the end of the reparandum.

### 5.6 The Word Correspondence

Now that we have decided which word in the reparandum will potentially license the current word, we need to predict the type of correspondence. We focus on correspondences involving exact word match (identical POS tag and word), word replacements (same POS tag), or no such correspondence.

$$C_i = \begin{cases} \textbf{m} & W_i \text{ is a word match of the word indicated by } L_i \\ \textbf{r} & W_i \text{ is a word replacement of the word indicated by } L_i \\ \textbf{x} & W_i \text{ has no correspondence (inserted word)} \\ \textbf{null} & \text{No active repair} \end{cases}$$

The context used for estimating the correspondence variable is exactly the same as that used for estimating the licensor.

### 5.7 Redefining the Speech Recognition Problem

Now that we have introduced the correction tags, we redefine the speech recognition problem so that it includes finding the most probable corrections tags.

$$\begin{aligned} \hat{W}\hat{D}\hat{C}\hat{L}\hat{O}\hat{R}\hat{E}\hat{I} &= \arg\max_{WDCLOREI} \Pr(WDCLOREI|A) \\ &= \arg\max_{WDCLOREI} \Pr(A|WDCLOREI)\Pr(WDCLOREI) \end{aligned} \quad (14)$$

The second term is the language model and can be rewritten as we did for Equation 12.

We have already discussed the context used for estimating the three new probability distributions. We also have a richer context for estimating the other five distributions. For these, we take advantage of the new definition of the utterance word and POS tags, which now accounts for the reparanda of repairs. Consider the following example.

**Example 33** (d93-13.1 utt64)

pick up and load two um the two boxcars on engine two

*reparandum* *ip*

In processing the word *the*, if we hypothesized that it follows a modification repair with editing term *um* and reparandum *two*, then we can now generalize with fluent examples, such as the following, in hypothesizing its POS tag and the word identity.

**Example 34** (d93-12.4 utt97)

and to make the orange juice and load the tankers

Thus, we can make use of the second knowledge source of Section 5.1.

Cleaning up fresh starts requires a slightly different treatment. Fresh starts abandon the current utterance, and hence the alteration starts a new utterance. But this new utterance will start differently than most utterances in that it will not begin with initial filled pauses, or phrases such as *let's see*, since these would have been counted as part of the editing term of the fresh start. Hence, when we clean up the reparanda of fresh starts, we leave the fresh start marker **Can**, just as we do for intonational boundaries.

For predicting the word and POS tags, we have an additional source of information, namely the values of the correspondence licensor and the correspondence type. Rather than use these two variables as part of the context that we give the decision tree algorithm, we use these tags to override the decision tree probability. If a word replacement or word match was hypothesized, we assign all of the POS probability to the appropriate POS tag. If a word match was hypothesized, we assign all of the word probability to the appropriate word.

## 6. Acoustic Cues

Silence, as well as other acoustic information, can also give evidence as to whether an intonational phrase, speech repair, or editing term occurred, as was shown in Table 8. In this section, we revise the language model to incorporate this information.

### 6.1 Redefining the Speech Recognition Problem

In the same way that speech recognizers hypothesize lexical items, they also hypothesize pauses. Rather than insert these into the word sequence (e.g., Zeppenfeld et al. 1997), we define the variable $S_i$ to be the amount of silence between words $W_{i-1}$ and $W_i$. We incorporate this information by redefining the speech recognition problem.

$$\hat{W}\hat{P}\hat{C}\hat{L}\hat{O}\hat{R}\hat{E}\hat{I}\hat{S} = \underset{WDCLOREIS}{\arg\max}\ \Pr(A|WDCLOREIS)\Pr(WDCLOREIS) \tag{15}$$

Again, the first term is the acoustic model, which one can approximate by $\Pr(A|WS)$, and thus reduce it to a traditional acoustic model. The second term is the new language model, which we rewrite as follows:

$$\begin{aligned}&\Pr(W_{1,N}D_{1,N}C_{1,N}L_{1,N}O_{1,N}R_{1,N}E_{1,N}I_{1,N}S_{1,N})\\ &= \prod_{i=1}^{N}\Pr(W_iD_iC_iL_iO_iR_iE_iI_iS_i|W_{1,i-1}D_{1,i-1}C_{1,i-1}L_{1,i-1}O_{1,i-1}R_{1,i-1}E_{1,i-1}I_{1,i-1}S_{1,i-1})\end{aligned}$$

We expand the silence variable first so that we can use it as part of the context in estimating the tags for the remaining variables.

We now have an extra probability in our model, namely the probability of $S_i$ given the previous context. The variable $S_i$ will take on values in accordance with the minimum time samples that the speech recognizer uses. To deal with limited amounts of training data, one could collapse these durations into larger intervals. Note that including this probability impacts the perplexity computation. Usually, prediction of silence durations is not included in the perplexity calculation. In order to allow comparisons between the perplexity rates of the model that includes silence durations and ones that do not, we exclude the probability of $S_i$ in the perplexity calculation.

### 6.2 Using Silence as Part of the Context

We now need to include the silence durations as part of the context for predicting the values of the other variables. However, it is just for the intonation, repair, and editing

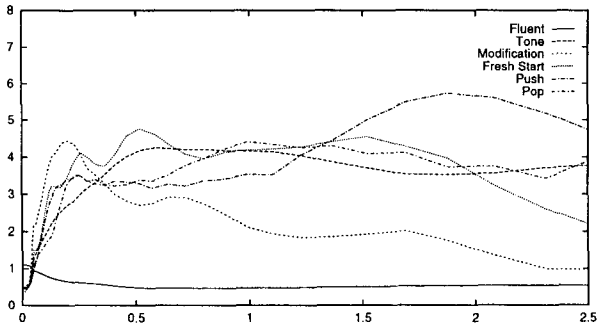


**Figure 7**
Preference for utterance tags given the length of silence.

term variables that this information is most appropriate. We could let the decision tree algorithm use the silence duration as part of the context in estimating the probability distributions. However, our attempts at doing this have not met with success, perhaps because asking questions about the silences fragments the training data and hence makes it difficult to model the influence of the other aspects of the context. Instead, we treat the silence information as being independent from the other context. Below we give the derivation for the intonation variable. For expository ease, we define $Context_i$ to be the prior context for deciding the probabilities for word $W_i$.

$$Context_i = W_{1,i-1}D_{1,i-1}C_{1,i-1}L_{1,i-1}O_{1,i-1}R_{1,i-1}E_{1,i-1}I_{1,i-1}S_{1,i-1}$$

The derivation is as follows.

