RECONSTRUCTING SPONTANEOUS SPEECH

by

Erin Colleen Fitzgerald

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

January, 2009

© Erin Colleen Fitzgerald 2009

All rights reserved
Abstract

The output of a speech recognition system is often not what is required for subsequent processing, in part because speakers themselves make mistakes (e.g. stuttering, self-correcting, or using filler words). A system would accomplish speech reconstruction of its spontaneous speech input if its output were to represent, in flawless, fluent, and content-preserved English, the message that the speaker intended to convey. These cleaner speech transcripts would allow for more accurate language processing as needed for natural language tasks such as machine translation and conversation summarization, which often assume a grammatical sentence as input.

Before attempting to reconstruct speech automatically, we seek to comprehensively understand the problem itself. We quantify the range, complexity, and frequencies of speaker errors common in spontaneous speech given a set of manual reconstruction annotations. This empirical analysis indicates the most frequent and problematic errors and thus suggests areas of focus for future reconstruction research.

The surface transformations recorded in our reconstruction annotations reflect underlying influences in the psycholinguistics and speech production models of spontaneous
speakers. We review standard theories and seek empirical evidence of both model assumptions and conclusions given the manual reconstructions and corresponding shallow semantic labeling annotation we collect. This investigation of naturally-occurring spontaneous speaker errors with manual semantico-syntactic analysis yields additional insight into the impact of spoken language on semantic structure and how these features can be used in future reconstruction efforts.

Finally, given our accumulated knowledge about the types, frequencies, and drivers of speaker-generated errors in spontaneous speech, we build a set of systems to automatically identify and correct a subset of the most frequent errors. Using a conditional random field classification model and lexical, syntactic, and shallow semantic features to train both word-level and utterance-level error classifiers, we show improvement in the correction of these errors over a state-of-the-art system.

First Reader: Frederick Jelinek

Second Reader: Sanjeev Khudanpur

Thesis Committee: Frederick Jelinek, Sanjeev Khudanpur, Andreas Andreou, Robert Frank, and Keith Hall
Acknowledgements

This work would not have been possible without the intellectual and personal support of many.

I would first and foremost like to thank my advisor Dr. Frederick Jelinek for his mentorship and support, for always being ready and willing to consider a new idea or approach, and for giving me the independence to pursue my research in the direction I believed was most promising. I would also like to extend my appreciation to the other members of my thesis committee. Dr. Sanjeev Khudanpur has always provided reliable feedback and a plethora of useful suggestions, regardless of the problem at hand. The constant enthusiasm and vision of how reconstruction could influence future research which Dr. Andreas Andreou has expressed helped keep my perspective of the problem fresh in the final months of research and writing. Finally, many thanks to former Johns Hopkins faculty Robert Frank and Keith Hall for continuing their collaborations with me in the past year. Dr. Frank’s perspective on the linguistic side of the reconstruction problem and Dr. Hall’s regular counsel during the development of both the definition of the reconstruction problem and the implementation of automatic reconstruction approaches were invaluable.
Beyond my stellar thesis committee, I was the beneficiary of the help of many additional collaborators. I would like to acknowledge the help of Jan Hajič, Petr Podveský and Petr Pajas of Charles University in building the annotation tool adapted for this work. Thanks also to Olga Babko-Malaya of the Linguistic Data Consortium for her help advising in the initial annotation process and for readily answering my many questions about predicate-argument annotation. Yi Zhang, Valia Kordoni, and Ulrich Schaefer of Universität des Saarlandes and DFKI taught me about Head-driven Phrase Structure Grammar (HPSG) and the hidden potential of deep syntax parsing. Regina Barzilay of MIT provided data and useful discussion regarding summarization and redundancy analysis. At Brown University, I extend special thanks to Matt Lease for his help in running the Johnson and Charniak disfluency identification system locally, and for other feedback. Mark Johnson and Eugene Charniak have given me many useful ideas over the past few years.

My time at Johns Hopkins would not have been the same without my friends and colleagues, past and present, in the Center for Language and Speech Processing. Carolina Parada was my right-hand woman for my last few years of grad school, helping me with PIRE work and escapes to play racquetball, and I thank her for it. My classmates Arnab, Lambert, Yi, and Chris have been tremendously supportive in and outside the lab. Paola, Ahmad, and Peng were great mentors, and Vidya, Noah, David, Roy, Markus, Nikesh, Ariya, and so many others helped to make the experience complete. Of course, none of my work or experiences would have been possible without the amazing behind-the-scenes work by CLSP’s administrators, especially Laura Graham, Sue Porterfield, Monique Folk,
and Desiree Cleves, as well as the computational support provided by system administrators like Eiwe Lingefors and Justin Martin.

Outside of CLSP, I led a double life as a “BME wanna-be”, and I was happy for the opportunity. I was fortunate to count Wilsaan, Jenny, Rachel, Jeremy, Bobby, Elias, and Tricia as some of my best friends these last few years. I thank Sami Alom Ruiz for being a true friend and supportive partner through the last six and a half years. He helped me to take myself less seriously – no easy task – and to find a new confidence in myself.

The constant support of my parents and brother and sister helped me through graduate school and to believe in myself; without their love and faith in my potential, I would not be where I am today.
To my parents.
## Contents

List of Tables .......................... xi

List of Figures .......................... xiii

1 Introduction to speech reconstruction ........................ 1
   1.1 Motivation ................................ 3
   1.2 Defining speaker errors .......................... 5
      1.2.1 Simple disfluency classes in spontaneous speech .... 8
      1.2.2 Why simple disfluencies are not enough .......... 9
      1.2.3 Construction error types .......................... 11
   1.3 Other considerations ................................ 20
      1.3.1 Problem scope .......................... 20
      1.3.2 Speaker errors across languages .................. 22
      1.3.3 Evaluating progress .......................... 23
      1.3.4 Reconciling terminology ......................... 23
   1.4 High level goals of this work .......................... 24

2 Linguistic resources for spontaneous speech reconstruction ..... 27
   2.1 Existing disfluency resources ......................... 28
   2.2 Motivation for building a reconstruction resource .... 30
   2.3 Data analysis and selection .......................... 31
      2.3.1 SU boundary resegmentation ..................... 34
   2.4 Extracting a densely errorful corpus ................... 36
   2.5 Manually reconstructing spontaneous speech ............... 39
      2.5.1 Developing the annotation tool ................... 39
      2.5.2 Annotation quality control ....................... 41
   2.6 Levels of annotation and a vocabulary of allowable changes 42
      2.6.1 Word and alignment level ....................... 43
      2.6.2 Reconstruction efficacy ......................... 44
      2.6.3 Semantic role labeling .......................... 46
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.7</td>
<td>Final refinement of the Spontaneous Speech Reconstruction corpus</td>
<td>47</td>
</tr>
<tr>
<td>2.8</td>
<td>Summary and conclusions</td>
<td>49</td>
</tr>
<tr>
<td>3</td>
<td>Empirical analysis of speaker errors</td>
<td>51</td>
</tr>
<tr>
<td>3.1</td>
<td>Evaluating reconstruction efforts</td>
<td>52</td>
</tr>
<tr>
<td>3.2</td>
<td>Inter-annotator agreement statistics</td>
<td>57</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Annotator-level agreement statistics</td>
<td>60</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Sentence boundary changes</td>
<td>61</td>
</tr>
<tr>
<td>3.3</td>
<td>Analysis of reconstructive transformations</td>
<td>62</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Insertions: Identification and correction</td>
<td>63</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Substitutions: Identification and correction</td>
<td>65</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Word reorderings: Identification and correction</td>
<td>66</td>
</tr>
<tr>
<td>3.3.4</td>
<td>Filler and reparandum agreement with SimpleMDE annotation</td>
<td>69</td>
</tr>
<tr>
<td>3.3.5</td>
<td>Consistency in error boundaries</td>
<td>69</td>
</tr>
<tr>
<td>3.4</td>
<td>Summary and conclusions</td>
<td>70</td>
</tr>
<tr>
<td>4</td>
<td>Underlying structure of spontaneous speech</td>
<td>74</td>
</tr>
<tr>
<td>4.1</td>
<td>Background</td>
<td>75</td>
</tr>
<tr>
<td>4.1.1</td>
<td>Psycholinguistic studies of speaker errors</td>
<td>78</td>
</tr>
<tr>
<td>4.1.2</td>
<td>Semantic role labeling</td>
<td>80</td>
</tr>
<tr>
<td>4.1.3</td>
<td>Potential benefit of semantic analysis to speech reconstruction</td>
<td>82</td>
</tr>
<tr>
<td>4.2</td>
<td>Data</td>
<td>84</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Alignments and alteration labels in the SSR corpus</td>
<td>85</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Semantic role labeling in the SSR corpus</td>
<td>86</td>
</tr>
<tr>
<td>4.3</td>
<td>Interannotator agreement statistics for SSR semantic role labeling</td>
<td>87</td>
</tr>
<tr>
<td>4.4</td>
<td>Syntactic variation between verbatim original and reconstructed strings</td>
<td>88</td>
</tr>
<tr>
<td>4.5</td>
<td>Analysis of semantics for spoken language</td>
<td>94</td>
</tr>
<tr>
<td>4.5.1</td>
<td>Labeled verbs and their arguments</td>
<td>97</td>
</tr>
<tr>
<td>4.5.2</td>
<td>Structural semantic statistics in cleaned speech</td>
<td>98</td>
</tr>
<tr>
<td>4.5.3</td>
<td>Structural semantic differences between verbatim speech and reconstructed speech</td>
<td>101</td>
</tr>
<tr>
<td>4.5.4</td>
<td>Testing generalizations required for automatic semantic role labeling for speech</td>
<td>106</td>
</tr>
<tr>
<td>4.6</td>
<td>Summary</td>
<td>107</td>
</tr>
<tr>
<td>5</td>
<td>Reconstruction and related text transformations</td>
<td>115</td>
</tr>
<tr>
<td>5.1</td>
<td>Previous efforts in simple edit detection</td>
<td>116</td>
</tr>
<tr>
<td>5.1.1</td>
<td>Edit detection via noisy channel TAG models</td>
<td>119</td>
</tr>
<tr>
<td>5.2</td>
<td>Viewing reconstruction through other natural language processing techniques</td>
<td>122</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Reconstruction by paraphrasing</td>
<td>123</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Reconstruction by summarization</td>
<td>126</td>
</tr>
<tr>
<td>5.2.3</td>
<td>Reconstruction by machine translation techniques</td>
<td>129</td>
</tr>
</tbody>
</table>
# List of Tables

2.1 Choosing data for reconstruction: Overview of data characteristics ........................................ 32
2.2 Precision, recall, and $F_\beta$ for the identification of poorly constructed sentences in the 500-sentence sample $S$ ......................................................... 37
2.3 Dimensions of reconstruction annotations for the SSR corpus ................................................ 42
3.1 Quantifying the amount of changes made during reconstruction .............................................. 57
3.2 Pairwise inter-annotator agreement statistics for SSR annotation .............................................. 58
3.3 Annotator-level minTER and alteration frequencies compared to other annotators reconstructed the same utterances ....................................................... 71
3.4 Reconstruction efficacy distribution .......................................................................................... 72
3.5 Actions taken by annotators during manual reconstruction ...................................................... 73
4.1 Predicate-modifying (and non-predicate-specific) argument labels in the Proposition Bank and Spontaneous Speech Reconstruction corpora ............................................... 81
4.2 Pairwise inter-annotator agreement for Fisher: semantic labels ................................................ 87
4.3 Internal syntactic structure removed and gained during reconstruction ..................................... 109
4.4 Most frequent phrasal categories for common arguments in the SSR ...................................... 110
4.5 Most frequent argument categories for common syntactic phrases in the SSR .......................... 111
4.6 Most common argument-to-predicate paths: automatic parses for reconstructed strings ............ 112
4.7 Most common argument-to-predicate paths: semantic labels projected onto manual parses for original strings .................................................. 113
4.8 Most common argument-to-predicate paths: automatic parses for verbatim original strings ................................................................................................................. 114
5.1 RT04 edit word detection EWER results .................................................................................. 118
5.2 JC04 disfluency detection performance on SSR test data ......................................................... 121
5.3 JC04 disfluency detection: rough copies vs. non-copies .......................................................... 121
5.4 BLEU score and Exact SU Match accuracy for Moses SMT-based reconstruction output, compared to original and inter-reference baselines .............................................. 131
5.5 ASR errors: content, function, and OOV analysis of ASR lattices . . . . . . . . 136
6.1 Overview of experimental setups for word-level error predictions . . . . . . 150
6.2 Word-level error prediction F1-score results: Data variation . . . . . . . . 152
6.3 Word-level error prediction F1-score results: Feature variation . . . . . . 153
6.4 Word-level error predictions: Exact Sentence Match Results . . . . . . . . 155
7.1 Comparison of individual poor construction identification approaches . . . . 166
7.2 Error predictions, post-SU identification: F1-score results . . . . . . . . . . 169
7.3 Error predictions, post-SU identification: Exact Sentence Match Results . . 171
8.1 Conversations included in resegmented SSR_CHECK and SSR_TEST . . . . . 188
8.2 Conversations included in resegmented SSR_TRAIN subcorpus . . . . . . . . 189
8.3 More conversations included in resegmented SSR_TRAIN subcorpus . . . . . 190
List of Figures

1.1 A parsed grammatical sentence which can be improved by reconstruction . 6
1.2 Parsed reconstruction of the sentence shown in Figure 1.1. . . . . . . . . . 7
1.3 Typical edit region structure . . . . . . . . . . . . . . . . . . . . . . . . . 8
2.1 A manual parse of an errorful sentence from the Fisher corpus . . . . . . . 29
2.2 Example of word-level simple disfluency labels . . . . . . . . . . . . . . . 37
2.3 Reconstruction and predicate-argument labeling by annotation tool . . . . 39
2.4 Predicate-argument labeling with the annotation tool. . . . . . . . . . . . . 46
4.1 Semantic role labeling for the sentence “some kids like video games”. . . . 80
4.2 Manual and automatic parses for a verbatim string and its manual recon-
struction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 89
4.3 Semantic role labeling for the sentence “some kids like video games” and
its verbatim source string “they like video games and stuff some kids do”. 94
4.4 Semantic role labeled automatic parses for a verbatim string and its manual
reconstruction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 95
4.5 Errorful automatic parse demonstrating predicate (verb) labeled as com-
mon noun NN . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 100
5.1 Example of word-level error class labels . . . . . . . . . . . . . . . . . . . 116
5.2 Refined word-level error class labels . . . . . . . . . . . . . . . . . . . . 121
5.3 Sentence compression paraphrasing example . . . . . . . . . . . . . . . . . 124
5.4 Extractive summarization: Comparing verbatim and official transcripts
from the European Parliamentary Proceedings . . . . . . . . . . . . . . . . 127
5.5 Reconstruction by machine translation . . . . . . . . . . . . . . . . . . . . 133
6.1 Refined word-level error class labels . . . . . . . . . . . . . . . . . . . . 141
6.2 Illustration of a conditional random field . . . . . . . . . . . . . . . . . . . 141
7.1 Automatically generated parse for an errorful SU, with accompanying rule
expansions and local c-commands. . . . . . . . . . . . . . . . . . . . . . . . 163
7.2 List of backchannel acknowledgements and fillers . . . . . . . . . . . . . . . . 166
Chapter 1

Introduction to speech reconstruction

“uh what used to be i know i r- remember were fairly simple uh tasks uh mathematical
is what i’m saying uh i- it’s like i sort of have forgotten uh how to do those things.
so for me the things that i used to know about algebra that’s all i can really think of at
the moment as an example it really is a strain to get back and do.”

“i have forgotten how to do what used to be fairly simple mathematical tasks. it really
is a strain for me to get back and do the things that i used to know about algebra.”

– Verbatim, and reconstructed, text taken from the Fisher Conversational Telephone Speech
corpus and Spontaneous Speech Reconstruction corpus (conversation fsh_60571_B)

In recent years, performance on natural language processing (NLP) tasks such as ma-
chine translation, information extraction, and summarization has been steadily improving,
but rarely applied to the most natural form of language input: spontaneous speech.

The output of a speech recognition system is often not what is required for subse-
quent processing, in part because speakers themselves make mistakes (e.g. stuttering, self-
correcting, or using filler words). Even leaving aside errors introduced by inaccurate auto-
matic speech recognition (ASR) used to transcribe spoken output, the limitations of dealing
with spontaneous speech output include lack of well-formedness, unpredictable structure,
CHAPTER 1. INTRODUCTION TO SPEECH RECONSTRUCTION

and the fact that text alone often is missing elements from the discourse crucial for semantic interpretation, such as prosody, gestures, visual aids, etc. If some of these problems could be resolved, translating and extracting information from spontaneous input would be more feasible.

ASR for conversational speech currently aims at matching a verbatim transcription, including elements such as filler words, speech corrections, and verbal redundancy which are non-essential and in fact problematic to the task of extracting information from the speaker’s intended content. For most tasks we would want to record the intended meaning of the spoken utterance, not report faithfully word-for-word each speech sound produced. In fact, as ASR improves, the word error rate (WER) approaches inter-annotator disagreement rates: for the NIST RT04 conversational telephone speech (CTS) evaluation, the top performing ASR system achieved 12.4% WER, while the word disagreement rate of the annotations ranged from 4-7% (Fiscus and Schwartz, 2004). This raises the question: when we evaluate speech transcription, what should we really be judging? As a way out of this conumdrum, spontaneous speech reconstruction can serve as a “smoothing” of extraneous components of verbatim ASR output.

A system would accomplish reconstruction of its spontaneous speech input if its output were to represent, in flawless, fluent, and content-preserved text, the message that the speaker intended to convey. Given disfluency, syntactic, and semantic analyses, we hope to learn common speech error patterns and find a way to automatically extract verb-argument structure and basic semantic analysis for each sentence. Using this high-level representa-
CHAPTER 1. INTRODUCTION TO SPEECH RECONSTRUCTION

In this work, we aim to identify missing, redundant, and inappropriate arguments and make necessary string transformations to produce a fluent, grammatical and meaning-preserved reconstruction.

The goal of studying manually reconstructed transcripts is to develop an algorithm to automatically reduce an utterance to its essential meaning and then generate clean text. This would include two primary steps:

1. identification and classification of all speaker errors, and
2. reconstruction of the sentence based on the interpretation of the intended utterance.

Reaching this goal requires a thorough analysis and understanding of the form of and drivers behind these errors. In this work, we seek to understand these problems and to implement a reconstruction system which, while not yet targeting the full range of existing problems, exceeds the accuracy of previously built related systems.

1.1 Motivation

The key motivation for this work is the hope that a cleaner, reconstructed speech transcript will allow for simpler and more accurate human and natural language processing, as needed for applications like machine translation, question answering, text summarization, and paraphrasing which often assume a grammatical sentence as input. This benefit has been directly demonstrated for statistical machine translation (SMT). Rao et al. (2007) gave evidence that simple disfluency removal from transcripts can improve BLEU (a standard
CHAPTER 1. INTRODUCTION TO SPEECH RECONSTRUCTION

SMT evaluation metric) up to 8% for sentences with disfluencies. The presence of disfluencies were found to hurt SMT in two ways: making utterances longer without adding semantic content (and sometimes adding false content) and exacerbating the data mismatch between the spontaneous input and the clean text training data.

Humans are experts at recognizing and almost instantaneously interpreting and reconstructing speaker-produced errors (Lickley, 1994). Without the assistance of auditory cues, the task becomes more difficult though still possible. Cleaning verbatim transcripts has been shown to improve readability of transcripts (as evaluated by comprehension measures in studies like Gibson et al. (2004); Jones et al. (2003, 2005)), which is a secondary goal of this work.

A system which could automatically identify speaker construction errors has been shown to be useful for the diagnosis of cognitive impairment or other speech impediments (Roark et al., 2007; Hollingshead and Heeman, 2004) and evaluation of non-native language learning. Reconstruction of such errors could help to correct common mistakes made by non-native speakers (Lee and Seneff, 2006).