$$\begin{aligned}
\Pr(I_i|S_i Context_i) &= \frac{\Pr(Context_i S_i|I_i)\Pr(I_i)}{\Pr(S_i Context_i)} \\
&\approx \frac{\Pr(Context_i|I_i)\Pr(S_i|I_i)\Pr(I_i)}{\Pr(S_i)\Pr(Context_i)} \\
&= \Pr(I_i|Context_i)\frac{\Pr(I_i|S_i)}{\Pr(I_i)} \qquad (16)
\end{aligned}$$

The second line involved the assumptions that $Context_i$ and $S_i$ are independent and that $Context_i$ and $S_i$ are independent given $I_i$. The first assumption is obviously too strong. If the previous word is a noun it is more likely that there will be a silence after it than if the previous word was an article. However, the assumptions allow us to model the silence information independently from the other context, which gives us more data to estimate its effect. The result is that we use the factor $\frac{\Pr(I_i|S_i)}{\Pr(I_i)}$ to adjust the probabilities computed by the decision tree algorithm, which does not use the silence durations. We guard against shortcomings by normalizing the adjusted probabilities to ensure that they sum to one.

To compute $\Pr(I_i|S_i)$, we group the silence durations into 30 intervals and then smooth the counts using a Gaussian filter. We do the same adjustment for the editing term and repair variables. For the editing term variable, we only do the adjustment if the intonation tag is null, due to a lack of data in which editing terms co-occur with intonational phrasing. For the repair variable, we only do the adjustment if the intonation tag is null and the editing term tag is not a push or pop. Figure 7 gives the adjustments for the resulting six equivalence classes of utterance tags. The ratio

between the curves gives the preference for one class over another, for a given silence duration. Silence durations were automatically obtained from a word aligner (Entropic Research Laboratory, Inc. 1994).

Silences between speaker turns are not used in computing the preference factor, nor is the preference factor used at such points. The end of the speaker's turn is determined jointly by both the speaker and the hearer. So when building a system that is designed to participate in a conversation, these silence durations will be partially determined by the system's turn-taking strategy. We also do not include the silence durations after word fragments since these silences were hand-computed.

## 7. Example

This section illustrates the workings of the algorithm. As in Section 3.4.1, the algorithm is constrained to the word transcriptions and incrementally considers all possible interpretations (those that do not get pruned), proceeding one word at a time. Since resolving speech repairs is the most complicated part of our model, we focus on this using the following example of overlapping repairs.

**Example 35** (d92a-2.1 utt95)

okay % uh and that will take a total of ↑ um let's see total of s- ↑ of seven hours
*reparandum ip:mod et reparandum ip:mod*

Rather than try to show all of the competing hypotheses, we focus on the correct interpretation, which, for this example, happens to be the winning interpretation. We contrast the probabilities of the correct tags with those of its competitors. For reference, we give a simplified view of the context that is used for each probability. Full results of the algorithm will be given in the next section.

### 7.1 Predicting "um" as the Onset of an Editing Term

Below, we give the probabilities involved in the correct interpretation of the word *um* given the correct interpretation of the words *okay uh and that will take a total of.* We start with the intonation variable. The correct tag of **null** is significantly preferred over the alternative, mainly because intonational boundaries rarely follow prepositions.

$$\Pr(I_{10}=\textbf{null} \mid \text{a total of}) = 0.9997$$
$$\Pr(I_{10}=\% \mid \text{a total of}) = 0.0003$$

For $I_{10}$ = **null**, we give the alternatives for the editing term tag. Since an editing term is not in progress, the only possible values are **Push** and **null**.

$$\Pr(E_{10}=\textbf{Push} \mid \text{a total of}) = 0.242$$
$$\Pr(E_{10}=\textbf{null} \mid \text{a total of}) = 0.758$$

With $E_{10}$ = **Push**, the only allowable repair tag is **null**. Since no repair has been started, the reparandum onset $O_{10}$ must be **null**. Similarly, since no repair is in progress, $L_{10}$, the correspondence licensor, and $C_{10}$, the correspondence type, must both be null.

We next hypothesize the POS tag. Below we list all of the tags that have a probability greater than 1%. Since we are starting an editing term, we see that POS tags

associated with the first word of an editing term have a high probability, such as **UH_FP** for *um*, **AC** for *okay*, **CC_D** for *or*, **UH_D** for *well*, and **VB** for the *let* in *let's see*.

$$\Pr(D_{10}=\textbf{UH\_FP} \mid \text{a total of } \textbf{Push}) = 0.731$$
$$\Pr(D_{10}=\textbf{AC} \mid \text{a total of } \textbf{Push}) = 0.177$$
$$\Pr(D_{10}=\textbf{CC\_D} \mid \text{a total of } \textbf{Push}) = 0.026$$
$$\Pr(D_{10}=\textbf{UH\_D} \mid \text{a total of } \textbf{Push}) = 0.020$$
$$\Pr(D_{10}=\textbf{VB} \mid \text{a total of } \textbf{Push}) = 0.026$$

For $D_{10}$ set to **UH_FP**, the word choices are *um*, *uh*, and *er*.

$$\Pr(W_{10}=\text{um} \mid \text{a total of } \textbf{Push UH\_FP}) = 0.508$$
$$\Pr(W_{10}=\text{uh} \mid \text{a total of } \textbf{Push UH\_FP}) = 0.488$$
$$\Pr(W_{10}=\text{er} \mid \text{a total of } \textbf{Push UH\_FP}) = 0.004$$

Given the correct interpretation of the previous words, the probability of the filled pause *um* along with the correct tags is 0.090.

### 7.2 Predicting "total" as the Alteration Onset

We now give the probabilities involved in the second instance of *total*, which is the alteration onset of the first repair, whose editing term *um let's see*, which ends an intonational phrase, has just finished. Again we start with the intonation variable.

$$\Pr(I_{14}=\textbf{\%} \mid \text{a total of } \textbf{Push} \text{ um let's see}) = 0.902$$
$$\Pr(I_{14}=\textbf{null} \mid \text{a total of } \textbf{Push} \text{ um let's see}) = 0.098$$

For $I_{14}$ = %, the editing term probabilities are given below. Since an editing term is in progress, the only possibilities are that it is continued or that it has ended.

$$\Pr(E_{14}=\textbf{Pop} \mid \text{a total of } \textbf{Push} \text{ um let's see \%}) = 0.830$$
$$\Pr(E_{14}=\textbf{ET} \mid \text{a total of } \textbf{Push} \text{ um let's see \%}) = 0.170$$

For $E_{14}$ = **Pop**, we give the probabilities for the repair variable. Since an editing term has just ended, the null tag for the repair variable is ruled out. Note the modification interpretation receives a score approximately one third of that of a fresh start. However, the repair interpretation catches up after the alteration is processed.

$$\Pr(R_{14}=\textbf{Mod} \mid \text{a total of } \textbf{Push} \text{ um let's see \% } \textbf{Pop}) = 0.228$$
$$\Pr(R_{14}=\textbf{Can} \mid \text{a total of } \textbf{Push} \text{ um let's see \% } \textbf{Pop}) = 0.644$$
$$\Pr(R_{14}=\textbf{Abr} \mid \text{a total of } \textbf{Push} \text{ um let's see \% } \textbf{Pop}) = 0.128$$

For $R_{14}$ = **Mod**, we give the probabilities assigned to the possible reparandum onsets.

For each, we give the proposed reparandum onset, $X$, and the words that precede it.

$$\begin{aligned}
&\Pr(O_{14,X}=\textbf{Onset} \mid W=\text{take a total} \quad X=\text{of} \quad R=\textbf{Mod}) = 0.589\\
&\Pr(O_{14,X}=\textbf{Onset} \mid W=\text{will take a} \quad X=\text{total} \quad R=\textbf{Mod}) = 0.126\\
&\Pr(O_{14,X}=\textbf{Onset} \mid W=\text{that will take} \quad X=\text{a} \quad R=\textbf{Mod}) = 0.145\\
&\Pr(O_{14,X}=\textbf{Onset} \mid W=\text{and that will} \quad X=\text{take} \quad R=\textbf{Mod}) = 0.023\\
&\Pr(O_{14,X}=\textbf{Onset} \mid W=\text{uh and that} \quad X=\text{will} \quad R=\textbf{Mod}) = 0.016\\
&\Pr(O_{14,X}=\textbf{Onset} \mid W=\text{\% uh and} \quad X=\text{that} \quad R=\textbf{Mod}) = 0.047\\
&\Pr(O_{14,X}=\textbf{Onset} \mid W=\text{okay \% uh} \quad X=\text{and} \quad R=\textbf{Mod}) = 0.047\\
&\Pr(O_{14,X}=\textbf{Onset} \mid W=\text{<turn> okay \%} \quad X=\text{uh} \quad R=\textbf{Mod}) = 0.003\\
&\Pr(O_{14,X}=\textbf{Onset} \mid W=\text{<turn>} \quad X=\text{okay} \quad R=\textbf{Mod}) = 0.003
\end{aligned}$$

With *total* as the reparandum onset, there are two possibilities for which word of the reparandum will license the current word—either the word *total* or *of*.

$$\begin{aligned}
&\Pr(L_{10,X}=\textbf{Corr} \mid W=\text{will take a} \quad X=\text{total} \quad R=\textbf{Mod}) = 0.973\\
&\Pr(L_{10,X}=\textbf{Corr} \mid W=\text{will take a} \quad X=\text{of} \quad R=\textbf{Mod}) = 0.027
\end{aligned}$$

With *total* as the correspondence licensor, we need to decide the type of correspondence: whether it is a word match, word replacement, or otherwise.

$$\begin{aligned}
&\Pr(C_{14}=\textbf{m} \mid W=\text{will take a} \quad L=\text{total} \quad \text{R=}\textbf{Mod}) = 0.5882\\
&\Pr(C_{14}=\textbf{r} \mid W=\text{will take a} \quad L=\text{total} \quad \text{R=}\textbf{Mod}) = 0.1790\\
&\Pr(C_{14}=\textbf{x} \mid W=\text{will take a} \quad L=\text{total} \quad \text{R=}\textbf{Mod}) = 0.2328
\end{aligned}$$

For the correct interpretation, the word correspondence is a word match with the word *total* and POS tag **NN**. Hence, the POS tag and identity of the current word are both fixed and hence have a probability of 1. Given the correct interpretation of the previous words, the probability of the word *total* along with the correct tags is 0.0111.

## 8. Results

In this section, we present the results of running our model on the Trains corpus. This section not only shows the feasibility of the model, but also supports the thesis that the tasks of resolving speech repairs, identifying intonational phrases and discourse markers, POS tagging, and speech recognition language modeling must be accomplished in a single model to account for the interactions between these tasks. We start with the models that we presented in Section 3, and vary which variables of Section 4, 5, and 6 that we include. All results in this section were obtained using the sixfold cross-validation procedure described in Section 3.4.1.

### 8.1 POS Tagging, Perplexity, and Discourse Markers

Table 10 shows that POS tagging, word perplexity, and discourse markers benefit from modeling intonational phrases and speech repairs. The second column gives the results of the POS-based language model of Section 3. The third column adds intonational phrase detection, which reduces the POS error rate by 3.8%, improves

**Table 10**
Comparison of POS tagging, discourse marker identification, and perplexity rates.

| | WD | WDI | WDCLORE | WDCLOREI | WDCLOREIS |
|---|---|---|---|---|---|
| POS Errors | 1,711 | 1,646 | 1,688 | 1,652 | 1,563 |
| POS Error Rate | 2.93 | 2.82 | 2.89 | 2.83 | 2.68 |
| DM Errors | 630 | 587 | 645 | 611 | 533 |
| DM Error Rate | 7.61 | 7.09 | 7.79 | 7.38 | 6.43 |
| Word Perplexity | 24.04 | 23.91 | 23.17 | 22.96 | 22.35 |
| Branching Perplexity | 26.35 | 30.61 | 27.69 | 31.59 | 30.26 |

discourse marker identification by 6.8%, and reduces perplexity slightly from 24.04 to 23.91. These improvements are of course at the expense of the branching perplexity, which increases from 26.35 to 30.61. Column four augments the POS-based model with speech repair detection and correction, which improves POS tagging and reduces word perplexity by 3.6%, while only increasing the branching perplexity from 26.35 to 27.69. Although we are adding five variables to the speech recognition problem, most of the extra ambiguity is resolved by the time the word is predicted. Thus, corrections can be sufficiently resolved by the first word of the alteration. Column five combines the models of columns three and four and results in a further improvement in word perplexity. POS tagging and discourse marker identification do not seem to benefit from combining the two processes, but both rates remain better than those obtained from the base model.

Column six adds silence information. Silence information is not directly used to decide the POS tags, the discourse markers, nor what words are involved; rather, it gives evidence as to whether an intonational boundary, speech repair, or editing term occurred. As the following sections show, silence information improves the performance on these tasks, and this translates into better language modeling, resulting in a further decrease in perplexity from 22.96 to 22.35, giving an overall perplexity reduction of 7.0% over the POS-based model. We also see a significant improvement in POS tagging with an error rate reduction of 9.5% over the POS-based model, and a reduction in the discourse marker error rate of 15.4%. As we further improve the modeling of the user's utterance, we should expect to see further improvements in the language model.