Some applications have a direct need for reconstruction into a very specific format. For example, transforming speech output into medical and legal transcriptions is an area of both commercial investment and research. In the medical transcription editing work described in Bisani et al. (2008), the authors observe both speaker self-correction and dropped function words and pronominal arguments, common in spontaneous speech, indicating additional applications for speech reconstruction.
1.2 Defining speaker errors

While reconstruction aims to transform errorful speaker constructions into well-formed text, doing so requires first defining what it means to be “errorful”. We say that an utterance (also called a sentence-like unit, or SU) is poorly constructed if

- it contains speaker self-repairs and disfluencies,
- it is ungrammatical (a disagreement in tense, number, or gender),
- it is incomplete (excluding completed/stand-alone noun phrases and non-content backchannel acknowledgements), or
- it is inaccurately segmented into sentence-like units.

Moreover, as our overriding goal is to transform speech transcripts into a form conducive to downstream natural language processing applications, we also consider transforms which reduce superfluous structure while minimally impacting the meaning of the sentence to be part of the reconstruction task. One simple example of this can be found in Figures 1.1 and 1.2, which show the change in automatically derived parse structure after reconstructing “she’s in the biggest dust bowl in the west bakersfield” to “she’s in bakersfield the biggest dust bowl in the west”. Without punctuation, the utterance can be misanalyzed, as seen in the automatically generated parse tree in Figure 1.1. The tree here shows “in the west bakersfield” as a single constituent that is a direct complement of the present-tense verb (VBZ) “s”. In the reconstruction, the phrase “the biggest dust bowl in the west” is correctly interpreted as a complement of the phrase “bakersfield”.

5
Figure 1.1: A parsed grammatical sentence which can be improved by reconstruction (see Figure 1.2). Without punctuation, the utterance can be misanalyzed, as seen in this automatically generated parse tree. The tree here shows “in the west bakersfield” as a single constituent that is a direct complement of the present-tense verb (VBZ) “s”.

“she's in the biggest dust bowl in the west bakersfield”
CHAPTER 1. INTRODUCTION TO SPEECH RECONSTRUCTION

Figure 1.2: Parsed reconstruction of the sentence shown in Figure 1.1. Notice that in the reconstruction, the phrase “the biggest dust bowl in the west” is correctly interpreted as a complement of the phrase “bakersfield”.

“she’s in bakersfield the biggest dust bowl in the west”
CHAPTER 1. INTRODUCTION TO SPEECH RECONSTRUCTION

1.2.1 Simple disfluency classes in spontaneous speech

The most common speaker errors, referred to here as simple disfluencies, are well-studied and have been the target of a number of automatic detection approaches, several of which are described in Chapter 5. Simple disfluencies include fillers – filled pauses like “um”, “ah”, and discourse markers like “you know” – as well as speaker edits consisting of a reparandum, an interruption point (IP), an optional interregnum (like “I mean”, indicating a change to the listener), and a repair region, as illustrated in Figure 1.3 (Shriberg, 1994).

![Figure 1.3: Typical edit region structure. In these and other examples, reparandum regions are in brackets (‘[’, ‘]’), interregna are in braces (‘{’, ‘}’), and interruption points are marked by ‘+’.

These reparanda (called edit regions by Johnson and Charniak (2004) and others) can be classified into three main groups:

1. In a repetition (above), the repair phrase is approximately identical to the reparandum.

2. In a revision, the repair phrase alters reparandum words to correct or expand upon the previously stated thought.

EX1: but [when he] + {i mean} when she put it that way

EX2: it helps people [that are going to quit] + that would be quitting anyway
3. In a **restart fragment** (also called a false start), an utterance is aborted and then restarted with a new train of thought.

   EX3: and [i think he's] + he tells me he's glad he has one of those

   EX4: [amazon was incorporated by] {uh} well i only knew two people there

In **simple cleanup** (a precursor to full speech reconstruction), all detected filler words are deleted, and the reparanda and interregna are deleted while the repair region is left intact. While this is a substantial initial step for speech reconstruction, more complex and less deterministic changes are often required for generating fluent and grammatical speech text, as illustrated in the following section.

### 1.2.2 Why simple disfluencies are not enough

We do not believe that simple disfluency cleanup is sufficient for speech reconstruction. In some cases, such as the repetitions mentioned above, simple cleanup is adequate for error corrections. However, simply deleting an identified reparandum region is not always optimal. We would like to consider preserving these reparanda (for restart fragments in particular) if

1. the reparandum contains content words, and

2. its information content is distinct from that in surrounding utterances.

In the first restart fragment example (EX3) in Section 1.2.1, the reparandum introduces no new active verbs or new content, and thus can be safely deleted. The second example (EX4)
CHAPTER 1. INTRODUCTION TO SPEECH RECONSTRUCTION

however demonstrates a case when the reparandum may be considered to have unique and preservable content of its own. For example, resolving the anaphoric reference “there” may require “amazon”. This could more appropriately be segmented away from the original sentence and treated as a separate though incomplete statement.

Additional examples of speaker errors and reconstructions in the form illustrated in Figure 1.3 which go beyond the simple edit region formalism include

EX5: [how can you get experience without] + it’s a catch-22

becomes

how can you get experience without \text{<ARG>}$^1$

it’s a catch-22

EX6: they like video games and stuff some kids do

becomes

some kids do like video games and stuff

or

some kids like video games

In the reconstructions above, parts of the traditional reparandum region are preserved rather than deleted. EX5 splits the false start region into its own utterance, and EX6 requires coreference identification and word reordering, in addition to the cleaning of non-contributing elements. In perfect reconstruction, we would be able to automatically achieve these same transformations automatically.

$^1$Here, the annotator inserted a null-valued argument to complete the utterance.
CHAPTER 1. INTRODUCTION TO SPEECH RECONSTRUCTION

EX6 also demonstrates the ambiguity sometimes present in determining the best possible reconstruction of a given sentence. This is especially true given our potentially conflicting goals of achieving reconstruction with minimal required changes and simplifying any superfluous structure or segments to what is required for meaning preservation.

1.2.3 Construction error types

Even if simple disfluencies could be detected and deleted without error, a large number of speaker-generated construction errors would still remain. In a manual reconstruction effort (described in Chapter 2), we found that 20% of errorfully constructed sentences require more than word deletions to become well-formed. The most common of these changes were phrase movements and word substitutions, neither of which can be detected nor carried out through the frameworks for treating simple disfluencies.

While thorough analysis has been conducted on the prevalence of filler words and edits (reparandum) in spontaneous conversation (Shriberg, 1994), we have found no comprehensive listing or analysis of deeper speech errors such as incomplete sentences, false starts, and the necessity of sentence reordering and word insertions, though some have been alluded to.\(^2\)

Our initial analysis of these issues was a manual study of eighteen conversations from the Fisher CTS corpus (Cieri et al., 2004). Reading through these transcripts and listening to the accompanying audio, 15 construction error types were identified and are listed below.

\(^2\)For example, in Shriberg (1994), asides and final ellipsis were considered to be “degenerate cases” and generally left unanalyzed.
CHAPTER 1. INTRODUCTION TO SPEECH RECONSTRUCTION

Illustrative examples of each type have been taken from the Fisher transcripts.

1. **Filler words and discourse markers:** Three dominant instances are included in the category of filler (Linguistic Data Consortium, 2004; Shriberg, 1994):

   (a) Filled Pauses: hesitation sounds that speakers employ to indicate uncertainty (or, sometimes to maintain control of a conversation while thinking of what to say next). Examples include (i.e. “um” and “eh”).

   (b) Discourse Markers: words or phrases that function primarily as structuring units of spoken language (i.e. “you know”, “like”, and “well”) (Zufferey and Popescu-Belis, 2004).

   (c) Explicit editing terms (EETs): overt statements from the speaker recognizing the existence of an edit disfluency. Examples include “I mean” and “or rather”, etc.

2. **Parenthetical asides:** Speakers occasionally embed qualifying or explanatory remarks within their primary sentence. While some studies have included this phenomenon within the category of fillers (error type #1 above), the issues in identifying and treating these instances warrant a separate treatment.

   **EX7** below is a deletable parenthetical phrase, while **EX8** shows a non-deletable parenthetical phrase.

   **EX7:** some college student brought *i don’t know really* cherry bombs or something on an airplane just to prove a point
CHAPTER 1. INTRODUCTION TO SPEECH RECONSTRUCTION

becomes

some college student brought cherry bombs or something on an airplane just to prove a point

EX8: so i will tell you that the game that i absolutely love i got it for christmas a couple of years ago well almost because it's more than a couple of years ago now was myst.

becomes

so i will tell you that the game that i absolutely love was myst.

i got it for christmas more than a couple of years ago now.

Question & Answer errors: A sub-case of the parenthetical aside phenomenon: once the internal question is answered, the statement of the question is, for most purposes, irrelevant.

EX9: And he almost gave it away that one time with what was it Zora I think.

becomes

He almost gave it away that one time with Zora.

Only a few examples of these errors were found naturally in the corpus data analyzed; given the problem’s complexity and sparsity, we do not specifically seek to solve these question and answer errors in the scope of this reconstruction work.

3. Extraneous phrases: Words spoken during speech not in error, but without contributing content to the given utterance.
CHAPTER 1. INTRODUCTION TO SPEECH RECONSTRUCTION

EX10: i just want to just sleep all weekend you know what i mean

    becomes

    i just want to sleep all weekend

4. Reparanda: Simple edits (as described in Section 1.2.1) which we expect to clean by deletion.

5. Lost content from reparanda: When speech edits occur, deleting the reparandum is not always the most desirable course of action. Restart fragments, while typically inessential to the meaning of the intended sentence, sometimes provide context and background which is lost when deleted during simple cleanup (see Section 5.1.1 and the repeated EX5 below), and should instead be preserved as a separate (possibly incomplete) sentence. Repetition reparanda occasionally give more detailed descriptions of arguments than do their respective repairs, as seen in EX11 below, and this information should be preserved by integrating it into the repair region.

EX5: [how can you get experience without] + it’s a catch-22

EX11: [but the 3-D games] + {uh} the games come with hints

    becomes

    but the 3-D games come with hints

6. Lost pronoun resolution from reparandum: This is a sub-case of Type #5. Our goal here is to ensure that a pronoun’s referring expression is preserved during cleanup.
CHAPTER 1. INTRODUCTION TO SPEECH RECONSTRUCTION

EX12: [and a lot of programs too on t. v. aren’t] + {um} they’re not rated

becomes

a lot of programs on t. v. are not rated

7. **Backtracking constructions:** These utterances include indirect speaker corrections or changes of structure, without an explicit reparandum region. Reconstructions without content loss require more than deletion and often more than reordering words as well. Backtrack constructions appear regularly, though not frequently, in speech (Hindle, 1983).

EX13: that’s where i’m originally from is michigan

becomes

i’m originally from michigan

EX14: we have now been married it will be four years in may

becomes

in may we will have been married for four years

8. **Redundant co-references:** Requires intra-sentence anaphora detection, content ranking, and deletion of the expressions with the least unique meaning.

EX15: but everyone else just to them it’s another soap opera

becomes

but to everyone else it’s another soap opera

EX16: i know myself i panicked
CHAPTER 1. INTRODUCTION TO SPEECH RECONSTRUCTION

becomes

i know i panicked

9. **Phrase-connecting errors**: Well-formed phrases sometimes lack the necessary connective function words and linking verbs to form a legitimate sentence.

EX17: i actually working in new jersey

becomes

i am actually working in new jersey

EX18: i go one two hours a day

becomes

i go for one or two hours a day

10. **Argument ellipsis**: Though some ellipsis occurs naturally in language, we want to detect when an essential argument is unintentionally missing and a placeholder should be inserted.

EX19: still wants to party

becomes

<ARG> still wants to party

EX20: she took them to

becomes

she took them to <ARG>
11. **Argument and adjunct reordering**: In English, word order is used to convey grammatical relations (Comrie, 1981). Accordingly, when referential redundancy is discovered in a sentence, not only should the most information-poor references (e.g. pronouns) be the first to be deleted, but their co-referred arguments may need to be moved into their place (see EX6, repeated below).

There is also an interpretational preference for adjunct phrases to appear as a complement of the verb they modify, to minimize ambiguity (EX22).

EX6: **they like video games and stuff some kids do**

*becomes*

some kids do like video games and stuff

*or*

some kids like video games

EX21: **{you know} every week maybe you would have one**

*becomes*

maybe you would have one every week

12. **Sentence boundary errors**: In spontaneous speech the boundaries identifying where a sentential thought begins and ends are not always well-defined. Automatically segmenting ASR output is an active research area (Liu, 2004; Roark et al., 2006), but missing, extraneous, and misplaced boundaries occur even in manually segmented transcripts, as addressed in Section 2.3.1.
CHAPTER 1. INTRODUCTION TO SPEECH RECONSTRUCTION

EX22: it saves trips you know when you have to have information and get your papers and everything together for school you can sit in the comfort of your own home and go on your computer.

becomes

it saves trips when you have to have information and get your papers and everything together for school.

you can sit in the comfort of your own home and go on your computer.

or

it saves trips.

when you have to have information and get your papers and everything together for school you can sit in the comfort of your own home and go on your computer.

13. **Discourse ordering errors:** During speech recognition, speaker channels are typically analyzed independently. However, discourse tracking may be useful for anaphora resolution and sentence completion. For example, consider the following:

EX23: **Speaker A:** It gives the parents uh

**Speaker B:** Guidelines.

**Speaker A:** Yeah.

Here, while speaker B does provide input, we may want to interpret his words as completing speaker A’s sentence, especially given speaker A’s confirmation.

While an interesting problem, we only found a few examples of this issue in our
study, and thus consider this issue to be outside the scope of our work.

14. **Quotation identification and resolution errors:** Simple processing of the following kind of errors could greatly improve downstream processing. Once again, however, this issue is considered to be outside the scope of this study.

**EX 24:** After you see a movie in the theater and then you see it at home you're like yeah.

Okay.

Whatever.

*becomes*

After you see a movie in the theater and then you see it at home you think it’s no big deal.

**EX 25:** so i looked this was a subject for me that i thought oh boy of all subjects to get.

*becomes*

i looked. *(restart fragment preserved for its unique content)*

this was a subject for me that I was surprised to get.

*or (given restrictions on lexical paraphrases)*

i looked.

this was a subject for me that i thought oh boy of all subjects to get.

15. **Other cleanup:** Includes tense/number agreement, missing or inappropriate determiners, grammar errors, and transcriber error correction.

**EX 26:** I haven’t saw the old one but I saw new one
becomes

I haven’t seen the old one but I saw the new one

EX27: i guess in like black in some areas

corrected to

i guess it might blacken some areas

The many spontaneous speech artifacts listed above, while distinct in nature, can be expressed using a common vocabulary of transformations (deletions, insertions, substitutions, and reorderings, discussed further in Chapter 2). In many cases (such as types #5, 6, 7, 8, 10, 11, 12, 14), shallow semantic or content analysis may also be required.

1.3 Other considerations

1.3.1 Problem scope

The scope of our analysis is limited to adult normal native speakers of American English. The analysis presented is conducted on conversational telephone speech (CTS) data.

We consider only discourse-independent, sentence-internal or sentence-adjacent error-correcting transformations. In this way, we also attempt to limit the world knowledge required to correct sentences: key arguments missing from speech output must be replaced by null placeholders rather than be inferred from the dialogue context, and paraphrasing beyond lexical deletion, insertion, and substitution is kept to a minimum.
CHAPTER 1. INTRODUCTION TO SPEECH RECONSTRUCTION

Correcting errorful automatic speech recognition (ASR) output is a major concern for the speech community at large, and we hope not only to apply speech reconstruction directly to automatically transcribed speech, but also eventually use speech reconstruction to correct some of these recognition errors themselves. For now, however, we consider it essential to first attempt automatic reconstruction on manual speech transcripts. As we are better able to model speaker errors and reconstruction on transcripts improves, future refinements to reconstruction methods will handle the new, ASR-system dependent errors introduced into the text. More discussion of speech reconstruction of ASR output is included in Section 5.3.

In keeping with the emphasis on transcripts, much of this work emphasizes gains possible using textual information from the transcripts without acoustic analysis or information from gestures or visual cues. In focusing on non-acoustic features, we hope to learn more about how utterances with errors differ from utterance without errors, and whether there exists a certain context which could even induce errors in spontaneous speech; were we to identify such a context, researchers could more easily study the psycholinguistic phenomenon which occur during speaker error generation.

Finally, no attempt is made to induce appropriate punctuation or word capitalization in this work, though this is an area of active research elsewhere (Batista et al., 2007; Paulik et al., 2008).
CHAPTER 1. INTRODUCTION TO SPEECH RECONSTRUCTION

1.3.2 Speaker errors across languages

The empirical analysis and automatic error correction pursued in this work is limited to American English speech transcripts. Psycholinguistic studies thus far have shown that speaker disfluencies tend to be similar across languages. There is a set of phenomena that does not exist in American English however: languages where spoken dialect does in fact consistently vary from the written form.

One example of such a language is Arabic, where regional dialects vary greatly from the formal dialect known as “Modern Standard Arabic” which comprises all written text but none of the spoken text (excluding the cases of formal speeches or news broadcasts) (Habash, 2006). While there is little published work on direct dialect normalization to formal form, in work such as (Chiang et al., 2006; Habash and Rambow, 2006) resources for the formal written dialect have served as a bridge to spoken dialect NLP tasks such as parsing and morphological analysis.

Another language with parallel morphologies between spoken and written language is Czech. For example\(^3\), in the sentence

\[
\text{Chtěl bych}_1 \text{ čtyřicet}_2 \text{ deka tvrdého}_3 \text{ sýra}_4 \text{ (spoken)} \rightarrow \\
\text{Chtěl bysem}_1 \text{ štyrycet}_2 \text{ deka tvrdýho}_3 \text{ sejra}_4 \text{ (written)}
\]

(translated as “I would like to have 400 g of hard cheese.”), four colloquial substitutions occur. The cases /bych → bysem/ (conditional marker ("would", 1st person singular) and /tvrdého→tvrdýho/ ("hard") are a manifestation of systematic differences between standard

\(^3\)Thanks to Silvie Cinková for providing this example of Czech non-error reconstruction.
CHAPTER 1. INTRODUCTION TO SPEECH RECONSTRUCTION

and colloquial morphology (e.g. endings, suffixes). The cases /čtyřicet → štyrycet/ (“forty”) and /sýra → sejra/ (“cheese”) have to do with stem vowel and consonant shifts between standard and colloquial Czech.

While these cases are interesting and have implications for natural language processing of speech in those languages, such substitutions happen much more infrequently if at all in English (one example perhaps being the phrase “was like” to express a thought or spoken statement). We will not emphasize such issues in this work.

1.3.3 Evaluating progress

As speech reconstruction progress is made, an evaluation metric which is both easily comparable and monotonic in quality must be determined. Complicating this issue for spontaneous speech reconstruction evaluation is the fact that a given utterance often has multiple transformations which fulfill the goals of reconstruction (well-formed and without content alteration).

We investigate potential error metrics in Section 3.1 and use and compare several throughout this work.

1.3.4 Reconciling terminology

This work covers a broad range of topics and terminology which may be unfamiliar even to experts in some areas covered. Additionally, a wide range of terminology has been
used to describe similar concepts in related work (for example, calling reparandum region words \textit{edits} (Fiscus et al., 2004; Johnson and Charniak, 2004) or \textit{DF} (Shriberg, 1994) or calling an interruption point a \textit{moment of interruption (MOI)} (Levelt, 1983)).

For clarification and disambiguation of common terminology and abbreviations, a glossary and an index of terms are included at the conclusion of this thesis on page 195.

Part-of-speech and syntactic categories used in this analysis are the standard set used in the Penn Treebank treebanking effort (Marcus et al., 1994). These are enumerated for convenience in Appendix I on page 183.

1.4 High level goals of this work

This thesis aims to give the reader a comprehensive understanding of the types, frequencies, and underlying drivers behind speaker-generated errors in spontaneous speech, while applying these insights to statistical frameworks for automatic detection and correction of the given phenomena.