### 8.2 Intonational Phrases

Table 11 demonstrates that modeling intonational phrases benefits from modeling silence information, speech repairs, and discourse markers. Column two gives the base results of modeling intonational phrase boundaries. Column three adds silence information, which reduces the error rate for turn-internal boundaries by 9.1%. Column four adds speech repair detection, which further reduces the error rate by 3.5%. Column five adds speech repair correction. Curiously, this actually slightly increases the error rate for intonational boundaries but the rate is still better than not modeling repairs at all (column four). The final result for within-turn boundaries is a recall rate of 71.8%, with a precision of 70.8%. The last column subtracts out the discourse marker modeling by using the POS tagset **P** of Section 3.4.5, which collapses discourse marker usage with sentential usages. Removing the modeling of discourse markers results in a 2.0% degradation in identifying turn-internal boundaries and 7.2% for end-of-turn boundaries.

**Table 11**
Comparison of errors in detecting intonational phrase boundaries.

| | WDI | WDIS | WDREIS | WDCLOREIS | WPCLOREIS |
|---|---|---|---|---|---|
| Within Turn | 3,585 | 3,259 | 3,145 | 3,199 | 3,262 |
| End of Turn | 439 | 439 | 436 | 433 | 464 |
| All Boundaries | 4,024 | 3,698 | 3,581 | 3,632 | 3,726 |

**Table 12**
Comparison of errors in detecting speech repairs.

| | WDRE | WDCLORE | WDCLOREI | WDCLOREIS | WPCLOREIS |
|---|---|---|---|---|---|
| All Repairs | 1,106 | 982 | 909 | 839 | 879 |
| Exact Repairs | 1,496 | 1,240 | 1,185 | 1,119 | 1,169 |
| Abridged | 161 | 187 | 173 | 170 | 183 |
| Modification | 747 | 512 | 489 | 459 | 497 |
| Fresh Starts | 588 | 541 | 523 | 490 | 489 |

### 8.3 Detecting Speech Repairs

We now demonstrate that detecting speech repairs benefits from modeling speech repair correction, intonational phrases, silences, and discourse markers. We use two measures to compare speech repair detection. The first measure, referred to as *All Repairs*, ignores errors that result from improperly identifying the type of repair, and hence scores a repair as correctly detected as long as it was identified as either an abridged repair, a modification repair, or a fresh start. For experiments that include speech repair correction, we further relax this rule. When multiple repairs have contiguous reparanda, we count all repairs involved (of the hand-annotations) as correct as long as the combined reparandum is correctly identified. Hence, for Example 29 given earlier, as long as the overall reparandum was identified as *from engine from*, both of the hand-annotated repairs are counted as correct.

We argued earlier that the proper identification of the type of repair is necessary for successful correction. Hence, the second measure, *Exact Repairs*, counts a repair as being correctly identified only if the type of the repair is also properly determined. Under this measure, a fresh start detected as a modification repair is counted as a false positive and as a missed repair. Just as with *All Repairs*, for models that include speech repair correction, if a misidentified repair is correctly corrected, then it is counted as correct. We also give a breakdown of this measure by repair type.

The results are given in Table 12. The second column gives the base results for detecting speech repairs. The third column adds speech repair correction, which improves the error rate from 46.2% to 41.0%, a reduction of 11.2%. Part of this improvement is attributed to better scoring of overlapping repairs. However, from an analysis of the results, we found that this could account for at most 32 of the 124 fewer errors. Hence, a reduction of at least 8.3% is directly attributed to incorporating speech repair correction. The fourth column adds intonational phrasing, which reduces the error rate for detecting repairs from 41.0% to 37.9%, a reduction of 7.4%. The fifth column adds silence information, which further reduces the error rate to 35.0%, a reduction of 7.7%. Part of this improvement is a result of improved intonational phrase modeling, and

**Table 13**
Comparison of errors in correcting speech repairs.

| | WDCLORE | WDCLOREI | WDCLOREIS | WPCLOREIS |
|---|---|---|---|---|
| All Repairs | 1,506 | 1,411 | 1,363 | 1,435 |
| Abridged | 187 | 175 | 172 | 185 |
| Modification | 616 | 563 | 535 | 607 |
| Fresh Starts | 703 | 673 | 656 | 643 |

part is a result of using pauses to detect speech repairs. This gives a final recall rate of 76.8% with a precision of 86.7%. In the last column, we show the effect of removing the modeling of discourse markers, which increases the error rate of detecting repairs by 4.8%.

### 8.4 Correcting Speech Repairs

Table 13 shows that correcting speech repairs benefits from modeling intonational phrasing, silences, and discourse markers. Column two gives the base results for correcting repairs, which is a recall rate of 61.9% and a precision of 71.4%. Note that abridged and modification repairs are corrected at roughly the same rate but the correction of fresh starts proves particularly problematic. Column three adds intonational phrase modeling. Just as with detecting repairs, we see that this improves correcting each type of repair, with the overall error rate decreasing from 62.9 to 58.9, a reduction of 6.3%. From Table 12, we see that only 73 fewer errors were made in detecting repairs after adding intonational phrase modeling, while 95 fewer errors were made in correcting them. Thus adding intonation phrases leads to better correction of the detected repairs. Column four adds silence information, which further reduces the error rate to 56.9%, a reduction of 3.4%. This gives a final recall rate of 65.9% with a precision of 74.3%. The last column subtracts out discourse marker modeling, which degrades the correction error rate by 5.2%. From Table 12, 40 errors were introduced in detecting repairs by removing discourse marker modeling, while 72 errors were introduced in correcting them. Thus modeling discourse markers leads to better correction of the detected repairs.

### 8.5 Collapsing Repair Distinctions

Our classification scheme distinguishes between fresh starts and modification repairs. Table 14 contrasts the full model (column 3) with one that collapses modification repairs and fresh starts (column 2). To ensure a fair comparison, the reported detection rates do not penalize incorrect identification of the repair type. We find that distinguishing fresh starts and modification repairs results in a 7.0% improvement in detecting repairs and a 6.6% improvement in correcting them. Hence, the two types of repairs differ enough both in how they are signaled and the manner in which they are corrected that it is worthwhile to model them separately. Interestingly, we also see that distinguishing between fresh starts and modification repairs improves intonational phrase identification by 1.9%. This improvement is undoubtedly attributable to the fact that the reparandum onset of fresh starts interacts more strongly with intonational boundaries than does the reparandum onset of modification repairs. As for perplexity and POS tagging, there was virtually no difference, except a slight increase in branching perplexity for the full model.

**Table 14**
Effect of collapsing modification repairs and fresh starts.

| | Collapsed | Distinct |
|---|---|---|
| Errors in Detecting Speech Repairs | 902 | 839 |
| Errors in Correcting Speech Repairs | 1,460 | 1,363 |
| Errors in Identifying Within-Turn Boundaries | 3,260 | 3,199 |

**Table 15**
Speech repair detection and correction results for full model.

| | Detection | | | Correction | | |
|---|---|---|---|---|---|---|
| | Recall | Precision | Error Rate | Recall | Precision | Error Rate |
| All Repairs | 76.79 | 86.66 | 35.01 | 65.85 | 74.32 | 56.88 |
| Abridged | 75.88 | 82.51 | 40.18 | 75.65 | 82.26 | 40.66 |
| Modification | 80.87 | 83.37 | 35.25 | 77.95 | 80.36 | 41.09 |
| Fresh Starts | 48.58 | 69.21 | 73.02 | 36.21 | 51.59 | 97.76 |
| Modification and Fresh Starts | 73.69 | 83.85 | 40.49 | 63.76 | 72.54 | 60.36 |

## 9. Comparison

Comparing the performance of our model to others that have been proposed is problematic. First, there are differences in corpora. The Trains corpus is a collection of dialogues between two people, both of whom realize that they are talking to another person. The ATIS corpus (MADCOW 1992), on the other hand, is a collection of human-computer dialogues. The rate of repairs in this corpus is much lower and almost all speaker turns consists of just one contribution. The Switchboard corpus (Godfrey, Holliman, and McDaniel 1992) is a collection of human-human dialogues, which are much less constrained and about a much wider domain. Even more extreme are corpora of professionally read speech. A second problem is that different systems employ different inputs; for instance, does the input include POS tags, utterance segmentation, or hand-transcriptions of the words that were uttered? We also note that this work is the first proposal that combines the detection and correction of speech repairs, the identification of intonational phrases and discourse markers, and POS tagging, in a framework that is amenable to speech recognition. Hence our comparison is with systems that address only part of the problem.

### 9.1 Speech Repairs

Table 15 gives the results of the full model for detecting and correcting speech repairs. The overall correction recall rate is 65.9% with a precision of 74.3%. In the table, we also report the results for each type of repair using the *Exact Repair* metric. To facilitate comparisons with approaches that do not distinguish between modification repairs and fresh starts, we give the combined results of these two categories.

Bear, Dowding, and Shriberg (1992) investigated the use of pattern matching of the word correspondences, global and local syntactic and semantic ill-formedness, and acoustic cues as evidence for detecting speech repairs. They tested their pattern matcher on a subset of the ATIS corpus from which they removed "all trivial" repairs, repairs that involve only the removal of a word fragment or a filled pause. For their

pattern-matching results, they achieved a detection recall rate of 76% with a precision of 62%, and a correction recall rate of 44% with a precision of 35%. They also tried combining syntactic and semantic knowledge in a "parser-first" approach—first try to parse the input and if that fails, invoke repair strategies based on word patterns in the input. In a test set containing 26 repairs Dowding et al. 1993, they obtained a detection recall rate of 42% with a precision of 85%, and a correction recall rate of 31% with a precision of 62%.

Nakatani and Hirschberg (1994) proposed that speech repairs should be detected in a "speech-first" model using acoustic-prosodic cues, without relying on a word transcription. In order to test their theory, they built a decision tree using a training corpus of 148 turns of speech. They used hand-transcribed prosodic-acoustic features such as silence duration, energy, and pitch, as well as traditional text-first cues such as presence of word fragments, filled pauses, word matches, word replacements, POS tags, and position of the word in the turn, and obtained a detection recall rate of 86% with a precision of 91%. The cues they found relevant were duration of pauses between words, word fragments, and lexical matching within a window of three words. Note that in their corpus, 73% of the repairs were accompanied by a word fragment, as opposed to 32% of the modification repairs and fresh starts in the Trains corpus. Hence, word fragments are a stronger indicator of speech repairs in their corpus than in the Trains corpus. Also note that their training and test sets only included turns with speech repairs; hence their "findings should be seen more as indicative of the relative importance of various predictors of [speech repair] location than as a true test of repair site location" (page 1612).

Stolcke and Shriberg (1996b) incorporated repair resolution into a word-based language model. They limited the types of repair to single and double word repetitions and deletions, deletions from the beginning of the sentence, and filled pauses. In predicting a word, they summed over the probability distributions for each type of repair (including no repair at all). For hypotheses that include a repair, the prediction of the next word was based upon a cleaned up representation of the context, and took into account whether a single or double word repetition was predicted. Surprisingly, they found that this model actually degrades performance, in terms of perplexity and word error rate. They attributed this to their treatment of filled pauses: utterance-medial filled pauses should be cleaned up before predicting the next word, whereas utterance-initial ones should be left intact, a distinction that we make in our model by modeling intonational phrases.

Siu and Ostendorf (1996) extended a language model to account for three roles that words such as filled pauses can play in an utterance: utterance-initial, part of a nonabridged repair, or part of an abridged repair. By using training data with these roles marked and a function-specific variable $n$-gram model (i.e., use a different context for the probability estimates depending on the function of the word), and summing over each possible role, they achieved a perplexity reduction of 82.9 to 81.1.

### 9.2 Utterance Units and Intonational Phrases

We now contrast our intonational phrase results with the results of other researchers in phrases, or other definitions of utterance units. Table 16 gives our performance. Most methods for detecting phrases use end-of-turn as a source of evidence; however, this is jointly determined by both participants. Hence, a dialogue system, designed to participate in the conversation, will not be able to take advantage of this information. For this reason, we focus on turn-internal intonational phrase boundaries.

**Table 16**
Intonational phrase results for full model.

| | Recall | Precision | Error Rate |
|---|---|---|---|
| Within Turn | 71.76 | 70.82 | 57.79 |
| End of Turn | 98.05 | 94.17 | 8.00 |
| All Boundaries | 84.76 | 82.53 | 33.17 |

Wightman and Ostendorf (1994) used preboundary lengthening, pausal durations, and other acoustic cues to automatically label intonational phrases and word accents. They trained a decision tree to estimate the probability of a phrase boundary given the acoustic context. These probabilities were fed into a Markov model whose state is the boundary type of the previous word. For training and testing their algorithm, they used a single-speaker corpus of news stories read by a public radio announcer. With this speaker-dependent model, they achieved a recall rate of 78.1% and a precision of 76.8%.[4] However, it is unclear how well this will adapt to spontaneous speech, where repairs might interfere with the cues that they use, and to speaker independent testing.