In Chapter 1, we have motivated speech reconstruction research and pointed out benefits of reconstructed speech both on their own (improving readability and comprehension of transcripts) and as input to a downstream natural language processing application like statistical machine translation. After reviewing the standard set of simple disfluency classes, we identified an additional set of construction error types commonly found in spontaneous speech transcripts, while constraining the scope of the work described herein.
CHAPTER 1. INTRODUCTION TO SPEECH RECONSTRUCTION

Our goal for the remainder of this work is to develop as unified an analysis as possible of these problems and to precisely quantify the frequency and reconstructive treatment of these phenomena. We will integrate our findings into automatic approaches for spontaneous speech reconstruction, as discussed in Chapters 6 and 7.

Before attempting to reconstruct speech automatically, we seek to comprehensively understand the problem itself. We carry out a manual reconstruction annotation effort on telephone conversation transcripts for use in this study as well as training for automatic reconstruction systems. The reconstruction annotation is conducted on a standard speech recognition corpus, not only determining final reconstructed strings but also labeling the types and motivations behind all changes made (Chapter 2).

Given this corpus of reference reconstructions, we verify the consistency and accuracy of the annotations, and study the characteristics of the selected reconstructive transformations (Chapter 3). This empirical analysis shows which errors are most frequent and problematic and thus suggests areas for focus in future reconstruction research. A survey of potential metrics for reconstruction progress evaluation are defined and discussed, for application throughout the remainder of the thesis.

The surface transformations recorded in our reconstruction annotations reflect underlying influences in the psycholinguistics and speech production models of the given speakers. In Chapter 4, we review selected theories and seek empirical evidence of both their assumptions and conclusions given the manual reconstructions and corresponding shallow semantic labeling we have collected. This unique large-scale investigation of naturally-
CHAPTER 1. INTRODUCTION TO SPEECH RECONSTRUCTION

occurring spontaneous speaker errors with manual syntactico-semantic analysis yields additional insight into the interaction between spoken language and semantic structure and context.

Given our accumulated knowledge about the types, frequencies, contexts, and drivers of speaker-generated errors in spontaneous speech (including a review in Chapter 5 of previous disfluency identification approaches and how related NLP techniques could be applied to the reconstruction problem), in Chapter 6 we build a system which seeks to automatically identify and correct the most frequent errors. Using a conditional random field classification model and a variety of lexical, syntactic, and shallow semantic features, we show improvement in the correction of these errors over a state-of-the-art system. In Chapter 7 we extend this approach by first building a maximum entropy classifier to identify errorful sentences, and then integrate the two approaches to further improve classification accuracy.

Finally, in Chapter 8 we summarize the contributions of this work and indicate future directions for reconstruction research.
Chapter 2

Linguistic resources for spontaneous speech reconstruction

Before we can automatically reconstruct errorful speech text, we want to know more about the types and prevalence of problems that occur in spontaneous speech such as those identified in Chapter 1. Given the variety of construction issues we hope to address, we determine to build a corpus specific for reconstruction (Section 2.1, 2.2), known as the Spontaneous Speech Reconstruction (SSR) corpus. The initial step for building the SSR is to pre-filter appropriate annotation data to optimize annotation efficiency by extracting a densely errorful data set to analyze (Section 2.3, 2.4). Using a suitable base corpus (Section 2.3) and an appropriate annotation tool (Section 2.5.1), we produce a manually annotated corpus of richly labeled gold standard reconstructions and semantic role label annotations (Sections 2.5-2.7).
2.1 Existing disfluency resources

The overall goal of this work is to build an automatic system which can reconstruct errorful speech text into a more grammatical and clean form. This requires both a thorough understanding of how speakers make errors as well as annotated corpora for system training and evaluation. Given these needs, parallel data labeled with errors and recommended transformations for correcting those errors is required. In this section, we survey available speech corpora with error annotation.

Two existing speech corpora – Switchboard (SWBD) (Godfrey et al., 1992) and Fisher (Cieri et al., 2004) – have been partially treebanked (manually parsed) by linguists at the Linguistic Data Consortium (LDC). These trees include a non-terminal label EDITED which is meant to cover the reparandum region of a simple edit structure (see an example of this annotation in Figure 2.1). However, this data lacks any demarcation of the corresponding repair region, and does not directly address the best approach for correcting speaker-generated errors in general.

The existing language resource most similar to the data necessary for reconstruction was produced by the LDC in preparation for the 2004 NIST Rich Transcription Metadata Extraction (MDE) task on the Fisher conversational telephone speech (CTS) corpus (Cieri et al., 2004; Strassel and Walker, 2004; Linguistic Data Consortium, 2004), and is referred to here as SimpleMDE data. The goals of this task include accurate sentence segmentation and identification of simple disfluencies and reparandum regions of the type described in Section 1.2.1. Unlike in the treebank annotation process, SimpleMDE annotators listened
to audio and therefore may have been more likely to catch variations in speaker intentions for sentence boundaries and other annotations.

While the SimpleMDE corpus is a useful starting point for speaker error identification, the errors labeled are limited to a few types (types #1, 4, and 12 in Section 1.2.3), and no recommendations are made as to how to fix the errors; as we have shown in examples like those for type #5 errors, simply deleting the identified reparandum regions is not always optimal. Similarly, discourse markers such as “you know” can be integral to meaning in certain contexts – and should then be preserved – while other discourse markers can be removed without impact.

Another related data resource, built after the work described in this chapter, was built by researchers at Charles University in Prague for the Companions project (Hajič et al., 2008; Cinková and Hajič, 2009). This resource (Prague Dependency Treebank of Spoken Lan-
CHAPTER 2. LINGUISTIC RESOURCES FOR RECONSTRUCTION

guage, or PDTSL) includes reconstruction for Czech and some English dialogues. PDTSL aligns verbatim text to text cleaned by deletions as well as potential insertions and substitutions. PDTSL also includes reconstruction of Czech spoken words to formal language (a concept described in Section 1.3.2). However, PDTSL does not include alignments on words inserted into the reconstruction or words deleted from the original string, and does not label motivations or error types behind any given move. The error types corrected, while unlisted, are generally less complex than those listed in Section 1.2.3.

2.2 Motivation for building a reconstruction resource

Transforming errorful text using supervised statistical methods requires a gold-standard corpus of manually reconstructed sentences. While some annotated corpora have previously been produced for related problems (as described in Section 2.1), there is a need for developing an expanded linguistic resource before it will be possible to train automatic models to accurately clean and transform speech transcripts.

Additionally, meaningful and comparable evaluation of progress is important in any task. A variety of potential error metrics are described in Section 3.1. We may choose to evaluate our work through these methods such as incorporating BLEU scores, comparing syntax trees, comparing language model scores, or using human evaluations to rate readability and information preservation (Jones et al., 2005). For all but this last method, a set of
CHAPTER 2. LINGUISTIC RESOURCES FOR RECONSTRUCTION

gold standard, manually-built reconstructions will be required for evaluative comparison.

Anticipating the training and evaluation needs ahead as research in this area progresses, we produce a medium-sized corpus of reconstructed and aligned telephone speech text annotated on multiple several levels, referred to as the Spontaneous Speech Reconstruction (SSR) corpus (Fitzgerald and Jelinek, 2008). This resource, shared with the linguistic and natural language processing research communities, includes aligned string transformations and predicate-argument structure.

2.3 Data analysis and selection

As it is necessary to generate our own reconstructed sentence corpus, not only for the research described in this work but with the goal of continued relevance for ongoing research elsewhere, the question of which existing speech corpus to annotate and conduct our research on is nontrivial.

We consider six different corpus types – conversational telephone speech (CTS), recorded meeting data (Morgan et al., 2001; Carletta, 2007), interview transcripts\(^1\), governmental proceedings (Mostefa et al., 2006), course lectures, and seminar speaker transcripts (Glass et al., 2005) – as we search for data which would be most suitable for long term reconstruction efforts, including those conducted directly on the output of automatic speech recognition (ASR) systems.

Our selection criteria includes the following:

\(^1\)From the Malach project; learn more at http://terpconnect.umd.edu/~dlrg/malach/
Table 2.1: Choosing data for reconstruction: Overview of data characteristics. A “?” indicates that results have not been published and could not be otherwise determined without the help of an in-house ASR system.

- **Spontaneously-produced** (i.e. not read) speech

- **Manual transcriptions available**

- **Minimal clean-up data issues** (ex. reasonable and consistent utterance boundaries in the manual transcripts; see Section 2.3.1)

- **Availability of high performing 1-best ASR output** (or at minimum, original audio files available for future ASR analysis). The speakers should have minimal accents and audio conditions should be conducive to accurate ASR output.

- **Ample examples of the construction errors we expect to confront** should exist in the training data such that statistically-based correction techniques can be considered. This may imply speech tasks with an elevated cognitive load, as this has been shown to correlate to the number of and complexity of speaker errors (Goldman-Eisler, 1968; Carletta et al., 1993; Bard et al., 2001).
CHAPTER 2. LINGUISTIC RESOURCES FOR RECONSTRUCTION

- **Availability of additional resources** such as manual parses, manual disfluency annotation, and ASR lattices

- **Task relevance**: Data should be in a domain where highly fluent, grammatical, and content preserved sentences would be of benefit to some end user or application.

Overall, we aim to select and annotate data which will continue to be relevant for related future research.

A broad analysis of the six considered data types can be found in Table 2.1. Task relevance is judged by presuming which datasets would most benefit an end user when grammatical and fully content preserved (e.g. what is likely to be considered for machine translation). After this initial comparison, we chose to eliminate the AMI and ICSI meeting data (due to high WER and problematic segmentation), the Malach interviews (due to a lack of ASR resources, strong foreign accents, and persistent sentence segmentation problems), and the European Parliamentary Proceedings (not considered spontaneous enough, with too few construction errors for our task; see Section 5.2.2 for additional details). Additionally, though the MIT course lectures (Math, Physics, Computer Science, and Artificial Intelligence) seemed promising as a data set, the added problem of dealing with spoken equations during technical lectures would distract us from our primary reconstruction goals. This would be an extremely interesting problem and data set to survey for reconstruction at a later time.

Given these considerations, we opted to focus our reconstruction research on the Fisher data, a well-resourced corpus which has already become a standard for the evaluation of
automatic speech recognition and simple disfluency detection methods. Though less task-relevant than other corpus types, Fisher topic prompts (including “Do you think religion is divisive or is it uniting?” and “If you were in a position of power, how would you improve the job market?”) were expected to reflect a higher speaker cognitive load than other corpora of telephone conversations between strangers, like SWBD.

The SSR corpus annotated in this work builds on the LDC SimpleMDE effort, using the same speech utterances (the development and evaluation sets of the Fisher conversational telephone speech corpus (Cieri et al., 2004)) and giving the annotators access to LDC disfluency labels for consideration during the annotation process. However, SSR goes a step further, explicitly correcting the text and allowing a broader vocabulary of transformation recommendations (including potential insertions, substitutions, and constituent moves) with labeled alignments, all described in Section 2.6. A shallow semantic predicate-argument analysis of the resultant grammatical reconstructions is also provided, as described in Section 2.6.3.

We chose to focus on a subset of this data – the DEV1, DEV2, and EVAL Fisher sub-corpora which served as the development and evaluation sets for the NIST RT04 evaluation – as these have been manually treebanked by the LDC.

### 2.3.1 SU boundary resegmentation

In spontaneous speech, the boundaries identifying where a sentence-like unit (SU) – a unit of speech which expresses a thought or idea but does not necessarily fit the formal
CHAPTER 2. LINGUISTIC RESOURCES FOR RECONSTRUCTION

definition of a grammatical sentence (Fiscus et al., 2004) – begins and ends are not always well-defined. Automatically segmenting ASR output is an active research area (Liu, 2004), but missing, extraneous, and misplaced boundaries often occur even in manually segmented transcripts.

For the Fisher subcorpora we selected to annotate (Section 2.3), multiple sentence segmentations have been produced in the past for the same data.

1. Initially the Fisher data was produced for ASR training and testing, and reference transcripts (and corresponding system output) were thus segmented to allow for sentence-level evaluation of ASR systems.

2. When manual syntactic parses were built for treebanking purposes, annotators at the Linguistic Data Consortium often adjusted utterance boundaries to optimize the context for their produced parses, and included additional constraints (such as one
matrix verb per sentence which split sentences like “he ate and she watched t. v.” into “he ate.” and “and she watched t. v.”).

3. Finally, when the SimpleMDE data was annotated (see Section 2.1), optimal utterance segmentation was considered to be a primary goal, and annotators again had to resegment the transcripts as they saw fit. Additionally, the Simple MDE data included a few utterances never before included in the transcripts, such as comments made outside of the primary telephone conversation (e.g. “hold on i’m on the phone”).

As a result of these conflicting manual segmentations for the same data, decisions needed
CHAPTER 2. LINGUISTIC RESOURCES FOR RECONSTRUCTION

to be made in order to combine resources such as ASR output and word-audio alignments, manual parse trees, and reparandum marked in the SimpleMDE data. Though the SimpleMDE boundaries (3) were the most recent refinement, the trees (2) were considered the most difficult to sufficiently resegment and the boundaries between the two typically agreed. Thus, we chose to resegment the time-aligned ASR transcripts and the SimpleMDE data according to the treebank segmentations (2).

2.4 Extracting a densely errorful corpus

While speaker errors are relatively common, the majority of speaker utterances are error-free. Before investing time and resources repairing speech segments manually, it is to our advantage to first attempt to identify which utterances are poorly constructed (defined in Section 1.2 as being ungrammatical, incomplete, or missing necessary sentence boundaries prior to reconstruction). Extracting these sentences allows us to produce a densely errorful data set for efficient annotation and effective training.

We implemented several approaches to automatically identify these sentences. To evaluate the methods, we randomly sampled 500 sentences from our data set and annotated each sentence $s$ in the sample $S$ as well-formed or poorly-formed, forming a set of poorly constructed sentences $P \subset S$, $P = 136$ (a prior probability of $\frac{136}{500} = 0.272$). We then considered several approaches for utterance-level identification of the poor constructions $P$. 
CHAPTER 2. LINGUISTIC RESOURCES FOR RECONSTRUCTION

he that’s g- uh that’s a relief
E E E E E FL - - --

Figure 2.2: Example of word-level simple disfluency labels, where – denotes a non-error, FL denotes a filler, and E generally denotes errorful words within reparanda.

Each classification method is evaluated by precision and recall – metrics defined in Section 3.1 which penalize false classifications and missed classifications, respectively. For the purpose of filtering sentences containing errors for annotation purposes, we prioritize recall (the percentage of errorful SUs identified) over precision (the percentage of identified SUs which are truly errorful) in order to compile the most complete set of errorful utterances possible, and thus combine the metrics with a weighted F-score, $\beta = 2$ (see Equations 3.8–3.10).

<table>
<thead>
<tr>
<th>Approach</th>
<th>Indicator</th>
<th>Precision</th>
<th>Recall</th>
<th>$F_1$</th>
<th>$F_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple edit detection</td>
<td>$</td>
<td>{\text{edits in } s}</td>
<td>\geq 1$</td>
<td>96.0</td>
<td>73.5</td>
</tr>
<tr>
<td>Parsable by HPSG</td>
<td>$s$ is not parsable</td>
<td>78.7</td>
<td>67.4</td>
<td>72.6</td>
<td>69.4</td>
</tr>
<tr>
<td>Edits $\cup$ HPSG</td>
<td></td>
<td>80.4</td>
<td>87.5</td>
<td>83.8</td>
<td>86.3</td>
</tr>
</tbody>
</table>

Table 2.2: Precision, recall, and $F_\beta$ (defined in Section 3.1) for the identification of poorly constructed sentences in the 500-sentence sample $S$.

Of approaches considered, the union of the Johnson and Charniak (2004) simple disfluency detection system described in Section 5.1.1 – where all sentences with one or more automatically identified “edits” such as those shown in Figure 2.2 were labeled as poor – and a deep linguistic (Head-driven Phrase Structure Grammar or HPSG, described in Section 7.1.2 and Pollard and Sag (1994); Callmeier (2001)) parser output – where all sentences not parsed were considered ill-formed – is found to yield the best overall sentence-level
CHAPTER 2. LINGUISTIC RESOURCES FOR RECONSTRUCTION

identification results with 80.4% precision, 87.5% recall, and 86.3% F2-score, as seen in Table 2.2. Accordingly, it is this combination which is used to extract data to be considered for annotation from the Fisher development and evaluation subcorpora. The 21,456 SUs of the subcorpora (DEV1, DEV2 and EVAL) were pruned down to 6,384 utterances likely to contain errors, to be manually reconstructed by trained annotators. The remainder of SUs are included in the final corpus to keep training data balanced, but they are assumed to have minimal errors with greater complexity than fillers.

We recognize that annotating only a subset rather than the entire data set inherently adds some bias to the error analysis that follows. For example, a speaker error type which is prevalent but not caught by either of the approaches described would not be considered in depth. A subjective analysis of errorful SUs not caught in the original 500 SU random sample does not suggest this is the case, however, and the benefits of resource conservation and the opportunity to annotate more errors than otherwise allowed by the available resources are considered to outweigh these bias risks.

Expanded work on identifying speech construction errors on the utterance level is described in Chapter 7.
2.5 Manually reconstructing spontaneous speech

In a four-month effort, we trained annotators to reconstruct the approximately 6,400 sentences (filtered from the full corpus as described in Section 2.4) from the Fisher CTS corpus for use as training and evaluation sets for future reconstruction endeavors.

2.5.1 Developing the annotation tool

To accomplish efficient and consistent annotation, we built an annotation tool that is task-specific, simple to use (even for annotators with little linguistic training), and capable
of storing and labeling the many types of changes we anticipated during the course of a
given sentence-level reconstruction.

We adapted a prototype annotation tool \(^2\) (built for labeling simple Czech disfluencies
of the type shown in Figure 1.3 of Section 1.2.1). Our revised tool has an expanded set of
allowable word change types and includes capabilities for semantic role labeling of verb
arguments as reconstruction features (see Section 2.6.3). A screen shot of the tool is shown
in Figure 2.3, and a view of the semantic role labeling function appears in Figure 2.4.

The tool features separate modes for sentence reconstruction and predicate-argument
labeling of reconstruction output, as well as a summary screen for reviewing annotation
work accomplished at a glance. As the reconstruction of each sentence begins, the tool
initializes the reconstruction to be the same string as the original utterance, and displays a
1:1 alignment between each reconstructed word and the corresponding original word. The
annotators then transform the reconstruction as required, maintaining links between the
original and reconstructed words and inserting, deleting, substituting, and moving words
as necessary.

Annotators are given access to the original audio files when needed to help reduce in-
terpretational ambiguity, and are able to correct most errors through the available set of
reconstruction actions listed in Section 2.6. The annotators also are presented (via word
highlighting within the tool display) with which of the original words the SimpleMDE
annotators had labeled as reparandum or fillers. This is done with the intention of en-

\(^2\)Core tool designed by Petr Podveský at Charles University in Prague, Czech Republic.
CHAPTER 2. LINGUISTIC RESOURCES FOR RECONSTRUCTION

couraging consistency with previous reparandum annotations and simplifying annotator decisions, while allowing annotators the flexibility to preserve regions previously marked as reparanda. Verbs, extracted from the manual parse trees, were also highlighted in the text to indicate candidate predicates of the SU to be labeled with semantic roles.

2.5.2 Annotation quality control

Because any given ill-formed sentence may have several valid reconstructions (as demonstrated in example EX6 of Section 1.2.3), each sentence is reconstructed independently by two or three annotators. This yields two major benefits: we are better able to ensure annotation quality, and the released data can contain multiple reconstructions to allow for more flexible task evaluation during the course of research. In all cases, annotators are encouraged to make the simplest changes necessary to make the sentence clean and grammatical with minimal change to its meaning and without extra-sentential “world knowledge” (Fitzgerald, 2007).

Annotator agreement of the SSR corpus annotation is investigated in Chapters 3 and 4.
### CHAPTER 2. LINGUISTIC RESOURCES FOR RECONSTRUCTION

<table>
<thead>
<tr>
<th>Feature Dimension</th>
<th>Feature</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word alignment</td>
<td>Connect each word in original string with corresponding word in the reconstruction</td>
<td>2.6.1</td>
</tr>
<tr>
<td>Alignment label</td>
<td>Basic (no change)</td>
<td>2.6.1</td>
</tr>
<tr>
<td>(identifies transformation</td>
<td>Delete {type}</td>
<td></td>
</tr>
<tr>
<td>motivation)</td>
<td>Insert {type}</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Substitute {type}</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Move {type}</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Add SU boundary</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Remove SU boundary</td>
<td></td>
</tr>
<tr>
<td>Reconstruction efficacy</td>
<td>Well-formed and grammatical sentence</td>
<td>2.6.2</td>
</tr>
<tr>
<td></td>
<td>Well-formed fragment with content</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fragment without content</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Backchannel Acknowledgement</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cannot repair utterance</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Unannotated</td>
<td></td>
</tr>
<tr>
<td>Semantic role level</td>
<td>Predicate (active verb) $v \in$ well-formed sentence $s$</td>
<td>2.6.3,</td>
</tr>
<tr>
<td></td>
<td>Mandatory arguments for verb $v$ labeled</td>
<td>2.6.3</td>
</tr>
<tr>
<td></td>
<td>Optional arguments for verb $v$ labeled</td>
<td></td>
</tr>
</tbody>
</table>

Table 2.3: Dimensions of reconstruction annotations for the SSR corpus. For each feature dimension, feature types or brief descriptions are given, along with a reference to sections where further details can be found.

### 2.6 Levels of annotation and a vocabulary of allowable changes

Each sentence is annotated on four levels, summarized in Table 2.3 and described in the following sections.

Note that additional instances of text enrichment, such as adding capitalization and punctuation, were considered to be outside the scope of this work.
2.6.1 Word and alignment level

During reconstruction, the words in each utterance were deleted, inserted, substituted, or moved as required to make the sentence as grammatical as possible without altering the original meaning and without the benefit of extra-sentential context. Sentence-like units (SUs) can also be split into two or merged with adjacent utterances as required. Alignments between the spoken words and their reconstructed “source” words (i.e. in the noisy channel paradigm) are explicitly marked, and for each alteration a corresponding alignment label is chosen. The labels serve to track the problems identified and to enrich training to reproduce these types of transformations.

Alteration labels include:

- **DELETE** words: fillers, repetitions/revisions, false starts, co-reference, leading conjugation, and extraneous phrases

- **INSERT** neutral elements, such as function words like “the”, auxiliary verbs like “is”, or undefined argument placeholders, as in “he wants <ARG>”

- **SUBSTITUTE** words to change tense or number, correct transcriber errors, and replace colloquial phrases (such as: “he was like...” → “he said...”).

- **MOVE** words (within sentence boundaries) and label as adjuncts, arguments, or other structural reorderings.

- **ADD SENTENCE BOUNDARIES** when splitting SUs.
CHAPTER 2. LINGUISTIC RESOURCES FOR RECONSTRUCTION

- REMOVE SENTENCE BOUNDARIES when connecting consecutive SUs.

Unaltered original words are aligned to the corresponding word in the reconstruction with an arc marked BASIC.

2.6.2 Reconstruction efficacy

Once reconstruction is complete, annotators identify the final state of each reconstruction to be at one of five levels of grammaticality and discourse function. This information allows us to limit automatic reconstruction training only to those utterances effectively reconstructed into the desired form, and also allows for closer analysis of problems for which, under the scope and constraints given in Section 1.3.1, reconstruction is not possible.

Efficacy types include:

- Well-formed and grammatical sentence

- Well-formed fragment with content (ex. “last June” or “why not?”)

- Fragment without content (ex. “and it uh”)

  Defined to lack a non-pronoun noun or non-auxiliary, active verb. These SUs likely could be deleted entirely before passing the reconstruction output to a downstream process

- Backchannel acknowledgement (ex. “uh-huh” or “sure”)

  A short (one or two word) SU which is produced to indicate that the speaker is
still paying attention to the conversation, without requesting attention or adding new content to the dialog.

• Cannot repair utterance

This efficacy label is selected when the original SU is either uninterpretable by the annotator or reconstruction would require extra-sentential knowledge (e.g. world knowledge or dialogue context).

Examples of such instances which annotators determined could not be fully reconstructed include

- “so th- the computers i- i- i- it was who- whoever came up with it gave to whomever came up with it”
- “and be first as that like this people without that knowledge do n’t have like just using the internet”
- “but when when you ’re a grandparent or th- say nephews and nieces you get all of the all of the benefits”
- “and she she ’s i ’ve known her for probably oh gosh probably since since ninety-four that she ’s been you know th- that we ’ve been close”

Prior to annotation, each SU is labeled “Unannotated”, but this was not considered a valid final efficacy label for submitted annotations.
2.6.3 Semantic role labeling

One overarching goal of speech reconstruction is to develop machinery to automatically reduce an utterance to its main meaning and then generate clean text. In order to do this, we would like to understand how semantic structure in spontaneous speech transcripts varies from the semantic structure of written text. Thus, this annotation effort also integrates shallow semantic annotation known as semantic role labeling.

For every “Well-formed and grammatical sentence” as defined in Section 2.6.2 (and only those utterances), all non-auxiliary verbs are identified and the corresponding predicate-argument structure is labeled according to the role-sets defined in the LDC PropBank annotation effort (Palmer et al., 2005). More details on this labeling can be found at PropBank role set definitions for given verbs can be reviewed at...
CHAPTER 2. LINGUISTIC RESOURCES FOR RECONSTRUCTION

found in Chapter 4.

2.7 Final refinement of the Spontaneous Speech Reconstruction corpus

The released Spontaneous Speech Reconstruction corpus includes both annotated reconstructions (prefiltered in Section 2.4) and, to minimize sample bias for application to real data, also includes the remaining unfiltered, non-annotated reconstructions from the Fisher corpus source. These are assumed\(^4\) to have no errors except for fillers identified by a rule-based classifier described in Johnson et al. (2004). 45.8%, or \(\frac{6,903}{15,072}\), of these unannotated utterances match or is a combination of words in a list of fifty-two backchannel acknowledgements (enumerated in Figure 7.2 of Chapter 7), and are almost certainly error-free.

The data quality across the original Fisher development and evaluation corpora varied. While Dev1 and Eval had annotations adjudicated between annotators, the Dev2 data did not, leading to some questions regarding the consistency of that data (Harper et al., 2005). In addition, the reconstruction and semantic annotation process described in this chapter was carried out in two distinct batches with non-overlapping annotators: Dev2+Eval and Dev1.

\(^4\)Since prefiltering recall was 87.5 on the 500 SU random sample, it is assumed that around 12% of errorful SUs remain in this larger, unfiltered set.
CHAPTER 2. LINGUISTIC RESOURCES FOR RECONSTRUCTION

To be certain that the data distribution is similar across the SSR train, development, and test partitions, we repartitioned the Fisher DEV1, DEV2, and EVAL into all three. This data, taken from the Fisher DEV1, DEV2, and EVAL subcorpora, is repartitioned into three new subcorpora: 17,162 training sentences (119,693 words), 2,191 sentences (14,861 words) in the development set, and 2,288 sentences (15,382 words) in the test set. Conversation-level resegmentations are fully detailed in Appendix II on page 187.

Some post-annotation data cleanup has been required, though we attempted to do it in an unbiased, standardized, and (where possible) automatic manner. Areas reexamined and corrected include

- Words from an original utterance left unaligned to a reconstructed word by an appropriate link.

- Original words and corresponding reconstructions aligned with a “basic” arc or word movement arc with nonequivalent lexical forms

- The labels on deleted “fillers” outside of a closed filler vocabulary were mapped to “extraneous phrase” (or to “restart” in the case of partial words)

- Verbs occurring in two reconstructions but only annotated for semantic role arguments in one.

These and other phenomena are described in further detail in Chapters 3 and 4.

We intend to continue building and revising the SSR corpus and hope to make the
data available publicly pending LDC legal approval\textsuperscript{5}. The rich data set will facilitate new research efforts beyond this work in the area of reconstructing and representing the structure of spontaneously produced speech.

### 2.8 Summary and conclusions

In preparation to carry out speech reconstruction on spontaneous output, we refined our analysis of the 15 types of speaker errors identified in Chapter 1, and – after determining that pre-existing data resources are inadequate for the changes needed in reconstruction (Section 2.2-1.2.3) – we opted to build our own annotated reconstruction corpus atop Fisher Conversational Telephone Speech transcripts (Section 2.3). In Section 2.4 we determined a reliable approach (recall of 87.5) of automatically identifying sentences likely to have construction error for manual reconstruction annotation.

Sections 2.5 and 2.6 detailed the procedure and constraints under which trained annotators manually reconstructed and labeled the Spontaneous Speech Reconstruction (SSR) corpus. This work yielded a medium-sized, detailed corpus of manual annotations (details summarized in Section 2.7) which will allow us to consider a variety of statistical models in our reconstruction research while continuing to learn about the drivers behind the errors themselves.

As a result of this annotation process, we now have the following new resources to consider for training and evaluation features:

\textsuperscript{5}http://www.clsp.jhu.edu/PIRE/ssr/
CHAPTER 2. LINGUISTIC RESOURCES FOR RECONSTRUCTION

- Two or three independently reconstructed strings for each of 6,384 original speech utterances, plus 15,072 additional utterances considered unlikely to require reconstruction
- Labeled word alignments between reconstructions and the original utterances
- Annotation of actions taken in the reconstruction process
- Sentence-type labels marking reconstruction efficacy
- For well-formed and grammatical reconstructions, semantic role labels for all verbs, identifying required and optional arguments and their sentence position.

In Chapter 3, we will give a thorough empirical analysis of the newly created resource. This includes an evaluation of the inter-annotator consistency of the spontaneous speech reconstructions labeled, and an analysis of the types, frequency, and context of errors identified and the transformations chosen for reconstruction.
Chapter 3

Empirical analysis of speaker errors

In Chapter 2 we described the production of a manually annotated corpus of richly labeled gold standard reconstructions and semantic role label annotations, known as the Spontaneous Speech Reconstruction (SSR) corpus. Given this richly annotated corpus of aligned reconstructions, in this chapter we now seek to analyze and understand the surface characteristics of the errors speakers make in spontaneously generated speech.

We open in Section 3.1 by reviewing evaluation metrics used in related work, which we also consider and use throughout the work in this dissertation, beginning in this chapter. Inter-annotator agreement is evaluated in a number of dimensions in Section 3.2. Section 3.3 includes more general surface analysis of changes made, including the raw frequencies for each transformative reconstruction change and analysis of how non-deletion changes may be generalized.
CHAPTER 3. EMPIRICAL ANALYSIS OF SPEAKER ERRORS

3.1 Evaluating reconstruction efforts

Scoring metrics for tasks such as automatic speech recognition (ASR) output are relatively well-established. ASR evaluation approaches primarily emphasize word error rate via edit distance for accuracy and a variety of confidence measures to determine system certainty that the top scoring hypothesis is indeed correct. As discussed in Section 1.3.3, evaluating reconstructions, which allow for more than one valid “solution” to a given ill-formed construction, is less straightforward.

As spontaneous speech reconstruction progress is made, we aim to find an appropriate measure of success for this task. A selection of potential error metrics are described below, with brief discussions of the anticipated advantages and disadvantages specific to the reconstruction task.

EWER:

Edit word error rate (EWER) (Fiscus et al., 2004) considers only the labeling of reparanda (referred to here as “edits” or “edit words”).

\[
\text{EWER} = \frac{\text{# non-reparanda words falsely labeled as edits} + \text{# reparanda left unlabeled}}{\text{# of reparanda words in reference } r}
\]  

This metric is used in the NIST RT04 metadata extraction evaluations (Ostendorf and Hillard, 2004). It offers the advantage of evaluating only those labels we’re identifying (rather than rewarding primarily error-free text when it lacks error labels), but does not comprehensively evaluate the higher goal of reconstructing the broad list of errors given in
CHAPTER 3. EMPIRICAL ANALYSIS OF SPEAKER ERRORS

Section 1.2.3.

WER/ minWER:

Word error rate (WER) is the standard metric for automatic speech recognition. After aligning two strings (one labeled as the reference \( r \)), WER is calculated as the following:

\[
\text{WER}(r) = \frac{\text{# of deletions, insertions, and substitutions in the alignment}}{\text{length of the reference string } r} \quad (3.2)
\]

\[
\text{minWER} = \min_{\text{references } r \in R} \text{WER}(r) \quad (3.3)
\]

WER is straightforward to calculate and understand and is well-known through the field. However, the metric overly penalizes word reorderings\(^1\) and is not robust considering that multiple string transformations may yield equally true reconstructions. minWER loosens this requirement somewhat by giving the edit distance ratio for the reference most similar to the compared string, but still suffers from some of the weaknesses of WER.

TER/ minTER:

Translation error rate (TER) is an error metric roughly equivalent to WER but with a specific penalty for word reordering, or shifts. Thus, under TER, when a given word or phrase moves from one position to another, it is given a single shift penalty rather than one deletion

\(^1\)For example, if a two-word phrase moves from one position in the sentence to another, the relatively change counts as two deletions from the origin and two insertions at the destination.
and one insertion per word.

\[ \text{TER}(r) = \frac{\text{# of dels, insert'ns, subs, and phrase shifts in the alignment}}{\text{length of the reference string } r} \]  \hspace{1cm} (3.4)

\[ \text{minTER} = \min_{r \in R} \text{TER}(r) \]  \hspace{1cm} (3.5)

TER is a clear extension to WER with a more appropriate penalty for word reorderings (sometimes required for reconstruction), and minTER again allows us to compare a string to multiple references, matching the closest. The disadvantage to this technique lies in the fact that reconstructions are still expected to completely match one reference or another, while in truth it may correctly transform part of the string as seen in one reference while transforming the second half as seen in a second reference. We may want to consider still more flexible evaluation metrics.

**BLEU:**

The BLEU metric (BiLingual Evaluation Understudy) (Papineni et al., 2002) is a standard evaluation metric used for statistical machine translation (SMT). This metric has been shown to have high correlation with human judgements of SMT output quality. To calculate BLEU, we calculate scores for individual substrings of length \( n \) within the string (called \( n \)-grams), seeking \( n \)-gram matches in one of possibly many reference texts. The score can be calculated on a sentence-by-sentence level or, more frequently, averaged over a full corpus.
CHAPTER 3. EMPIRICAL ANALYSIS OF SPEAKER ERRORS

\[
\log \text{BLEU} = B \times \frac{1}{N} \sum_{n=0}^{N} \log \text{ngram\_ratio}(n) \tag{3.6}
\]

\[
\text{ngram\_ratio}(n) = \frac{|n\text{-grams seen in both evaluated and reference texts}|}{|n\text{-grams seen in evaluated text}|} \tag{3.7}
\]

\[= \text{n\text{-gram precision (see Eqn. 3.8)} \]

where \(N\) is the length of the longest \(n\)-gram considered and \(B\) is a brevity penalty assigned to counter the bias the metric otherwise gives to shorter strings.

For reconstruction, this metric offers the advantage of rewarding locally fluent output and allowing for easy comparison against multiple references (as are provided by the SSR corpus). While BLEU is a somewhat controversial evaluation approach for the SMT field (Callison-Burch and Osborne, 2006), spontaneous speech reconstruction is expected to have fewer free variables comparatively (having a default word choice and relatively fixed order). Thus, BLEU and other \(n\)-gram matching metrics like WER and TER may be more meaningful for this task.

**WORD- AND SU-LEVEL PRECISION AND RECALL:**

Precision and recall are widely used measures for evaluating information retrieval and other forms of statistical classification. While precision, defined in Equation 3.8, is a measure of the accuracy of a set of assigned labels, recall (Equation 3.9) is a measure of completeness, or of how many items of class \(c\) are assigned class \(c\).

\[F_\beta\text{-score} \text{ is the weighted harmonic mean of precision and recall. To give precision and recall equal importance (i.e. false labels and missed labels are equally costly), we set } \beta = 1;\]

55
CHAPTER 3. EMPIRICAL ANALYSIS OF SPEAKER ERRORS

to raise the impact of recall we set \( \beta > 1 \) and to reduce its impact as compared to precision we set \( \beta < 1 \). When unspecified, it is assumed that \( \beta = 1 \).

\[
\text{precision } P = \frac{|\text{correct } = \text{class } c \text{ item correctly labeled } c|}{|\text{correct}| + |\text{non-class } c \text{ item falsely labeled } c|} \quad (3.8)
\]

\[
\text{recall } R = \frac{|\text{correct}|}{|\text{correct}| + |\text{class } c \text{ item mislabeled non-}c|} \quad (3.9)
\]

\[
\text{F-score } F_{\beta} = \frac{(1 + \beta^2) \ast P \ast R}{\beta^2 \ast P + R} \quad (3.10)
\]

**Word-level** F-score for reconstruction disfluency identification combines the precision and recall of word-level classification labels such as those defined in Section 2.6.1 of the last chapter. On the other hand, **sentence-like unit (SU)-level** evaluation here refers to the binary classification of whether an utterance is errorful (as defined in Section 1.2), and F-score is calculated accordingly.

**EXACT SU MATCH:**

This metric strictly evaluates the percentage of instances in which a given proposed reconstruction \( s \) exactly matches some reference string \( r \) in a reference set \( R \).

\[
\text{SU_match} = \frac{1}{S} \sum_{s \in S} \max_{r \in R} \delta(s, r) \quad (3.11)
\]

where \( \delta(s, r) \) is an indicator function which equals one when \( s \) and \( r \) are equal strings, and zero otherwise.

The six evaluation metrics described have different advantages and disadvantages when applied to reconstruction, as described. We use EWER only to report related simple dis-
CHAPTER 3. EMPIRICAL ANALYSIS OF SPEAKER ERRORS

fluency identification efforts by (Fiscus et al., 2004) in Chapter 5. MinWER, minTER, and BLEU are used in this chapter to described interannotator agreement in Section 3.2. Precision/recall/F-score measures are used to evaluate the accuracy and coverage of identification approaches in Sections 2.4, 3.3.4, 5.1.1, and automatic reconstruction efforts in Chapters 6 and 7 (which are also evaluated using the Exact SU Match metric).

3.2 Inter-annotator agreement statistics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Original vs. reference set</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLEU</td>
<td>61.3</td>
</tr>
<tr>
<td>minTER</td>
<td>38.7</td>
</tr>
</tbody>
</table>

Table 3.1: Quantifying the amount of changes made during reconstruction. In this table, we show the average BLEU and minTER between the untransformed verbatim original text (full annotated corpus) and the corresponding manual reconstructions in the data filtered in Section 2.4).

Before any data resource can be used for training or testing in a natural language processing task such as reconstruction, it is important to first verify its reliability. In this section, we evaluate that inter-annotator variation and judge that the variance corresponds to the non-determinism of the task and that on the whole, annotator decisions are fairly consistent. Examining the reconstruction annotations produced, and moreover various agreement statistics between annotators reconstructing the same sentence, we see that much variance exists in the set of valid reconstructions for a given sentence.

In Table 3.1, we show the average BLEU and minTER scores (defined in Section 3.1)
CHAPTER 3. EMPIRICAL ANALYSIS OF SPEAKER ERRORS

between the untransformed original text and the corresponding manual reconstructions. These values demonstrate the extent of the changes made during reconstruction. A BLEU score of 61.32 would indicate extremely high overlap in a statistical machine translation (SMT) setting, but is not surprising here given the tendency of reconstructions to contain most of the same words at the same positions as the original string. This score serves primarily as a baseline and point of comparison as we continue our evaluation of SSR. A minTER score of 38.7 is also primarily calculated for later comparison, but also demonstrates a high rate of reconstructive change in our annotations. This number can be interpreted as nearly 40% of words in our pre-filtered annotation corpus undergoing some form of transformation during reconstruction.