Wang and Hirschberg (1992) also looked at detecting intonational phrases. Using automatically labeled features, including POS tag of the current word, category of the constituent being built, distance from last boundary, and word accent, they built decision trees to classify each word as to whether it has an intonational boundary. Note that they do not model interactions with other tasks, such as POS tagging. With this approach, they achieved a recall rate of 79.5% and a precision rate of 82.7% on a subset of the ATIS corpus. Excluding end-of-turn data gives a recall rate of 72.2% and a precision of 76.2%. These results group speech repairs with intonational boundaries and do not distinguish between them. In their corpus, there were 424 disfluencies and 405 turn-internal boundaries. The performance of the decision tree that does not classify disfluencies as intonational boundaries is significantly worse. However, these results were achieved with one-tenth the data of the Trains corpus.

Kompe et al. (1995) combined acoustic cues with a statistical language model to find intonational phrases. They combined normalized syllable duration, length of pauses, pitch contour, and energy using a multilayered perceptron that estimates the probability $\Pr(v_i|c_i)$, where $v_i$ indicates if there is a boundary after the current word and $c_i$ is the acoustic features of the neighboring six syllables. This score is combined with the score from a statistical language model, which determines the probability of the word sequence with the hypothesized phrase boundary inserted using a backoff strategy.

$$\Pr(v_i|c_i)\Pr^{\xi}(\ldots w_{i-1}w_i v_i w_{i+1}w_{i+2}\ldots)$$

Building on this work, Mast et al. (1996) segmented speech into speech acts as the first step in automatically classifying them and achieved a recognition accuracy of 92.5% on turn-internal boundaries using Verbmobil dialogues. This translates into a recall rate of 85.0%, a precision of 53.1%, and an error rate of 90.1%. Their model, which employs rich acoustic modeling, does not account for interactions with speech repairs or discourse markers, nor does it redefine the speech recognition language model.

Meteer and Iyer (1996) investigated whether modeling linguistic segments, segments with a single independent clause, improves language modeling. They computed

4 Derivations of recall and precision rates are given in detail in Heeman (1997).

the probability of the sequence of words with the hypothesized segment boundaries inserted into the sequence. Working on the Switchboard corpus, they found that predicting linguistic boundaries improved perplexity from 130 to 127. Similar to this work, Stolcke and Shriberg (1996a) investigated how the language model can find the boundaries. Their best results were obtained by using POS tags as part of the input, as well as the word identities of certain word classes, in particular, filled pauses, conjunctions, and certain discourse markers. However, this work does not incorporate automatic POS tagging and discourse marker identification.

### 9.3 Discourse Markers

The full model results in 533 errors in discourse marker identification, giving an error rate of 6.43%, a recall of 97.26%, and a precision of 96.32%. Although numerous researchers have noted the importance of discourse markers in determining discourse structure, there has not been a lot of work in actually identifying them.

Hirschberg and Litman (1993) examined how intonational information can distinguish between discourse and sentential interpretation for a set of ambiguous lexical items. They used hand-transcribed intonational features and only examined discourse markers that were one word long, as we have. They found that discourse usages were either an intermediate phrase by themselves (or in a phrase consisting entirely of ambiguous tokens), or they are first in an intermediate phrase (or preceded by other ambiguous tokens) and are either de-accented or have a low word accent. In a monologue of approximately 12,500 words, their model achieved a recall rate of 63.1% with a precision of 88.3%. Many of the errors occurred on coordinate conjuncts, such as *and*, *or*, and *but*, which proved problematic for annotating as well, since "the discourse meanings of conjunction as described in the literature ... seem to be quite similar to the meanings of sentential conjunction" (page 518).

Litman (1996) used machine learning techniques to identify discourse markers. The best set of features for predicting discourse markers were lengths of intonational and intermediate phrase, positions of token in intonational and intermediate phrase, composition of intermediate phrase (token is alone in intermediate phrase or phrase consists entirely of potential discourse markers), and identity of the token. The algorithm achieved a success rate of 85.5%, which translates into a discourse marker error rate of 37.3%, in comparison to the rate of 45.3% for Hirschberg and Litman (1993). Direct comparisons with our error rate of 6.4% are problematic since our corpus is five times as large and we use task-oriented human-human dialogues, which include a lot of turn-initial discourse markers for coordinating mutual belief. In any event, the work of Litman and Hirschberg indicates the usefulness of modeling intermediate phrase boundaries and word accents. Conversely, our approach does not force decisions to be made independently and does not assume intonational annotations as input; rather, we identify discourse markers as part of the task of searching for the best assignment of discourse markers along with POS tags, speech repairs, and intonational phrases.

## 10. Conclusion and Future Work

In this paper, we redefined the speech recognition language model so that it also identifies POS tags, intonational phrases, and discourse markers, and resolves speech repairs. This language model allows the speech recognizer to model the speaker's *utterances*, rather than simply the words involved. This allows it to better account for the words involved and allows it to return a more meaningful analysis of the speaker's turn for later processing. The model incorporates identifying intonational phrases, discourse markers, and POS tags, and detecting and correcting speech repairs; hence,

interactions that exist between these tasks, as well as the task of predicting the next word, can be modeled.

Constraining our model to the hand-transcription, it is able to identify 71.8% of all turn-internal intonational boundaries with a precision of 70.8%, identify 97.3% of all discourse markers with a precision of 96.3%, and detect and correct 65.9% of all speech repairs with a precision of 74.3%. These results are partially attributable to accounting for the interaction between these tasks: modeling intonation phrases improves speech repair detection by 7.4% and correction by 6.3%; modeling speech repairs improves intonational phrase identification by 3.5%; modeling repair correction improves repair detection by 8.3%; modeling repairs and intonational phrases improves discourse marker identification by 15.4%; and removing the modeling of discourse markers degrades intonational phrase identification by 2.0%, speech repair detection by 4.8%, and speech repair correction by 5.2%. Speech repairs and intonational phrases create discontinuities that traditional speech recognition language models and POS taggers have difficulty modeling. Modeling speech repairs and intonational phrases results in a 9.5% improvement in POS tagging and a 7.0% improvement in perplexity. Part of this improvement is from exploiting silences to give evidence of the speech repairs and intonational phrase boundaries.