<table>
<thead>
<tr>
<th>Statistic</th>
<th>% pairwise matching</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact string match</td>
<td>54%</td>
</tr>
<tr>
<td>Efficacy-type match (see Section 2.6)</td>
<td>84%</td>
</tr>
<tr>
<td>Words contained in one reference and seen in the other</td>
<td>93%</td>
</tr>
<tr>
<td>Same number of words in both reconstructions</td>
<td>56%</td>
</tr>
<tr>
<td>Average minWER (Eqn. 3.3) between reconstructions</td>
<td>16%</td>
</tr>
<tr>
<td>Average minTER (Eqn. 3.5) between reconstructions</td>
<td>14.5%</td>
</tr>
<tr>
<td>Average BLEU (Eqn. 3.6) between reconstructions</td>
<td>84.1%</td>
</tr>
<tr>
<td>Alignment label matches</td>
<td>87%</td>
</tr>
<tr>
<td>Alignment label matches (only changed arcs)</td>
<td>63%</td>
</tr>
<tr>
<td>Alignment $\kappa$ statistic (Eqn. 3.12)</td>
<td>0.67</td>
</tr>
</tbody>
</table>

Table 3.2: Pairwise inter-annotator agreement statistics for manual reconstructions of SSR data (excluding semantic annotation, addressed in Chapter 4).

A set of agreement statistics for SSR can be reviewed in Table 3.2; note that semantic role label annotations will not be evaluated until Section 4.2). The finished annotation
CHAPTER 3. EMPIRICAL ANALYSIS OF SPEAKER ERRORS

product yielded several interesting observations. Pairwise comparisons between any two reconstructions of the same string match exactly just over half of the time (54%), though any given word in one reconstruction appears in the other reconstruction 93% of the time. The average minWER between any pair of corresponding reconstructions is 16% while minTER is 14.5%; the difference of the two shows the impact of word reordering on inter-reconstruction variation, while the relatively high error rates between references help confirm that edit distance between a hypothesis reconstruction and any fixed reconstruction serves as a second baseline as we consider minTER as an evaluation metric.

87% of the time any pair of annotators made the same reconstruction decision for a given word in the original string (i.e. the alignment labels matched). Variance is attributed to the non-determinism of the reconstruction process and indeed to specific annotation styles of the annotators (discussed in Section 3.2.1), some of whom were more likely to delete than to move words, etc.

A standard measure of inter-annotator reliability in a classification task is a kappa statistic, which we calculate between alignment change labels. A kappa statistic $\kappa$ (Cohen, 1960) is an index comparing the agreement between two sets of categorical decisions against that which might be expected by chance, ranging from +1 (perfect agreement) via 0 (no agreement above that expected by chance) to -1 (complete disagreement).

\[
\text{Agreement statistic } \kappa = \frac{|\text{agreement}| - E(|\text{agreement}|)}{\text{Number of trials} - E(|\text{agreement}|)} \quad (3.12)
\]

\[
E(|\text{agreement}|) = \sum_{\text{classes } c \in C} \prod_{\text{annotators } a \in A} p_a(c) \quad (3.13)
\]

where $c \in C$ are the set of label classes (here, those described in Section 2.6.1), $a \in A$ is
our set of annotators, and of course $p_a(c)$ is the prior probability of annotator $a$ assigning the label $c$.

As shown in Table 3.2, our calculated kappa statistic for this set of experiments is 0.67. While there is disagreement in the field as to how to exactly interpret these scores, 0.67 is generally considered to be substantial agreement (Landis and Koch, 1977), though this number is inflated given that alignment labels within a SU are not independent which is a typical assumption of the metric.

3.2.1 Annotator-level agreement statistics

No two annotators reconstructed exactly the same way all of the time (differing on average with a 14.5% minTER), which is not surprising given that only 54% of doubly reconstructed SUs matched exactly (Table 3.2). Table 3.3 illustrates some of the differences in reconstructing patterns of different annotators – and some of the potential evaluation difficulties ahead. Though the annotators didn’t all annotate the same data, which does impact the reconstruction actions chosen, there are clear differences between the annotation types generated. Some annotators varied more than others, and in fact annotation styles and change type preferences varied by annotator. For example, Annotator E made nearly twice the average number of reorderings and substitutions per annotated SU (reordering words 0.33 times per annotated SU on average, or for around 33% of SUs on average.). On the other hand, Annotator C in general made the fewest average changes, and in the end had the least variation from fellow annotators (11.5% minTER).
CHAPTER 3. EMPIRICAL ANALYSIS OF SPEAKER ERRORS

To account for this variation and improve corpus consistency, we eliminate all but the two least deviant annotations (in terms of edit distance between reconstruction) from the SSR training corpus used in automatic reconstruction systems described in Chapters 6 and 7. When conversations are annotated more than twice, those annotations which varied the most from the rest of the set are removed. However, in the SSR test subcorpus, all reconstruction annotations are preserved for evaluation purposes.

3.2.2 Sentence boundary changes

As described in Section 2.6, annotators had the freedom not only to change words within sentences, but also to add and remove sentence-like unit (SU) boundaries from the given LDC parse-derived utterance segmentation (Section 2.3.1).

In the combined sets of annotated data, annotators

- split SUs into two (added a new boundary) 390 times, and
- joined SUs (removed an SU boundary) 37 times.

Though annotators accessed the same transcripts and original audio, they often did not agree on when and where to add or remove SU boundaries. We have ignored these mis-matching reconstructions pending resolution in the overall analysis conducted in this chapter as well as system training in Chapters 6 and 7. In general though, the question of what causes annotators to differ in SU segmentation decisions is a very interesting one. While we do not investigate it thoroughly in this work, we hope that the question is studied
CHAPTER 3. EMPIRICAL ANALYSIS OF SPEAKER ERRORS

more thoroughly in the future.

Specific examples of annotator disagreements regarding SU segmentation are detailed in Appendix III on page 191.

3.3 Analysis of reconstructive transformations

The Spontaneous Speech Reconstruction (SSR) corpus produced in this work includes 6,384 spontaneously spoken sentence-like units (SUs), each annotated two or more times for quality control and future evaluation mechanisms. This resource supplements previously existing LDC manually generated parse trees, transcripts, and edit labels for a subsection of the Fisher corpus. The additions include sentence-level reconstruction with labeled word-level alignments, to be used for future research into deep sentence cleanup for spontaneous speech. In this section, we seek to better understand the types of reconstructive changes required in the SSR annotation of spontaneous speech transcripts, which we hope may yield new quantifiable insights into the structure of disfluent natural speech text.

We first notice the general efficacy of the reconstruction efforts, shown in Table 3.4. Most utterances (76.1%) could be fully reconstructed into both well-formed and grammatical sentences, while an additional 12.5% were made well-formed, but lacked verbs or other essential components to make them full sentences with a predicate-argument structure. Only 1.5% of utterances were deemed irreparable by annotators; examples of such...
CHAPTER 3. EMPIRICAL ANALYSIS OF SPEAKER ERRORS

utterances are included in Section 2.6.2.

Cumulative statistics from reconstruction type analysis of segments of the SSR data can be seen in Table 3.5. We see here, for example, that by far the most common construction change is word deletions, especially filler words, repetitions, and restart fragments. Of the remaining changes, word movement and substitutions are the most frequent.

Other notable characteristics of the reconstructed corpus include:

- 11% of sentences are left unchanged by at least one of the annotators – the precision of our poor construction identification methods in Section 2.4 appears to be quite high (though potentially impacted by annotator expectations of finding errors to correct in the data).

- The average sentence length in our original data is 13.55 words, while in reconstructions average length dropped to 9.57 words. The average reconstruction:original utterance length reduction ratio is 0.73.

- 20% of sentences required more than deletion changes.

3.3.1 Insertions: Identification and correction

Three types of insertions into original transcript are permitted in SSR annotations:

- Insert function word

- Insert neutral verb

- Insert neutral argument
CHAPTER 3. EMPIRICAL ANALYSIS OF SPEAKER ERRORS

Analyzing the annotations, we see that inserted function words are almost always inserted conjunctions and occasionally prepositions and determiners (the three most common: “and”, “that”, and “then”). We hypothesize that detecting and correcting these problems would be more effectively accomplished not within same classification framework as simple disfluencies, but using methods more similar to related work on determiner recovery (Turner and Charniak, 2007).

Inserted neutral verbs tend to be auxiliaries like “are” (progressive verb forms) and “have” (perfect forms with participle) and “will” (future forms) before finite verbs. These too we believe can be better identified independently from the error points following false starts, repetitions, and parentheticals.

Inserted neutral arguments are the most ambiguous of the inserted actions in reconstruction, in part because there is intentionally little information in the inserted form (<ARG>), and also because the inserted arguments could serve as placeholders for both missing nominal and sentential segments. As such, annotator alignments shown that the insertions were generated from verbs, prepositions, adjectives, and other word types.

Examples:

• nominal placeholder: “I had a good <ARG>”

• sentential placeholder: “I wanted to move because <ARG>”

• ambiguous placeholder: “I wanted to move to <ARG>”

(<ARG> could be “Dallas” or “start a new life”)
CHAPTER 3. EMPIRICAL ANALYSIS OF SPEAKER ERRORS

Further study is required to better understand the generation point (error point) within in the original SU which drives or indicates these argument insertions (e.g. the above examples are all sentence-final, which is common though not required) and which indicators of these problems are the most common.

3.3.2 Substitutions: Identification and correction

Three types of substitutions are permitted during reconstruction:

- Replace colloquial phrase
- Substitute tense change
- Substitute transcript/speaker error

Of the three substitution types, tense changes are the most common, and over 70% of the lexical items changed are verbs (exceptions include “kind” → “kinds”, etc.). Morphological analysis will likely be required to make any inroads in identifying substitutions made in the transcripts and reconstructing them to their proper form, especially in terms of tense changes.

Colloquial changes almost entirely are made up of transformations from “was like” (or similar forms) to “said”, etc. These transformations can likely be learned by rules.

Transcript and speaker errors are far more complicated to approach, and are closer than any other error type to the issues which would be encountered while reconstructing output from automatic speech reconstruction. Some errors given this label appear to ac-
CHAPTER 3. EMPIRICAL ANALYSIS OF SPEAKER ERRORS

Actually be morphological errors – further cleaning of the SSR corpus may help to catch such occurrences to keep phenomenon labeling consistent. The six most common transcript/speaker errors are

- “a” → “an”
- “the” → “they”
- “your” → “you’re”
- “you’re” → “your”
- “loosing” → “losing”
- “was” → “were”

As this list demonstrates, these errors vary widely, and will require a number of different approaches in order to solve the reconstructions automatically.

3.3.3 Word reorderings: Identification and correction

Three types of reorderings are permitted into original transcript:

- Move argument
- Move adjunct
- Other grammatical fixes
CHAPTER 3. EMPIRICAL ANALYSIS OF SPEAKER ERRORS

**Move argument:**

Argument phrases which are reordered within sentences are highly correlated with deleted co-referents. Of all reconstructive changes, arguments moved tend to occur later in both original and reconstructed sentences. They tend to move right, which indicates that many of these arguments may either be instances of fronting or otherwise stating an argument once and then using a co-referent pronoun in a more grammatical context later in the sentence, rather than the opposite (first stating a pronoun and then identifying its antecedent).

**Move adjunct:**

Adjunct reordering impacts a wide variety of words and phrases. The most commonly moved adjunct phrase – “probably” – accounts for less than 5% of adjunct reorderings. Adverbs are the most commonly affected (31% of moved adjunct phrases were adverb-final, meaning that the last word of the phrase string was assigned an adverbial part-of-speech tag by the automatic parser). Further study of the data is required to confirm the necessity of these moves for reconstruction.

In Section 4.5.3, we compare the placement of semantic role labels assigned in the SSR annotations, as well as automatic and manual parse phrase constituents, with tokens labeled as reordered adjuncts.
CHAPTER 3. EMPIRICAL ANALYSIS OF SPEAKER ERRORS

**Fix grammar:**

In general, annotators were instructed to move constituents only when necessary, and to always move the word with a lower ranking (i.e. in head-finding rules). For example, if “cat red” were changed to “red cat”, the adjective “red”, not the (head) noun “cat”, would be labeled as the moved constituent.

The primary exception to this rule was the frequent moving of phrases of non-confidence like “I guess” and “I think”, which often are spoken sentence-internal or sentence final, but in fact serve as the root of the complementing sentence and more appropriately belong prior to their complement. For example, consider example EX below.

EX  “I wanted to move to Dallas I guess to start a new life”\(^3\)

  becomes

  “I guess I wanted to move to Dallas to start a new life.”

If an annotator were to follow the previously defined reconstruction guidelines then each word of the phrase “I wanted to move to Dallas” would be marked as a move rather than simply the phrase “I guess”. This would be both time-consuming for the annotator and risked the loss of other reconstruction information, since in our framework only one change could be labeled per word alignment arc. Thus, this change was labeled as being done to “Fix [the] grammar”. We believe that these issues can be identified deterministically, though this has not been shown empirically within the course of this work.

\(^3\)Unreconstructed form has potential for misinterpretation, especially when lacking punctuation.
3.3.4 Filler and reparandum agreement with SimpleMDE annotation

As the SSR corpus was built atop Fisher transcriptions and SimpleMDE simple disfluency annotation, we can compare our annotators’ labeling of filler and reparandum words with the previous annotations done via treebanking and simple disfluency markup. Annotator fillers had precision of 76% and recall of 47%. Most of this discrepancy had to do with the fact that annotators only labeled the fillers they deleted, so backchannel fillers or discourse markers considered to be necessary for sentence interpretation are not identified. Additionally, discourse markers with verbs such as “you know” are not treebanked with the sought \texttt{uh} part-of-speech, so these instances deleted by annotators hurt our precision.

Reparandum precision and recall is 88% and 91%, respectively. Much of the discrepancy here can be explained by the fact that edit words again are often preserved by the annotators as unique content.

3.3.5 Consistency in error boundaries

Ferreira et al. (2004) and others have found that false starts and repeats tend to end at certain points of phrases, which is also found to be generally true for our annotated data (as described in Section 3.3.5, where the final POS and phrase position for the end of each speaker error is studied).
3.4 Summary and conclusions

Given the richly annotated Spontaneous Speech Reconstruction corpus, in this chapter we focus and quantify the surface characteristics of the errors speakers make in spontaneously generated speech. Evaluation metrics used in related work are reviewed in Section 3.1. Interannotator agreement is evaluated in a number of dimensions in Section 3.2. Section 3.3 includes more general surface analysis of changes made, including the raw frequencies for each transformative reconstruction change and analysis of how non-deletion changes may be generalized.
<table>
<thead>
<tr>
<th>Annotator</th>
<th>% well-formed of SUs annotated</th>
<th>minTER vs. rest</th>
<th>Deletions per SU</th>
<th>Insertions per SU</th>
<th>Reorderings per SU</th>
<th>Substitutions per SU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall</td>
<td>81% of 6139</td>
<td>14.5</td>
<td>4.12</td>
<td>0.09</td>
<td>0.18</td>
<td>0.13</td>
</tr>
<tr>
<td>Annotator A</td>
<td>80% of 2463</td>
<td>14.0</td>
<td>4.20</td>
<td>0.09</td>
<td>0.22</td>
<td>0.14</td>
</tr>
<tr>
<td>Annotator B</td>
<td>79% of 2860</td>
<td>17.0</td>
<td>4.06</td>
<td>0.07</td>
<td>0.13</td>
<td>0.08 (l)</td>
</tr>
<tr>
<td>Annotator C</td>
<td>86% of 2163</td>
<td>11.5 (l)</td>
<td>3.38 (l)</td>
<td>0.10</td>
<td>0.05 (l)</td>
<td>0.08 (l)</td>
</tr>
<tr>
<td>Annotator D</td>
<td>80% of 3715</td>
<td><strong>22.8 (h)</strong></td>
<td><strong>4.67 (h)</strong></td>
<td>0.09</td>
<td>0.29</td>
<td>0.17</td>
</tr>
<tr>
<td>Annotator E</td>
<td>83% of 2468</td>
<td>19.9</td>
<td>4.45</td>
<td><strong>0.12 (h)</strong></td>
<td><strong>0.34 (h)</strong></td>
<td><strong>0.22 (h)</strong></td>
</tr>
<tr>
<td>Annotator F</td>
<td>85% of 2074</td>
<td>12.3</td>
<td>3.80</td>
<td>0.11</td>
<td>0.11</td>
<td>0.12</td>
</tr>
<tr>
<td>Annotator G</td>
<td>81% of 2410</td>
<td>12.6</td>
<td>3.58</td>
<td>0.10</td>
<td>0.07</td>
<td>0.11</td>
</tr>
<tr>
<td>Annotator H</td>
<td>85% of 1345</td>
<td>14.4</td>
<td>3.95</td>
<td>0.10</td>
<td>0.07</td>
<td>0.10</td>
</tr>
<tr>
<td>Annotator I</td>
<td>82% of 2139</td>
<td>16.0</td>
<td>4.61</td>
<td>0.11</td>
<td>0.29</td>
<td>0.17</td>
</tr>
<tr>
<td>Annotator J</td>
<td>78% of 2340</td>
<td>15.3</td>
<td>4.00</td>
<td><strong>0.06 (l)</strong></td>
<td>0.12</td>
<td>0.10</td>
</tr>
</tbody>
</table>

Table 3.3: Annotator-level minTER and alteration frequencies compared to other annotators reconstructed the same utterances. Overall, there were around 4.12 deletions made per SU, while insertions, reorderings, and substitutions were only made in around 9%, 18%, and 13% (respectively) of SUs. The lower half of the table however demonstrates the per annotator reconstruction differences. For each column, the annotator with the highest minTER or alteration rate is marked in **bold (h)**, while the lowest minTER or alteration rate is marked in *italics (l).*
CHAPTER 3. EMPIRICAL ANALYSIS OF SPEAKER ERRORS

<table>
<thead>
<tr>
<th>Reconstructed SU Efficacy Types</th>
<th>Percent of reconstructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Well-formed Sentences</td>
<td>76.1%</td>
</tr>
<tr>
<td>Well-formed fragments with content</td>
<td>12.5%</td>
</tr>
<tr>
<td>Contentless fragments</td>
<td>7.2%</td>
</tr>
<tr>
<td>Backchannel acknowledgement</td>
<td>2.7%</td>
</tr>
<tr>
<td>Cannot make sentence well-formed</td>
<td>1.5%</td>
</tr>
</tbody>
</table>

Table 3.4: Reconstruction efficacy distribution
Table 3.5: **Actions taken by annotators during manual reconstruction.** This table illustrates the general distributions of reconstruction error types in the development half of the annotations. Action-specific percentages are out of the total set of reconstruction changes made, thus excluding unaltered words. *: “Insert function word” was initially the default tag when annotators made insertions; these assigned POS tags indicate potential annotator error for insertion labeling.
Chapter 4

Underlying structure of spontaneous speech

The surface transformations recorded in our reconstruction annotations in Chapter 2 reflect underlying influences in the psycholinguistic and speech production models of the given spontaneous speakers. In this chapter we review select theories and seek empirical evidence of both their assumptions and conclusions given the manual reconstructions and corresponding shallow semantic labeling annotation we collect. This unique large-scale investigation of naturally-occurring spontaneous speaker errors with manual semantico-syntactic analysis yields additional insight into the impact of spoken language on semantic structure and context.

This chapter serves as further preparatory ground work for using the Spontaneous Speech Reconstruction (SSR) corpus to further understand theory of speaker errors, as
CHAPTER 4. UNDERLYING STRUCTURE OF SPONTANEOUS SPEECH

well as intelligence feature design for automatic systems for shallow semantic labeling and speech reconstruction.

4.1 Background

Despite recent progress in performance for natural language processing tasks such as machine translation, information extraction, and question answering, relatively few of these systems besides transcription have been applied directly to spontaneous speech. Moreover, there has historically been little consideration of how to analyze the underlying semantic-syntactic structure of speech.

A system would accomplish reconstruction of its spontaneous speech input if its output were to represent, in flawless, fluent, and content-preserved English, the message that the speaker intended to convey. Examples of such reconstructions are seen in the following examples.

EX1: that's uh that's a relief

becomes

that's a relief

EX2: how can you do that without + it's a catch-22

becomes

how can you do that without <ARG>

it's a catch-22
CHAPTER 4. UNDERLYING STRUCTURE OF SPONTANEOUS SPEECH

EX3: they like video games some kids do

  becomes

  some kids like video games

In EX1, reconstruction requires only the deletion of a simple filled pause and speaker repetition (or reparandum (Shriberg, 1994)). The second example EX2 shows a restart fragment, where an utterance is aborted by the speaker and then restarted with a new train of thought. Reconstruction here requires

1. The detection of an interruption point (denoted ’+’ in the example) between the abandoned thought and its replacement,

2. Determination that the abandoned portion contains unique and preservable content and should be made a new sentence rather than be deleted (which would alter meaning)

3. Analysis showing that a required argument must be inserted in order to complete the sentence.

Finally, in the third example EX3, in order to produce the given reconstruction, a system must

1. Detect the anaphoric relationship between “they” and “some kids”

2. Detect the referral of “do” to “like [video games]”

3. Make the necessary word reorderings and deletion of the uninformative lexemes.
CHAPTER 4. UNDERLYING STRUCTURE OF SPONTANEOUS SPEECH

These three examples show varying degrees of difficulty for the task of automatic reconstruction. In each case, we also see that semantics of the reconstruction are more straightforward given existing linguistic theories to analyze than those of the verbatim original string. Such analysis not only informs us of what the speaker intended to communicate, but also reveals insights into the types of errors speakers make when speaking spontaneously and how and where these errors occur. In other words, the semantic labeling of reconstructed sentences, when combined with the reconstruction alignments, may yield new quantifiable insights into the structure of disfluent natural speech text.

In this chapter, we investigate this relationship further. Generally, we seek to answer two questions:

- What generalizations about the underlying structure of errorful and reconstructed speech utterances are possible?
- Are these generalizations sufficiently robust as to be incorporated into statistical models in automatic systems identifying semantic roles in speech text, and how can we define “sufficiently robust”?

We begin by reviewing psycholinguistic analyses of spontaneous speech in Section 4.1.1 and previous work in the automatic labeling of structural semantics (semantic role labels) in Section 4.1.2. We motivate the analysis not only in terms of discovery but also regarding its potential application to automatic speech reconstruction research in Section 4.1.3. In Section 4.2 we again describe the Spontaneous Speech Reconstruction (SSR) corpus and the manual semantic role labeling it includes. Section 4.4 analyzes structural
CHAPTER 4. UNDERLYING STRUCTURE OF SPONTANEOUS SPEECH

differences between verbatim and reconstructed text in the SSR as evaluated by a combi-
nation of manual and automatically generated phrasal constituent parses, while Section 4.5
combines syntactic structure and semantic label annotations to determine the consistency
of patterns and their comparison to similar patterns in the Wall Street Journal-based Propo-
sition Bank corpus (Palmer et al., 2005). We conclude by offering a high level analysis of
discoveries made and suggesting areas for continued analysis in the future.

4.1.1 Psycholinguistic studies of speaker errors

In recent decades, many linguists have attempted to analyze some artifacts of speech
production (Goldman-Eisler, 1968; Laver, 1980; Bock, 1982; Bard et al., 2001; Shriberg,
1994): primarily filled and unfilled pauses (Schachter et al., 1991; Fox Tree and Schrock,
1999), but also deeper errors (Levelt, 1983; Clark and Wasow, 1998; Ferreira et al., 2004).
By studying this work and these theories, we draw upon the observations of linguists to
inform our own analysis and feature development for automatic reconstruction systems.

In early work analyzing disfluencies beyond pauses, Levelt (1983, 1989) presented a
“framework of speech production”. It was proposed that the locations to which a speaker
may retrace, after identifying that a moment of interruption (interruption point) has oc-
curred, are those for which the resulting repair produces a well-formed syntactic coordina-
tion with the verbatim original utterance. In other words, the left hand side of a reparandum
should be chosen such that the reparandum and the repair are of similar phrase types and
could be coordinated by conjunction. This observation is considered again as we develop
CHAPTER 4. UNDERLYING STRUCTURE OF SPONTANEOUS SPEECH

features for automatic error detection and correction systems (Chapters 6 and 7).

In another disfluency study (Hindle, 1983), it is observed that, for any given speaker-produced error,

1. correction strategies must be linguistically rule-governed to ensure that humans can understand and internally correct the verbatim output heard, and

2. linguistic cues must be available to signal the occurrence of a repair and to trigger correction strategies.

These insights motivate the study described in Sections 4.4 and 4.5.

In one particularly compelling study, Ferreira et al. (2004) investigates the impact on speech comprehension of speaker disfluencies. It finds that information associated with misarticulated verbs lingers in a process termed overlay. A model of disfluency processing building phrase structures using tree-adjoining grammars (TAGs) – a grammar formalism presented by Joshi and Schabes (1997), similar to context-free grammars but using trees rather than symbols as the basic rewriting expansion units – is presented. The work compared disfluency processing by listeners to garden-path reanalysis (Fodor and Inoue, 2000), where the internal human processing “parser” initially builds an incorrect syntactic analysis. Upon encountering a word that cannot be integrated into that structure, the mental parser then tries to revise the interpretation, according to the presented theory.

Using these observations, we move on to consider automatic approaches for semantic analysis.
CHAPTER 4. UNDERLYING STRUCTURE OF SPONTANEOUS SPEECH

4.1.2 Semantic role labeling

Figure 4.1: Semantic role labeling for the sentence “some kids like video games”.

Every verb is associated with a set of core and optional argument roles, sometimes called a roleset. For example, the verb "say" must have a sayer and an utterance which is said, along with an optionally defined hearer and any number of locative, temporal, manner, etc. modifiers. The role types of different verbs and their syntactic realizations are direct reflection of underlying semantics (Levin, 1993).

The task of predicate-argument labeling (sometimes called semantic role labeling or SRL) assigns this simple who did what to whom when, where, why, how, etc. structure to sentences (see Figure 4.1 and Table 4.1), often for downstream processes such as information extraction and question answering. Reliably identifying and assigning these roles to grammatical text like news text is an active area of research (Gildea and Jurafsky, 2002; Pradhan et al., 2004, 2008), using training resources like Palmer et al. (2005)’s Proposition Bank (PropBank), a 300k-word corpus with semantic role relations labeled for verbs in the Wall Street Journal section of the Penn Treebank.

According to PropBank specifications, core arguments for each predicate are assigned a corresponding label ARG0-ARG5 (where ARG0 is a prototypical agent, ARG1 is a prototypical patient or theme, etc. (Palmer et al., 2005)).
CHAPTER 4. UNDERLYING STRUCTURE OF SPONTANEOUS SPEECH

<table>
<thead>
<tr>
<th>Predicate Modifier Tag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAU</td>
<td>cause</td>
</tr>
<tr>
<td>DIR</td>
<td>direction</td>
</tr>
<tr>
<td>EXT</td>
<td>extent</td>
</tr>
<tr>
<td>LOC</td>
<td>location</td>
</tr>
<tr>
<td>MNR</td>
<td>manner</td>
</tr>
<tr>
<td>PNC</td>
<td>purpose</td>
</tr>
<tr>
<td>TMP</td>
<td>time</td>
</tr>
<tr>
<td>ADV</td>
<td>general-purpose</td>
</tr>
<tr>
<td>PART</td>
<td>verb particle affecting meaning (e.g. “make up”)</td>
</tr>
<tr>
<td>DIS</td>
<td>discourse connectives</td>
</tr>
<tr>
<td>MOD</td>
<td>modal verbs (e.g. “will have gone”)</td>
</tr>
<tr>
<td>NEG</td>
<td>negation marker</td>
</tr>
<tr>
<td>RCP</td>
<td>reciprocal (e.g. “feed yourself”)</td>
</tr>
</tbody>
</table>

Table 4.1: Predicate-modifying (and non-predicate-specific) argument labels in the Proposition Bank and Spontaneous Speech Reconstruction corpora, labeled in addition to the predicate specific core argument labels ARG0-5. The lower set of tags, while not formally adjunct arguments like those in the first grouping, represent additional modifications not specific to a particular verb lexeme.

Modifying adjunct arguments are defined in Table 4.1 and take a non-ARGN form. The upper half of Table 4.1 shows predicate-modifying (and non-predicate-specific) adjunct argument labels, used in both the Proposition Bank and Spontaneous Speech Reconstruction corpora. The lower set of tags, while not formally adjunct arguments like those in the first grouping, represent additional modifications not specific to a particular verb lexeme, and are labeled in PropBank and in SSR to allow for a more complete annotation. Note that, for the MOD example, “have” is an auxiliary, not a modal. Auxiliaries are considered part of the verb form itself, and thus are not labeled.

A common approach to automatic semantic role labeling is to separate the process into
CHAPTER 4. UNDERLYING STRUCTURE OF SPONTANEOUS SPEECH

two steps: argument identification (role vs. non-role) and argument classification (ARG1 vs. ARG2 vs. LOC, etc.). For both tasks, standard cue features in automatic systems include

1. the lexical identity of the verb in question

2. the syntactic type of the word or phrase

3. analysis of the syntactic path between that verb and the prospective argument, and

4. the position (to the left or to the right) for which the candidate argument falls in respect to its predicate.

In Gildea and Palmer (2002), the effect of parser accuracy on semantic role labeling is quantified, and consistent quality parses were found to be essential when automatically identifying semantic roles on Wall Street Journal text. We investigate the consistency and frequency patterns of these statistics in the SSR data in Sections 4.5.2 and 4.5.4.

4.1.3 Potential benefit of semantic analysis to speech reconstruction

With an adequate amount of appropriately annotated conversational text, we believe methods such as those referred to in Section 4.1.2 could be adapted for transcriptions of spontaneous speech. Furthermore, given a set of semantic role labels on an ungrammatical string, and armed with the knowledge of a set of core semantico-syntactic principles which constrain the set of possible grammatical sentences, we hope to discover and take advantage of new cues for construction errors in automatic spontaneous speech reconstruction.
CHAPTER 4. UNDERLYING STRUCTURE OF SPONTANEOUS SPEECH

There are other potential uses for semantics in reconstruction. In inspecting current state-of-the-art ASR output and transcripts, it is obvious that SU segmentation is still highly inaccurate (Roark et al., 2006). Shallow semantic analysis during reconstruction may help to improve SU segmentation. In cases where verb arguments appear to be missing, a system might consider expanding our search to adjacent SUs. Likewise, if fragment SUs are found, we consider whether they ought to be arguments or adjuncts to the verbs of surrounding SUs. In both of these cases, the removal of a sentence boundary may be required. Alternately, identifying cohesive chunks within a SU might help in partitioning a single SU into several distinct utterances.

Another application of semantics for reconstruction is in redundancy detection. In order to determine whether a speaker self-repair such as a restart or revision is worth preserving (i.e. error type #5 in Section 1.2.3), we will need some meaning-based metric for determining and comparing information content. Some utterances themselves are fragments which contribute no content (such as the SU “that was uh”) and ideally should be removed from the speech transcript altogether.

Given that a fragment contains unique content, we should identify the verb or the type of nominal phrase (i.e. location) and fill in the missing arguments. In some cases this implies resolving noun or verb ellipsis, such as the reconstruction shown in EX2 from the introduction of this chapter, repeated below.

**EX2:** how can you do that without + it’s a catch-22

becomes
CHAPTER 4. UNDERLYING STRUCTURE OF SPONTANEOUS SPEECH

how can you do that without <ARG>

it's a catch-22

4.2 Data

We conducted our experiments on the Spontaneous Speech Reconstruction (SSR) corpus (see Fitzgerald and Jelinek (2008) and Chapter 2), a 82,000-word corpus of reconstruction annotations atop a subset of Fisher conversational telephone speech (CTS) data (Cieri et al., 2004). This data offers the advantage of

- multiple manual reconstructions per sentence (reflecting the non-determinism of reconstructive corrections),
- manual word alignments between corresponding verbatim and cleaned sentence-like units (SUs) which are labeled with transformation types (see Section 4.2.1), and
- annotated semantic role labels on predicates and their arguments for all grammatical reconstructions.

The fully reconstructed portion of the SSR corpus consists of 6,116 SUs and 82,000 words total\(^1\). While far smaller than the 300,000-word PropBank corpus, we believe that this data will be adequate for an short investigation to form an initial characterization of semantic structure of verbatim and reconstructed speech.

\(^1\)There are 6,116 rather than 6,348 SUs due to 268 annotated SUs left unanalyzed given unresolved SU boundary mismatches. See Appendix III on page 191 for further details.
4.2.1 Alignments and alteration labels in the SSR corpus

In the SSR corpus, the words in each reconstructed utterance were manually deleted, inserted, substituted, or moved as required. This was done to make the sentence as grammatical as possible without altering the original meaning and without the benefit of extra-sentential context. Alignments between the verbatim original words and their reconstructed “source” words (i.e. in the noisy channel paradigm) are explicitly defined, and for each alteration a corresponding alteration label is chosen.

Alteration labels as listed in Section 2.6 include:

- **DELETE** words: fillers, repetitions/revisions, false starts, co-reference, leading conjugation, and extraneous phrases

- **INSERT** neutral elements, such as function words like “the”, auxiliary verbs like “is”, or undefined argument placeholders, as in “he wants <ARG>”

- **SUBSTITUTE** words to change tense or number, correct transcriber errors, and replace colloquial phrases (such as: “he was like...” → “he said...”)

- **REORDER** words (within sentence boundaries) and label as *adjuncts, arguments, or other structural reorderings*

Unchanged original words are aligned to the corresponding word in the reconstruction with an arc marked **BASIC**.
4.2.2 Semantic role labeling in the SSR corpus

One fundamental goal of speech reconstruction as defined in Fitzgerald and Jelinek (2008) is to develop machinery to automatically reduce an utterance to its underlying meaning and then generate clean text. In order to do this, we would like to understand how semantic structure in spontaneous speech text varies from the semantic structure of written text. Here, we can take advantage of the semantic role labeling included in the SSR annotation effort.

Rather than attempt to label incomplete utterances or unreconstructable phrases, SSR annotators assigned semantic annotation only to those utterances which could become well-formed and grammatical after manual reconstruction. Therefore, only these utterances (about 76% of the annotated SSR data) can be given a semantic analysis in the following sections. In addition, due to limited resources, semantic analysis for some of the data annotated with reconstruction alignments has not been completed as of this version of the SSR. Altogether, 3,626 of 4,756 (76%) well-formed and grammatical sentences and of all 6,116 (59%) annotated and analyzed sentences were predicate-labeled by annotators.

For these well-formed and grammatical sentences, all (non-auxiliary and non-modal) verbs were identified by annotators and the corresponding predicate-argument structure was labeled according to the role-sets defined in the PropBank annotation effort. In most cases, though some verbs (such as those in “I’m dorming right now” and “that sucks”) were either

\footnote{PropBank roleset definitions for given verbs can be reviewed at http://www.cs.rochester.edu/~gildea/Verbs/}
CHAPTER 4. UNDERLYING STRUCTURE OF SPONTANEOUS SPEECH

out of vocabulary or had a usage beyond the roles defined for PropBank, in which case annotators were advised by linguistics at the LDC as to the most appropriate argument assignment.

We believe the transitive bridge between the aligned verbatim and reconstructed sentences and the predicate-argument labels for those reconstructions (detailed further in Section 4.5) may yield insight into the structure of speech errors and how to extract these verb-argument relationships in verbatim and errorful speech text.

4.3 Interannotator agreement statistics for SSR semantic role labeling

<table>
<thead>
<tr>
<th>Statistic</th>
<th>% pairwise matching</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact string match</td>
<td>53%</td>
</tr>
<tr>
<td>Percentage of utterances labeled</td>
<td>72%</td>
</tr>
<tr>
<td>“Well-formed and grammatical”</td>
<td></td>
</tr>
<tr>
<td>Same number of verbs labeled</td>
<td>85%</td>
</tr>
<tr>
<td>Same verbs annotated</td>
<td>74%</td>
</tr>
<tr>
<td>Same verb role types labeled</td>
<td>62%</td>
</tr>
</tbody>
</table>

Table 4.2: Some pairwise inter-annotator agreement statistics for semantic labels of Fisher data.

Shallow semantic structure (SRL) should be the same or similar between manual reconstructions if content has indeed been preserved. Even when reconstructed strings aren’t exact matches, we note that the verbs labeled for their semantic roles should be approxi-
CHAPTER 4. UNDERLYING STRUCTURE OF SPONTANEOUS SPEECH

mately the same if the meaning is indeed preserved in both reconstructions. For any pair of reconstructions, the same verbs were annotated 85% of the time.

Examples of when this did not happen include instances of “I guess” at the end of a sentence, for which there was often disagreement regarding its contribution to the meaning of the string. At times these were deleted as fillers, and were at other times preserved. In other instances, annotator error was to blame: sometimes verbs were missed and their arguments left unlabeled, or auxiliaries (like “is” in “is taking”) were errorfully labeled as full verbs.

4.4 Syntactic variation between verbatim original and reconstructed strings

Unlike most psycholinguistic analyses of speech production, the analysis presented here is tied to observations in a corpus built from naturally occurring data. We note that these observations are made empirically given automatic analysis of the SSR corpus annotations, though with the caveat that annotated reconstructions potentially contain some errors.

As we begin our analysis, we first aim to understand the types of syntactic changes which occur during the course of spontaneous speech reconstruction. Syntactic evaluation of speech and reconstructed structure is based on the following resources:

1. the manual parse $P_{vm}$ for each verbatim original SU (from SSR)
CHAPTER 4. UNDERLYING STRUCTURE OF SPONTANEOUS SPEECH

Figure 4.2: Manual (a) and automatic (b, c) parses for a verbatim string (a, b) and its manual reconstruction (c)
2. the automatic parse $P_{v_u}$ of each verbatim original SU

3. the automatic parse $P_{r_u}$ of each reconstructed SU

Examples of each are shown in Figure 4.2 for the verbatim string “they like video games some kids do” and its reconstruction “some kids like video games”.

We note that automatic parses (using the state of the art Charniak (1999) parser) of verbatim, unreconstructed strings are likely to contain many errors due to the inconsistent structure of verbatim spontaneous speech (Harper et al., 2005). While this limits the reliability of syntactic observations, it represents the current state of the art for syntactic analysis of unreconstructed spontaneous speech text.

On the other hand, automatically obtained parses for cleaned reconstructed text are more likely to be accurate given the simplified and more predictable structure of these SUs. This observation is unfortunately not evaluable without first manually parsing all reconstructions in the SSR corpus, but is assumed in the course of the following syntax-dependent analysis.

In reconstructing from errorful and disfluent text to clean text, a system makes not only surface changes but also changes in underlying constituent dependencies and parser interpretation. We can quantify these changes in part by comparing the internal context-free structure between the two sets of parses. For example, in the verbatim string automatic parse $P_{v_u}$ shown in Figure 4.2b, the nine expanded non-terminal rules are

- $S \rightarrow \text{NP} \text{ VP}$
- $\text{NP} \rightarrow \text{PRP}$
while in the corresponding reconstruction (also automatic) parse $P_{ra}$ in Figure 4.2c, the four non-terminal rule expansions are

- $S \rightarrow NP \ VP$
- $NP \rightarrow DT \ NNS$
- $VP \rightarrow VBP \ NP$
- $NP \rightarrow JJ \ NNS$

Comparing the two sets of expansions, we see that while only $\frac{4}{9} = 44\%$ of the expansions in $P_{va}$ are preserved in $P_{ra}$, all $\frac{4}{4}$ NT expansions in the reconstruction parse $P_{ra}$ are contained within the verbatim original parse $P_{va}$. This tells us that $P_{va}$ contained extraneous structure as expected, and $P_{ra}$’s structure may possibly be extractable directly from the automatic parse $P_{va}$, even though $P_{ra}$ is not a subtree of $P_{va}$ (or $P_{vm}$, for that matter).

To better understand the amount of internal syntactic structure preserved during reconstruction (according to the automatic and manual parsers), we continuing this tree pairing
analysis for the rest of the development corpus (SSR check set), considering both $P_{vm}$ vs. $P_{ra}$ and $P_{va}$ vs. $P_{ra}$. Statistics are compiled in Table 4.3 and analyzed below.

- Overall, 64.2% (54.0%) of expansion rules in a verbatim speech automatic (manual) parse $P_v$ also occur in reconstruction parses $P_{ra}$, and 92.4% (86.8%) of reconstruction parse $P_{ra}$ expansions come directly from the verbatim speech automatic (manual) parse $P_v$ (from columns two and three of Table 4.3).