More work still needs to be done. First, with the exception of pauses, we do not consider acoustic cues. This is a rich source of information for detecting (and distinguishing between) intonational phrases, interruption points of speech repairs, and even discourse markers. It would also help in determining the reparandum onset of fresh starts, which tend to occur at intonational boundaries. Acoustic modeling is also needed to identify word fragments. The second area is extending the model to incorporate higher level syntactic and semantic processing. This would not only allow us to give a much richer output from the model, but it would also allow us to account for interactions between this higher-level knowledge and modeling speakers' utterances, especially in detecting the ill-formedness that often occurs with speech repairs. It would also aid in finding richer correspondences between the reparandum and alteration, such as between the noun phrase and pronoun in the following example.

**Example 36** (d93-14.3 utt27)

$$\underbrace{\text{the engine can take as many}}_{\textit{reparandum}} \uparrow_{ip} \underbrace{\text{um}}_{et} \underbrace{\text{it can take}}_{\textit{alteration}} \text{ up to three loaded boxcars}$$

The third area of future research is to show that our model works on other languages. Although the model encodes the basic structure of speech repairs, intonational phrases, and discourse markers, actual parameters are learned from a training corpus. Preliminary work on a Japanese corpus indicates that the model is not language specific (Heeman and Loken-Kim 1999). The fourth and most important area is to incorporate our work into a speech recognizer. We have already used our POS-based model to rescore word-graphs, which results in a one percent absolute reduction in word error rate in comparison to a word-based model (Heeman 1999). Our full model, which accounts for intonational phrases and speech repairs, should lead to a further reduction, as well as return a richer understanding of the speech.

**Acknowledgments**

We wish to thank Allen Black, John Dowding, K.H. Loken-Kim, Tsuyoshi Morimoto, Massimo Poesio, Eric Ringger, Len Schubert, Elizabeth Shriberg, Michael Tanenhaus, and David Traum. We also wish

to thank Eva Bero, Bin Li, Greg Mitchell, Andrew Simchik, and Mia Stern for their help in transcribing and giving us useful comments on the annotation schemes. Funding gratefully received from NSERC Canada, NSF under grant IRI-9623665, DARPA—Rome Laboratory under research contract F30602-95-1-0025, ONR/DARPA under grant N00014-92-J-1512, ONR under grant N0014-95-1-1088, ATR Interpreting Telecommunications Laboratory and CNET, France Télécom.

**References**

Allen, James F., Lenhart K. Schubert, George Ferguson, Peter Heeman, Chung Hee Hwang, Tsuneaki Kato, Marc Light, Nathaniel Martin, Bradford Miller, Massimo Poesio, and David Traum. 1995. The Trains project: A case study in building a conversational planning agent. *Journal of Experimental and Theoretical AI*, 7:7–48.

Bahl, Lalit R., J. K. Baker, Frederick Jelinek, and Robert L. Mercer. 1977. Perplexity—A measure of the difficulty of speech recognition tasks. In *Proceedings of the 94th Meeting of the Acoustical Society of America.*

Bahl, Lalit R., Peter F. Brown, Peter V. deSouza, and Robert L. Mercer. 1989. A tree-based statistical language model for natural language speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(7):1001–1008.

Bard, Ellen G. and Robin J. Lickley. 1997. On not remembering disfluencies. In *Proceedings of the 5th European Conference on Speech Communication and Technology*, pages 2855–2858.

Beach, Cheryl M. 1991. The interpretation of prosodic patterns at points of syntactic structure ambiguity: Evidence for cue trading relations. *Journal of Memory and Language*, 30(6):644–663.

Bear, John, John Dowding, and Elizabeth E. Shriberg. 1992. Integrating multiple knowledge sources for detection and correction of repairs in human-computer dialogue. In *Proceedings of the 30th Annual Meeting*, pages 56–63. Association for Computational Linguistics.

Bear, John, John Dowding, Elizabeth E. Shriberg, and Patti Price. 1993. A system for labeling self-repairs in speech. Technical Note 522, SRI International.

Bear, John and Patti Price. 1990. Prosody, syntax, and parsing. In *Proceedings of the 28th Annual Meeting*, pages 17–22. Association for Computational Linguistics.

Black, Ezra, Fred Jelinek, John Lafferty, David Magerman, Robert Mercer, and Salim Roukos. 1992. Towards history-based grammars: Using richer models for probabilistic parsing. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 134–139. Morgan Kaufman.

Blackmer, Elizabeth R. and Janet L. Mitton. 1991. Theories of monitoring and the timing of repairs in spontaneous speech. *Cognition*, 39:173–194.

Breiman, Leo, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. 1984. *Classification and Regression Trees.* Wadsworth & Brooks.

Brown, Gillian and George Yule. 1983. *Discourse Analysis.* Cambridge University Press.

Brown, Peter F., Vincent J. Della Pietra, Peter V. deSouza, Jenifer C. Lai, and Robert L. Mercer. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

Charniak, Eugene, C. Hendrickson, N. Jacobson, and M. Perkowitz. 1993. Equations for part-of-speech tagging. In *Proceedings of the National Conference on Artificial Intelligence*, pages 784–789.

Chow, Yen-Lu and Richard Schwartz. 1989. The n-best algorithm: An efficient procedure for finding top n sentence hypotheses. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 199–202.

Church, Kenneth. 1988. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the 2nd Conference on Applied Natural Language Processing*, pages 136–143.

Dowding, John, Jean M. Gawron, Doug Appelt, John Bear, Lynn Cherny, Robert Moore, and Douglas Moran. 1993. Gemini: A natural language system for spoken-language understanding. In *Proceedings of the 31st Annual Meeting*, pages 54–61. Association for Computational Linguistics.

Entropic Research Laboratory, Inc. 1994. *Aligner Reference Manual.* Version 1.3.

Godfrey, John J., Edward C. Holliman, and Jane McDaniel. 1992. SWITCHBOARD: Telephone speech corpus for research and development. In *Proceedings of the International Conference on Audio, Speech and Signal Processing*, pages 517–520.