- Column four of Table 4.3 shows the rule types most often dropped from the verbatim string parses $P_{vm}$ and $P_{va}$ in the transformation to reconstruction. There appears to be great consistency between the $P_{vm} - P_{ra}$ and $P_{va} - P_{ra}$ comparisons here, at least with the ten most frequent rules. We note that $P_{vm}$ includes empty categories (e.g. in [$NP \rightarrow \text{-NONE-}$]) and EDITED reparandum categories while the automatic parses $P_{va}$ do not. In addition, the $P_{va}$ parses appear to commonly suggest full clause non-terminals for the verbatim parses which are not in turn selected for automatic parses of the reconstruction (e.g. $[SBAR \rightarrow S], [PRN \rightarrow S], [S \rightarrow VP]$). This suggests that these rules may be used to handle errorful structures not seen by the trained grammar.

- Rule-types such as those in column five of Table 4.3 are the most often “generated” in the automatic parse of the string reconstruction $P_{ra}$ (unseen in the manual parse $P_{vm}$ for the verbatim string).

$[S \rightarrow VP]$ is the rule most often seen in $P_{ra}$ but not $P_{vm}$, but only the ninth most
common for \( P_{ra} \) vs. \( P_{va} \). This may indicate a characteristic of the parser (i.e. perhaps the parser often hypothesizes this expansion rule but treebank annotators rarely did).

On the other hand, since rules like \([S \rightarrow NP \ VP], [PP \rightarrow IN \ NP], \) and \([SBAR \rightarrow IN \ S] \) appear in a reconstruction parse but not corresponding verbatim parse at similar frequencies regardless of whether \( P_{vm} \) or \( P_{va} \) are being compared, we are more confident that these patterns are effects of the verbatim-reconstruction comparison and not the specific parser used in analysis. The fact that these patterns occur indicates that it is these common rules which are most often confounded by spontaneous speaker errors.

- Given a Levenshtein alignment between altered rules, the most common changes within a given non-terminal phrase are detailed in column six of Table 4.3.

These patterns can be analyzed as saying that \( NP \) and \( EDITED \) in \( P_{vm} \) are commonly irrelevant or disfluent and followed by a preserved rule (patterns #1, 2, 6, 7, and 9). Additionally, automatic parses often eliminate empty traces preceding rules as seen in pattern #3 (unsurprising since parsers like Charniak (1999) do not produce empty categories like \(-NONE-\)). Pattern #5 indicates that adverb phrases \( ADVP \), typically optional adjuncts, are often dropped from the end of rules when comparing manual parses for verbatim text \( P_{vm} \) to automatic parses for reconstructed text \( P_{ra} \).

In the column six (Levenshtein alignment) results for \( P_{va} : P_{ra} \) analysis, we see that the most common aligned rule changes reflect the most basic of errors: a leading coordinator (#1 and 2) and rules proceeded by unnecessary filler words (#3 and
5). Complementary rules #7 and 9 (e.g. $VP \rightarrow [rule][rule\ SBAR]$ and $VP \rightarrow [rule\ SBAR][rule]$) show that complementing clauses are both added and removed, possibly in the same SU (i.e. a phrase shift), during reconstruction.

### 4.5 Analysis of semantics for spoken language

![Semantic role labeling for the sentence “some kids like video games” and its verbatim source string “they like video games and stuff some kids do”](image)

Figure 4.3: Semantic role labeling for the sentence “some kids like video games” and its verbatim source string “they like video games and stuff some kids do”

To analyze the semantic and syntactic patterns found in speech data and its corresponding reconstructions, we project semantic role labels from strings into automatic and manual parses, and moreover from their post-reconstruction source to the verbatim original speech strings via the SSR manual word alignments, as shown in Figures 4.3 and 4.4.

Again, some of these analyses are parser-dependent, but we anticipate consistent if not consistently accurate treatment of various speech-dependent phenomena by the Charniak (1999) parser.

The automatic SRL mapping procedure from the reconstructed string $W_r$ to related
Figure 4.4: Semantic role labeled automatic parses for a verbatim string (b) and its manual reconstruction (c), using the mapping and propagation procedure in Section 4.5.
CHAPTER 4. UNDERLYING STRUCTURE OF SPONTANEOUS SPEECH

parses $P_r$, $P_v$, and $P_{va}$ and the verbatim original string $W_v$ is as follows.

1. **Tag** each reconstruction word $w_r \in W_r$ with the annotated SRL tag $t_{w_r}$.

   (a) Tag each verbatim word $w_v \in W_v$ aligned to $w_r$ via a `BASIC`, `REORDER`, or `SUBSTITUTE` alteration label with the SRL tag $t_{w_r}$ as well.

   (b) Tag each verbatim word $w_v$ aligned to $w_r$ via a `DELETE REPETITION` or `DELETE CO-REFERENCE` alignment with a *shadow* of that SRL tag $t_{w_r}$ (see the lower tags in Figure 4.3 for an example).

   Note: Verbatim original words $w_v$ with any other alignment label is ignored in this semantic analysis as SRL labels for the aligned reconstruction word $w_r$ do not directly translate to them (for example, a deleted filler word collects no semantic content from the reconstruction, and deleted restart fragments by definition typically have distinctly different semantic content than the text which follows).

2. **Overlay** tagged words of string $W_v$ and $W_r$ with the automatic (or manual) parse of the same string.

3. **Propagate** labels. For each constituent in the parse, if all children within a syntactic constituent expansion (or all but `EDITED` or `INTJ`) has a given SRL tag for a given predicate, we tag that NT with the semantic label information as well.

4. **Trace** label duplication. In PropBank annotations atop the Wall Street Journal Penn Treebank, nodes for empty categories called traces often serve as arguments (as in
CHAPTER 4. UNDERLYING STRUCTURE OF SPONTANEOUS SPEECH

“[ARG0 John,*] tried [ARG0 *trace*] to kick [ARG1 the football]”), but as SSR SRL annotation was conducted directly on strings of words and not tree structures, the semantic roles in that corpus are instead assigned directly to the co-indexed argument of the trace. To more directly compare between SSR and PropBank annotations, we therefore duplicate semantic roles on antecedents onto the trace node, when considering a manual parse with traces labeled.

For the manual parse of the verbatim original string $W_v$ (which includes trace indices), if semantic argument has a trace elsewhere in the parse, we copy traced SRL tags to that co-indexed non-terminal node.

4.5.1 Labeled verbs and their arguments

In the 3,626 well-formed and grammatical SUs labeled with semantic roles in the SSR, 895 distinct verbs were labeled with core and adjunct arguments as defined in Table 4.1 and in Section 4.1.2. The most frequent of these verbs was the orthographic form “s” which was labeled 623 times, or in roughly 5% of analyzed sentences. Other forms of the verb “to be”, including “is”, “was”, “be”, “are”, “re”, “m”, and “being”, were labeled over 1,500 times, or at a rate of nearly one in half of all well-formed reconstructed sentences. The verb frequencies roughly follow a Zipfian distribution (Zipf, 1949), where most verb words appear only once (49.9%) or twice (16.0%)\(^3\).

On average, 1.86 core arguments (ARG[0-4]) are labeled per verb, but the specific argu-

\(^3\)In the discussed analysis, only one semantic annotation is considered per original utterance.
CHAPTER 4. UNDERLYING STRUCTURE OF SPONTANEOUS SPEECH

ment types and typical argument numbers per predicate are verb-specific. For example, the
ditransitive verb “give” has an average of 2.61 core arguments for its 18 occurrences, while
the verb “divorced” (whose core arguments “initiator of end of marriage” and “ex-spouse”
are often combined, as in “we divorced two years ago”) was labeled 11 times with an average
of 1.00 core arguments per occurrence.

In the larger PropBank corpus, annotated atop Wall Street Journal news text, the most
frequently reported verb root is “say”, with over ten thousand labeled appearances in vari-
ous tenses\footnote{The reported PropBank analysis ignores past and present participle (passive) usage; we do not do this in our analysis.}, and again most verbs occur two or fewer times.

4.5.2 Structural semantic statistics in cleaned speech

A true reconstruction of a verbatim spoken utterance can be considered an underlying
form, analogous to that of Chomskian theory or Harris’s conception of transformation (Har-
riss, 1957). In this view, the verbatim original string is the surface form of the sentence, and
as in linguistic theory should be constrained in some manner similar to constraints between
Logical Form (LF) and Surface Form (SF).

In general we seek to determine the semantic structure underlying errorful speech and
what licenses these changes from underlying (intended) form to spoken form during speech
production. In this section, we identify additional trends which may lead to answers to these
questions.

To answer these questions, we seek to identify the following patterns.
The relative frequencies of core and modifying argument assignments in the SSR data.

The most common phrasal category for common arguments in common contexts are listed in Table 4.4.

Note that ARG1 arguments are actually labeled more often than PRED arguments in the data, which indicates that these arguments were very often spread across several phrase tree nodes rather than being collected into a single parent in Step #3 of the mapping procedure outlined at the beginning of Section 4.5. This could be due to parser or semantic annotator error, but most likely is due to the linguistic phenomenon of splitting argument constituents (see Palmer et al., 2005, Section 3.4).

Nearly 1% of PRED labels were assigned to verbatim words whose parses assigned them to nouns. The most common reason for this was errorful parses, such as that shown in Figure 4.5.2. Another reason is substitutions from non-verbs into verbs during the reconstruction process, such as “was like” → “said” and “made stereotypes” → “stereotyped”.

The most frequent syntactic positions for arguments in the SSR are listed in Table 4.5.

Semantic roles like ARG4 defined on a per verb basis so general results should be interpreted with caution. We note again that a single semantic role does not necessarily correspond to a single non-terminal tree node (for example, if an argument is split or in
Figure 4.5: Errorful automatic parse demonstrating predicate (verb) labeled as common noun NN.
CHAPTER 4. UNDERLYING STRUCTURE OF SPONTANEOUS SPEECH

the case of flat tree structures where some children are part of a role but others are not (see Step #3 of the mapping procedure laid out in the opening of Section 4.5).

4.5.3 Structural semantic differences between verbatim speech and reconstructed speech

In this section, we compare the placement of semantic role labels with reconstruction-type labels assigned in the SSR annotations.

These analyses were conducted on \( P_r \) parses of reconstructed strings, the strings upon which semantic labels were directly assigned.

RECONSTRUCTIVE DELETIONS

According to Clark and Wasow (1998), speakers often repeat the first word of major constituents. In this section, we verify and expand on this claim.

Q: Where within the sentence is a redundant and deleted coreference most likely to occur?

In Chapter 3 (see Table 3.5), we catalogue the normalized start positions within the verbatim original sentence of various speaker errors corrected during reconstruction. Among these observations, we see that restart fragments start earliest in the SU on average (which also indicates that, unlike repetitions, they are unlikely to occur later in the SU as well).
CHAPTER 4. UNDERLYING STRUCTURE OF SPONTANEOUS SPEECH

Deleted coreferences also occur fairly early on average (about a quarter into the SU).

**Q:** Is there a relationship between speaker error types requiring deletions and the argument *shadows* contained within?

Only two deletion types – repetitions/revisions and co-references – have direct alignments between deleted text and preserved text and thus can have argument shadows from the reconstruction marked onto the verbatim text.

Of 9,082 propagated deleted repetition/revision phrase nodes from $P_{va}$, we found that 31.0% of arguments within were ARG1, 22.7% of arguments were ARG0, 8.6% of nodes were predicates labeled with semantic roles of their own, and 8.4% of argument nodes were ARG2. Just 8.4% of “delete repetition/revision” nodes were modifier (vs. core) arguments, with TMP and CAU labels being the most common.

Far fewer (353) nodes from $P_{va}$ represented deleted co-reference words. Of these, 57.2% of argument nodes were ARG1, 26.6% were ARG0 and 13.9% were ARG2. 7.6% of “argument” nodes here were SRL-labeled predicates, and 10.2% were in modifier rather than core arguments, the most prevalent were TMP and LOC.

These observations indicate to us that redundant co-references are far most likely to occur for ARG1 roles (most often objects, though also subjects for copular verbs (i.e. “to be”) and others, and appear more likely than random to occur in core argument regions of an utterance rather than in optional modifying material.
CHAPTER 4. UNDERLYING STRUCTURE OF SPONTANEOUS SPEECH

RECONSTRUCTIVE INSERTIONS

Q: When null arguments are inserted into reconstructions of errorful speech, what semantic role do they typically fill?

Three types of insertions were made by annotators during the reconstruction of the SSR corpus. Inserted function words, the most common, were also the most varied. Analyzing the automatic parses of the reconstructions $P_{ra}$, we find that the most commonly assigned parts-of-speech (POS) for these elements was fittingly IN (21.5%, preposition), DT (16.7%, determiner) and CC (14.3%, conjunction). Interestingly, we found that the next most common POS assignments were noun labels, which may indicate errors in labeling.

Other inserted word types were auxiliary or otherwise neutral verbs, and, as expected, most POS labels assigned by the parses were verb types, mostly VBP (non-third person present singular). About half of these were labeled as predicates with corresponding semantic roles; the rest were unlabeled which makes sense as true auxiliary verbs were not labeled in the process.

Finally, around 147 insertion types made were neutral arguments (given the orthographic form <ARG>). 32.7% were common nouns and 18.4% of these were labeled personal pronouns PRP. Another 11.6% were adjectives JJ. We found that 22 (40.7%) of 54 neutral argument nodes directly assigned as semantic roles were ARG1, and another 33.3% were ARG0. Nearly a quarter of inserted arguments became part of a larger phrase serving as a modifier argument, the most common of which were CAU and LOC.
CHAPTER 4. UNDERLYING STRUCTURE OF SPONTANEOUS SPEECH

RECONSTRUCTIVE SUBSTITUTIONS

Q: How often do substitutions occur in the analyzed data, and is there any semantic consistency in the types of words changed?

230 phrase tense substitutions occurred in the SSR corpus. Only 13 of these were directly labeled as predicate arguments (as opposed to being part of a larger argument), 8 of which were Arg1.

Morphology changes generally affect verb tense rather than subject number, and with no real impact on semantic structure. Colloquial substitutions of verbs, such as “he was like...” → “he said...”, yield unusual semantic analysis on the unreconstructed side as non-verbs were analyzed as verbs.

RECONSTRUCTIVE WORD RE-ORDERINGS

Q: How does the predicate-argument labeling relate? If it’s a phrase, what type of phrase is it labeled as?

Word reorderings were labeled as argument movements occurred 136 times in the 3,626 semantics-annotated SUs in the SSR corpus. Of these, 81% were directly labeled as arguments to some sentence-internal predicate. 52% of arguments were Arg1, 17% were Arg0, and 13% were predicates. 11% were labeled as modifying arguments rather than core arguments, which may indicate confusion on the part of the annotators and possibly necessary cleanup.
CHAPTER 4. UNDERLYING STRUCTURE OF SPONTANEOUS SPEECH

Syntactically, 56% of arguments were labeled as nouns or noun phrases, 12% were verbs, and 8% were ADVP, ADJP, or PP.

More commonly labeled than argument movement was adjunct movement, assigned to 206 phrases. 54% of these “moved adjuncts” were not directly labeled as predicate arguments, but were within other labeled arguments. Of those remaining, 48% were unintuitively labeled as core arguments (primarily ARG1 and ARG2), again perhaps indicating errors in the annotation. The most commonly labeled adjunct types were TMP (19% of all arguments), ADV (13%), and LOC (11%).

Syntactically, 25% of reordered adjuncts were assigned ADVP by the automatic parser, 19% were assigned NP, 18% were labeled PP, and remaining common NT assignments included IN, RB, and SBAR.

Finally, 239 phrases were labeled as being reordered for the general reason of fixing the grammar, the default change assignment given by the annotation tool when a word was moved. This category was meant to encompass all movements not included in the previous two categories (arguments and adjuncts), including moving “I guess” from the middle or end of a sentence to the beginning, determiner movement, etc. Semantically, 63% of nodes were directly labeled as predicates or predicate arguments. 34% of these were PRED, 28% were ARG1, 27% were ARG0, 8% were ARG2, and 8% were roughly evenly distributed across the adjunct argument types.

Syntactically, 31% of these changes were NPs, 16% were ADVPs, and 14% were VBPs (24% were verbs in general). The remaining 30% of changes were divided amongst 19
syntactic categories from CC to DT to PP.

4.5.4 Testing generalizations required for automatic semantic role labeling for speech

The results described in Gildea and Palmer (2002) show that parsing dramatically helps during the course of automatic SRL. We hypothesize that the current state-of-art for parsing speech is adequate to generally identify semantic roles in spontaneously produced speech text. For this to be true, features for which SRL is currently dependent on such as consistent predicate-to-parse paths within automatic constituent parses must be found to exist in data such as the SSR corpus.

Using the trees in Figure 4.2, we expand upon the definition of predicate-argument path. For tree (c), for example, the path from predicate VBP → “like” to the argument ARG0 (NP → “some kids”) is [VBP ↑ VP ↑ S ↓ NP]. As trees grow more complex, as well as more errorful (as expected, for example, for automatic parses of verbatim speech text), the paths seen are more sparsely observed (i.e. the probability density is less concentrated at the most common paths than similar paths seen in the PropBank annotations). We thus consider two path simplifications as well:

- **compressed**: only the source, target, and root nodes are preserved in the path (so the path above becomes [VBP ↑ S ↓ NP])
- **POS class clusters**: rather than distinguish, for examples, between different tenses
of verbs in a path, we consider only the first letter of each non-terminal (NT). Thus, clustering compressed output, the new path from predicate to \( \text{ARG0} \) becomes

\[ \text{V} \uparrow \text{S} \downarrow \text{N}. \]

The most frequent predicate-to-argument paths in the SSR corpus are shown in Tables 4.7, 4.8, and 4.6. The top paths are similarly consistent regardless of whether paths are extracted from \( P_{ru} \), \( P_{vm} \), or \( P_{va} \), but we see that the distributions of paths are much flatter (i.e. a greater number and total relative frequency of path types) going from manual to automatic parses and from parses of verbatim to parses of reconstructed strings.

We found that syntactic paths from predicates to arguments were roughly as consistent as those presented for Wall Street Journal data (Palmer et al., 2005), though as expected these patterns degraded when considered for automatically parsed verbatim and errorful data. As a result, we believe that automatic models may be trained, but if entirely dependent on automatic parses of verbatim strings, an SRL-labeled resource much bigger than the SSR and perhaps even PropBank may be required.

### 4.6 Summary

In this chapter, we sought to find generalizations about the underlying structure of errorful and reconstructed speech utterances, in the hopes of determining semantic-based features which can be incorporated into statistical models for reconstruction as well as in automatic systems identifying semantic roles in speech text. Psycholinguistic theories of
speech production errors were surveyed. Interannotator agreement for semantic annotation in the SSR corpus was evaluated, and separately we analyzed syntactic and semantic variation between original and reconstructed utterances according to manually and automatically generated parses and manually labeled semantic roles.

We found that syntactic paths from predicates to arguments were roughly as consistent as those presented for Wall Street Journal data (Palmer et al., 2005), though as expected these patterns degraded when considered for automatically parsed verbatim and errorful data. As a result, we believe that automatic models may be trained, but if entirely dependent on automatic parses of verbatim strings, an SRL-labeled resource much bigger than the SSR and perhaps even PropBank may be required.
Table 4.3: Internal syntactic structure removed and gained during reconstruction. This table compares the rule expansions for each verbatim string parse (manually parsed $P_{vm}$ and automatically parsed $P_{va}$) and the automatic parse of the corresponding reconstruction in the SSR corpus ($P_{ra}$).