Heeman, Peter A. 1997. *Speech Repairs, Intonational Boundaries and Discourse Markers: Modeling Speakers' Utterances in Spoken Dialogue.* Doctoral dissertation, Department of Computer Science,

University of Rochester.
Heeman, Peter A. 1999. POS tags and decision trees for language modeling. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 129–137.
Heeman, Peter A. and James F. Allen. 1994. Detecting and correcting speech repairs. In *Proceedings of the 32nd Annual Meeting*, pages 295–302. Association for Computational Linguistics.
Heeman, Peter A. and James F. Allen. 1995. *The Trains Spoken Dialog Corpus*. CD-ROM, Linguistics Data Consortium.
Heeman, Peter A. and K. H. Loken-Kim. 1999. Detecting and correcting speech repairs in Japanese. In *Proceedings of the ICPhS Satellite Meeting on Disfluency in Spontaneous Speech*, pages 43–46.
Heeman, Peter A. , K. H. Loken-Kim, and James F. Allen. 1996. Combining the detection and correction of speech repairs. In *Proceedings of the 4th International Conference on Spoken Language Processing*, pages 358–361.
Hindle, Donald. 1983. Deterministic parsing of syntactic non-fluencies. In *Proceedings of the 21st Annual Meeting*, pages 123–128. Association for Computational Linguistics.
Hirschberg, Julia and Diane Litman. 1993. Empirical studies on the disambiguation of cue phrases. *Computational Linguistics*, 19(3):501–530.
Jelinek, Frederick. 1985. Self-organized language modeling for speech recognition. Technical report, IBM T.J. Watson Research Center.
Jelinek, Frederick. and Robert L. Mercer. 1980. Interpolated estimation of Markov source parameters from sparse data. In *Proceedings of the Workshop on Pattern Recognition in Practice*, pages 381–397.
Katz, Slava M. 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35(3):400–401.
Kikui, Gen-ichiro and Tsuyoshi Morimoto. 1994. Similarity-based identification of repairs in Japanese spoken language. In *Proceedings of the 3rd International Conference on Spoken Language Processing*, pages 915–918.
Kompe, Ralf, Andreas Kießling, Heinrich Niemann, Elmar Nöth, E. Günter Schukat-Talamazzini, A. Zottmann, and Anton Batliner. 1995. Prosodic scoring of word hypotheses graphs. In *Proceedings of the 4th European Conference on Speech Communication and Technology*, pages 1333–1336.
Levelt, Willem J. 1983. Monitoring and self-repair in speech. *Cognition*, 14:41–104.
Lickley, Robin J. and Ellen G. Bard. 1992. Processing disfluent speech: Recognizing disfluency before lexical access. In *Proceedings of the 2nd International Conference on Spoken Language Processing*, pages 935–938.
Lickley, Robin J., Richard C. Shillcock, and Ellen G. Bard. 1991. Processing disfluent speech: How and when are disfluencies found? In *Proceedings of the 2nd European Conference on Speech Communication and Technology*, pages 1499–1502.
Litman, Diane J. 1996. Cue phrase classification using machine learning. *Journal of Artificial Intelligence Research*, 5:53–94.
MADCOW. 1992. Multi-site data collection for a spoken language corpus. In *Proceedings of the DARPA Workshop on Speech and Natural Language Processing*, pages 7–14.
Magerman, David M. 1994. *Natural Language Parsing as Statistical Pattern Recognition*. Doctoral dissertation, Department of Computor Science, Stanford University.
Marcus, Mitchell P., Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.
Martin, J. G. and W. Strange. 1968. The perception of hesitation in spontaneous speech. *Perception and Psychophysics*, 53:1–15.
Mast, Marion, Ralf Kompe, Stefan Harbeck, Andreas Kießling, Heinrich Niemann, Elmar Nöth, E. Günther Schukat-Talamazzini, and Volker Warnke. 1996. Dialogue act classification with the help of prosody. In *Proceedings of the 4th International Conference on Spoken Language Processing*, pages 1728–1731.
Meteer, Marie and Rukmini Iyer. 1996. Modeling conversational speech for speech recognition. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 33–47.
Nakatani, Christine H. and Julia Hirschberg. 1994. A corpus-based study of repair cues in spontaneous speech. *Journal of the Acoustical Society of America*, 95(3):1603–1616.
Nooteboom, S. G. 1980. Speaking and unspeaking: Detection and correction of phonological and lexical errors. In Victoria A. Fromkin, editor, *Errors in Linguistic Performance: Slips of the Tongue,*

*Ear, Pen, and Hand*. Academic Press, pages 86–97.

O'Shaughnessy, Douglas. 1994. Correcting complex false starts in spontaneous speech. In *Proceedings of the International Conference on Audio, Speech and Signal Processing*, pages 349–352.

Ostendorf, Mari, Colin Wightman, and Nanette Veilleux. 1993. Parse scoring with prosodic information: An analysis/synthesis approach. *Computer Speech and Language*, 7(2):193–210.

Schiffrin, Deborah. 1987. *Discourse Markers*. Cambridge University Press.

Shriberg, Elizabeth E. 1994. *Preliminaries to a Theory of Speech Disfluencies*. Doctoral dissertation, University of California at Berkeley.

Shriberg, Elizabeth E., Rebecca Bates, and Andreas Stolcke. 1997. A prosody-only decision-tree model for disfluency detection. In *Proceedings of the 5th European Conference on Speech Communication and Technology*, pages 2383–2386.

Shriberg, Elizabeth E. and Robin J. Lickley. 1993. Intonation of clause-internal filled pauses. *Phonetica*, 50(3):172–179.

Silverman, Ken, Mary Beckman, John Pitrelli, Mari Ostendorf, Colin Wightman, Patti Price, Janet Pierrehumbert, and Julia Hirschberg. 1992. ToBI: A standard for labelling English prosody. In *Proceedings of the 2nd International Conference on Spoken Language Processing*, pages 867–870.

Siu, Man-hung and Mari Ostendorf. 1996. Modeling disfluencies in conversational speech. In *Proceedings of the 4th International Conference on Spoken Language Processing*, pages 382–391.

Stolcke, Andreas and Elizabeth E. Shriberg. 1996a. Automatic linguistic segmentation of conversational speech. In *Proceedings of the 4th International Conference on Spoken Language Processing*, pages 1001–1004.

Stolcke, Andreas and Elizabeth E. Shriberg. 1996b. Statistical language modeling for speech disfluencies. In *Proceedings of the International Conference on Audio, Speech and Signal Processing*, pages 405–408.

Traum, David R. and Peter A. Heeman. 1997. Utterance units in spoken dialogue. In Elisabeth Maier, Marion Mast, and Susann LuperFoy, editors, *Dialogue Processing in Spoken Language Systems*, Lecture Notes in Artificial Intelligence. Springer-Verlag, Heidelberg, pages 125–140.

Wang, Michelle Q. and Julia Hirschberg. 1992. Automatic classification of intonational phrase boundaries. *Computer Speech and Language*, 6:175–196.

Wightman, Colin and Mari Ostendorf. 1994. Automatic labeling of prosodic patterns. *IEEE Transactions on Speech and Audio Processing*, 2(4):469–481.

Zeppenfeld, Torsten, Michael Finke, Klaus Ries, Martin Westphal, and Alex Waibel. 1997. Recognition of conversational telephone speech using the Janus speech engine. In *Proceedings of the International Conference on Audio, Speech and Signal Processing*, pages 1815–1818.