<table>
<thead>
<tr>
<th>Verbatim parse $P_{vm}$</th>
<th>$P_v$ rules in $P_{ra}$</th>
<th>$P_r$ rules in $P_v$</th>
<th>Rules most frequently dropped from $P_v$</th>
<th>Rules most frequently added to $P_r$</th>
<th>Levenshtein-aligned expansion changes ($P_v/P_r$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_{vm}$</td>
<td>54.0%</td>
<td>86.8%</td>
<td>1. NP → PRP</td>
<td>1. S → VP</td>
<td>1. S → [NP rule] / [rule]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2. ROOT → S</td>
<td>2. S → NP VP</td>
<td>2. S → [NP VP] / [VP]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4. INTJ → UH</td>
<td>4. PP → IN NP</td>
<td>4. SBAR → [−NONE− S] / [S]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5. PP → IN NP</td>
<td>5. VP → VBZ NP</td>
<td>5. VP → [rule ADVP] / [rule]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>7. NP → −NONE−</td>
<td>7. NP → NP PP</td>
<td>7. S → [EDITED NP VP] / [NP VP]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10. EDVENT → NP</td>
<td>10. ADVP → RB</td>
<td>10. VP → [rule] / [rule SBAR]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$P_{va}$</td>
<td>64.2%</td>
<td>92.4%</td>
<td>1. NP → PRP</td>
<td>1. S → NP VP</td>
<td>1. S → [CC rule] / [rule]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2. ROOT → S</td>
<td>2. PP → IN NP</td>
<td>2. S → [CC NP VP] / [NP VP]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5. PP → IN NP</td>
<td>5. S → NP ADVP VP</td>
<td>5. S → [INTJ NP VP] / [NP VP]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>7. SBAR → S</td>
<td>7. SBAR → S</td>
<td>7. VP → [rule] / [rule SBAR]</td>
</tr>
</tbody>
</table>
### Table 4.4: Most frequent phrasal categories for common arguments in the SSR (mapping SRLs onto automatically parsed verbatim string). Only syntactic categories marked with arguments more than 300 times in the data are listed. *PB05* refers to the PropBank data described in Palmer et al. (2005).
### CHAPTER 4. UNDERLYING STRUCTURE OF SPONTANEOUS SPEECH

<table>
<thead>
<tr>
<th>Data</th>
<th>NT</th>
<th>Total</th>
<th>Four most common argument labels, with rel. frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_{ta}$</td>
<td>NP</td>
<td>10541</td>
<td>ARG1 (48.1%) ARG0 (37.0%) ARG2 (10.1%) Tmp (1.8%)</td>
</tr>
<tr>
<td>$P_{ra}$</td>
<td></td>
<td>10218</td>
<td>ARG1 (46.9%) ARG0 (41.0%) ARG2 (8.9%) Tmp (1.4%)</td>
</tr>
<tr>
<td>$PB05$</td>
<td></td>
<td></td>
<td>ARG2 (34.3%) ARG1 (23.6%) ARG4 (18.9%) ARG3 (12.9%)</td>
</tr>
<tr>
<td>$PB05$</td>
<td>Subj-NP</td>
<td></td>
<td>ARG0 (79.0%) ARG1 (16.8%) ARG2 (2.4%) Tmp (1.2%)</td>
</tr>
<tr>
<td>$PB05$</td>
<td>Obj-NP</td>
<td></td>
<td>ARG1 (84.0%) ARG2 (9.8%) Tmp (4.6%) ARG3 (0.8%)</td>
</tr>
<tr>
<td>$P_{ta}$</td>
<td>PP</td>
<td>1714</td>
<td>ARG1 (34.0%) ARG2 (30.0%) Loc (13.8%) Tmp (7.4%)</td>
</tr>
<tr>
<td>$P_{ra}$</td>
<td></td>
<td>1777</td>
<td>ARG1 (31.2%) ARG2 (29.5%) Loc (14.8%) Tmp (8.8%)</td>
</tr>
<tr>
<td>$PB05$</td>
<td></td>
<td></td>
<td>Loc (46.6%) Tmp (35.3%) Mnr (4.6%) Dis (3.4%)</td>
</tr>
<tr>
<td>$PB05$</td>
<td>PP-at</td>
<td></td>
<td>Ext (34.7%) Loc (27.4%) Tmp (23.2%) Mnr (6.1%)</td>
</tr>
<tr>
<td>$P_{ta}$</td>
<td>ADVP</td>
<td>1519</td>
<td>ARG2 (21.4%) ARG1 (19.4%) Adv (16.9%) Tmp (15.2%)</td>
</tr>
<tr>
<td>$P_{ra}$</td>
<td></td>
<td>1444</td>
<td>ARG2 (22.1%) Adv (20.2%) Tmp (16.3%) ARG1 (12.2%)</td>
</tr>
<tr>
<td>$PB05$</td>
<td></td>
<td></td>
<td>Tmp (30.3%) Mnr (22.2%) Dis (19.8%) Adv (10.3%)</td>
</tr>
<tr>
<td>$P_{ta}$</td>
<td>IN</td>
<td>1213</td>
<td>ARG1 (38.8%) ARG2 (17.6%) CAU (13.1%) Tmp (4.6%)</td>
</tr>
<tr>
<td>$P_{ra}$</td>
<td></td>
<td>671</td>
<td>ARG1 (27.4%) Adv (17.1%) CAU (15.4%) ARG2 (10.3%)</td>
</tr>
<tr>
<td>$PB05$</td>
<td></td>
<td></td>
<td>ARG1 (61.3%) ARG2 (14.1%) Tmp (10.8%) CAU (8.3%)</td>
</tr>
<tr>
<td>$P_{ta}$</td>
<td>SBAR</td>
<td>930</td>
<td>ARG1 (62.3%) ARG2 (12.3%) Tmp (10.2%) CAU (9.9%)</td>
</tr>
<tr>
<td>$P_{ra}$</td>
<td></td>
<td>1241</td>
<td>Adv (36.0%) Tmp (30.4%) ARG1 (16.8%) PRP* (2.4%)</td>
</tr>
<tr>
<td>$PB05$</td>
<td></td>
<td></td>
<td>ARG1 (46.1%) ARG2 (14.1%) Tmp (10.8%) CAU (8.3%)</td>
</tr>
<tr>
<td>$P_{ta}$</td>
<td>VP</td>
<td>895</td>
<td>Pred (48.5%) ARG1 (31.6%) ARG2 (13.3%) CAU (1.6%)</td>
</tr>
<tr>
<td>$P_{ra}$</td>
<td></td>
<td>708</td>
<td>Pred (66.1%) ARG1 (19.8%) ARG2 (10.7%) ARG0 (0.7%)</td>
</tr>
<tr>
<td>$PB05$</td>
<td></td>
<td></td>
<td>NEG (66.8%) ARG1 (10.0%) ARG2 (9.3%) Adv (5.8%)</td>
</tr>
<tr>
<td>$P_{ta}$</td>
<td>RB</td>
<td>763</td>
<td>NEG (73.8%) ARG2 (6.9%) Adv (6.4%) ARG1 (6.3%)</td>
</tr>
<tr>
<td>$P_{ra}$</td>
<td></td>
<td>720</td>
<td>NEG (91.4%) Adjuncts (3.3%) Dis (1.6%) Dir (1.4%)</td>
</tr>
<tr>
<td>$PB05$</td>
<td></td>
<td></td>
<td>NEG (66.8%) ARG1 (10.0%) ARG2 (9.3%) Adv (5.8%)</td>
</tr>
<tr>
<td>$P_{ta}$</td>
<td>ADJP</td>
<td>620</td>
<td>ARG2 (71.9%) ARG1 (21.0%) Pred (2.9%) Mnr (1.0%)</td>
</tr>
<tr>
<td>$P_{ra}$</td>
<td></td>
<td>692</td>
<td>ARG2 (76.3%) ARG1 (16.6%) Pred (3.8%) Mnr (0.9%)</td>
</tr>
<tr>
<td>$PB05$</td>
<td>WHNP</td>
<td>599</td>
<td>ARG1 (61.3%) ARG0 (20.7%) ARG2 (14.9%) Loc (0.8%)</td>
</tr>
<tr>
<td>$PB05$</td>
<td>WHNP</td>
<td>535</td>
<td>ARG1 (60.9%) ARG0 (23.4%) ARG2 (13.1%) Pur (0.6%)</td>
</tr>
<tr>
<td>$PB05$</td>
<td>S</td>
<td>523</td>
<td>ARG1 (69.8%) ARG2 (16.4%) CAU (2.7%) Tmp (2.1%)</td>
</tr>
<tr>
<td>$PB05$</td>
<td>S</td>
<td>526</td>
<td>ARG1 (71.7%) ARG2 (17.3%) Pred (3.0%) Pur (1.9%)</td>
</tr>
<tr>
<td>$PB05$</td>
<td>S</td>
<td>523</td>
<td>ARG1 (76.0%) Adv (8.5%) ARG2 (7.5%) PRP* (2.4%)</td>
</tr>
<tr>
<td>$PB05$</td>
<td>MD</td>
<td>449</td>
<td>Mod (73.1%) ARG1 (18.3%) ARG2 (5.1%) CAU (1.1%)</td>
</tr>
<tr>
<td>$PB05$</td>
<td>MD</td>
<td>427</td>
<td>Mod (86.2%) ARG1 (11.2%) ARG2 (1.2%) ARG0 (0.5%)</td>
</tr>
<tr>
<td>$PB05$</td>
<td>MD</td>
<td></td>
<td>Mod (97.4%) Adjuncts (3.3%) ARG1 (0.2%) Mnr (0.0%)</td>
</tr>
<tr>
<td>$PB05$</td>
<td>TO</td>
<td>414</td>
<td>ARG1 (50.7%) Mod (23.9%) ARG2 (17.6%) Pur (1.9%)</td>
</tr>
<tr>
<td>$PB05$</td>
<td>TO</td>
<td>299</td>
<td>ARG1 (40.1%) Mod (39.8%) ARG2 (13.4%) Pur (1.7%)</td>
</tr>
</tbody>
</table>

Table 4.5: Most frequent argument categories for common syntactic phrases in the SSR (mapping SRLs onto automatically parsed verbatim string). Only syntactic categories marked with arguments more than 300 times in the data are listed. *PB05 refers to the PropBank data described in Palmer et al. (2005). *The PRP tag was not used in the SSR. Non-terms without a row of *PB05 had no results reported in that work.
CHAPTER 4. UNDERLYING STRUCTURE OF SPONTANEOUS SPEECH

<table>
<thead>
<tr>
<th>Argument Path from Predicate</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBP ↑ VP ↑ S ↓ NP</td>
<td>6.09%</td>
</tr>
<tr>
<td>VB ↑ VP ↑ VP ↑ S ↓ NP</td>
<td>5.14%</td>
</tr>
<tr>
<td>VB ↑ VP ↓ NP</td>
<td>4.98%</td>
</tr>
<tr>
<td>VBD ↑ VP ↑ S ↓ NP</td>
<td>3.97%</td>
</tr>
<tr>
<td>VBZ ↑ VP ↑ S ↓ NP</td>
<td>3.07%</td>
</tr>
<tr>
<td><strong>558 more path types</strong></td>
<td>76.76%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Argument Path from Predicate</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>VB ↑ S ↓ NP</td>
<td>9.00%</td>
</tr>
<tr>
<td>VBP ↑ S ↓ NP</td>
<td>6.46%</td>
</tr>
<tr>
<td>VB ↑ VP ↓ NP</td>
<td>6.41%</td>
</tr>
<tr>
<td>VBD ↑ S ↓ NP</td>
<td>4.29%</td>
</tr>
<tr>
<td>VBZ ↑ S ↓ NP</td>
<td>3.12%</td>
</tr>
<tr>
<td><strong>269 more path types</strong></td>
<td>70.72%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Argument Path from Predicate</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>V ↑ S ↓ N</td>
<td>31.25%</td>
</tr>
<tr>
<td>V ↑ V ↓ N</td>
<td>15.78%</td>
</tr>
<tr>
<td>V ↑ V ↓ A</td>
<td>9.00%</td>
</tr>
<tr>
<td>V ↑ V ↓ P</td>
<td>8.37%</td>
</tr>
<tr>
<td>V ↑ V ↓ S</td>
<td>8.32%</td>
</tr>
<tr>
<td><strong>56 more path types</strong></td>
<td>27.28%</td>
</tr>
</tbody>
</table>

Table 4.6: Most common argument-to-predicate paths: automatic parses for reconstructed strings.
## 4. UNDERLYING STRUCTURE OF SPONTANEOUS SPEECH

<table>
<thead>
<tr>
<th>Original string, manual parse</th>
<th>Argument Path from Predicate</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicate-Argument Paths</td>
<td>VBP ↑ VP ↑ S ↓ NP</td>
<td>4.79%</td>
</tr>
<tr>
<td></td>
<td>VB ↑ VP ↓ NP</td>
<td>4.43%</td>
</tr>
<tr>
<td></td>
<td>VB ↑ VP ↑ VP ↑ S ↓ NP</td>
<td>4.27%</td>
</tr>
<tr>
<td></td>
<td>VBD ↑ VP ↑ S ↓ NP</td>
<td>3.13%</td>
</tr>
<tr>
<td></td>
<td>VP ↑ S ↓ NP</td>
<td>2.71%</td>
</tr>
<tr>
<td></td>
<td>661 more path types</td>
<td>80.67%</td>
</tr>
<tr>
<td>Compressed</td>
<td>VB ↑ S ↓ NP</td>
<td>7.76%</td>
</tr>
<tr>
<td></td>
<td>VB ↑ VP ↓ NP</td>
<td>5.78%</td>
</tr>
<tr>
<td></td>
<td>VBP ↑ S ↓ NP</td>
<td>5.26%</td>
</tr>
<tr>
<td></td>
<td>VP ↑ S ↓ NP</td>
<td>5.21%</td>
</tr>
<tr>
<td></td>
<td>VBD ↑ S ↓ NP</td>
<td>3.44%</td>
</tr>
<tr>
<td></td>
<td>302 more path types</td>
<td>72.55%</td>
</tr>
<tr>
<td>POS class+ compressed</td>
<td>V ↑ S ↓ N</td>
<td>28.14%</td>
</tr>
<tr>
<td></td>
<td>V ↑ V ↓ N</td>
<td>14.70%</td>
</tr>
<tr>
<td></td>
<td>V ↑ V ↓ P</td>
<td>7.30%</td>
</tr>
<tr>
<td></td>
<td>V ↑ V ↓ A</td>
<td>7.19%</td>
</tr>
<tr>
<td></td>
<td>V ↑ V ↓ S</td>
<td>5.37%</td>
</tr>
<tr>
<td></td>
<td>63 more path types</td>
<td>37.30%</td>
</tr>
</tbody>
</table>

Table 4.7: Most common argument-to-predicate paths: semantic labels projected onto manual parses for original strings.
### Table 4.8: Most common argument-to-predicate paths: automatic parses for verbatim original strings.

<table>
<thead>
<tr>
<th>Argument Path from Predicate</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBP ↑ VP ↑ S ↓ NP</td>
<td>4.88%</td>
</tr>
<tr>
<td>VB ↑ VP ↑ VP ↑ S ↓ NP</td>
<td>3.89%</td>
</tr>
<tr>
<td>VB ↑ VP ↓ NP</td>
<td>3.75%</td>
</tr>
<tr>
<td>VBD ↑ VP ↑ S ↓ NP</td>
<td>2.80%</td>
</tr>
<tr>
<td>VBZ ↑ VP ↑ S ↓ NP</td>
<td>2.71%</td>
</tr>
<tr>
<td>943 more path types</td>
<td>81.97%</td>
</tr>
<tr>
<td>VB ↑ S ↓ NP</td>
<td>7.28%</td>
</tr>
<tr>
<td>VB ↑ VP ↓ NP</td>
<td>5.83%</td>
</tr>
<tr>
<td>VBP ↑ S ↓ NP</td>
<td>5.29%</td>
</tr>
<tr>
<td>VBD ↑ S ↓ NP</td>
<td>3.48%</td>
</tr>
<tr>
<td>VBZ ↑ S ↓ NP</td>
<td>2.98%</td>
</tr>
<tr>
<td>332 more path types</td>
<td>75.14%</td>
</tr>
<tr>
<td>V ↑ S ↓ N</td>
<td>25.76%</td>
</tr>
<tr>
<td>V ↑ V ↓ N</td>
<td>17.49%</td>
</tr>
<tr>
<td>V ↑ V ↓ A</td>
<td>8.22%</td>
</tr>
<tr>
<td>V ↑ V ↓ V</td>
<td>7.73%</td>
</tr>
<tr>
<td>V ↑ V ↓ P</td>
<td>7.55%</td>
</tr>
<tr>
<td>59 more path types</td>
<td>33.25%</td>
</tr>
</tbody>
</table>
Chapter 5

Reconstruction and related text transformations

In this chapter we present an overview of applied methods related to our goal of automatic reconstruction of spontaneous speech.

We begin by reviewing previous work on simple disfluency identification (Section 5.1), and emphasize the strengths of these approaches which we hope to replicate and weaknesses which we hope to address in this work.

While there were no previous efforts explicitly solving the spontaneous speech reconstruction task defined in Chapter 1, there are several other fields of natural language processing (NLP) which seek to solve tasks with similar string transformations. We investigate what has been done in areas such as paraphrasing, summarization, and statistical machine translation and whether we can apply those techniques to the reconstruction problem (Sec-
he that ‘s g- uh that ‘s a relief
E E E E FL - - - -

Figure 5.1: Example of word-level error class labels, where - denotes a non-error, FL denotes a filler, and E generally denotes errorful words within reparanda. Repeated from Figure 2.2.

Finally, in Section 5.3, we review current automatic speech recognition (ASR) performance on the Fisher data corresponding to the SSR reconstruction annotations, and determine – if reconstruction were to be applied directly to this output in the future – the maximum benefit that reconstruction could provide in terms of selecting competing hypothesized words from a lattice based on reconstructive fluency and grammaticality models.

5.1 Previous efforts in simple edit detection

Most previous efforts in simple edit detection and cleaning have essentially cast the task into a word tagging problem, as shown in Figure 5.1.

Early disfluency detection work:

Early work in automatic detection (such as Bear et al. (1992)) relied on basic pattern matching (i.e. looking for identical sequences of words) to identify speaker errors, then eliminated false candidate errors using semantic and syntactic information. In Nakatani and
Hirschberg (1993), acoustic and prosodic information is specifically targeted as potential cues for instances of self-repair. Though correlations were determined, little improvement in detecting errorful words is achieved. While Core and Schubert (1999) describes a possible framework to identify errors using dialog parser with underlying metarule mechanisms for editing terms, simple repairs, and second-speaker interference, no results are presented.

Heeman and Allen (1999) presents a statistical language model-based approach (Equation 5.5) in which the speech recognition problem is redefined to include the identification of POS tags, discourse markers, speech repairs and intonational phrases. The conditioning context $W_{i-1}$ is partitioned into classes using decision trees.

An extended review of early disfluency identification systems can be found in Fox Tree (1993).

**Parsing to predict edit regions:**

Several manually annotated resources labeling the syntactic structure of speech text, including EDITED subtree identification, are available (such as the Switchboard (SWBD) and Fisher corpora (Godfrey et al., 1992; Cieri et al., 2004)). In work such as Charniak (1999), these treebanks are used to train automatic context-free grammar (CFG) parsers (which make local decisions without consideration of long-distance dependencies). The CFG parsers, in turn, predict phrase constituent structure of new text given the patterns learned from the treebank.

While these parsers might be presumed to predict non-terminal labels such as EDITED
Table 5.1: RT04 reparandum detection EWER results, as reported in Fiscus et al. (2004). Though both results are weak, detection performance is substantially better when systems access reference transcriptions.

<table>
<thead>
<tr>
<th>Data type</th>
<th>EWER</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASR output</td>
<td>75%</td>
</tr>
<tr>
<td>ASR output + transcriptions</td>
<td>42%</td>
</tr>
</tbody>
</table>

as readily as they predict non-terminal labels like NP (noun phrase) or S (sentence), it turns out that this is not the case. Parsers almost never predict edit structure despite the existence of these structures in training\(^1\), and moreover parsers trained on data with simple edits have been found to perform worse than parsers trained on the same data with the \textit{EDITED} subtrees removed. Charniak and Johnson (2001) mentions and Kahn et al. (2005) quantifies the result that, when training on SWBD directly, parsers generally perform worse than when separately detecting and removing disfluent regions, parsing the remainder, and reinserting the disfluent region as a flat phrase.

Therefore, parsing speech data with the hope of directly producing \textit{EDITED} nodes indicating reparandum regions does not appear viable given current parsing methods.

\textbf{RT04 Metadata Extraction evaluation:}

In the 2004 Rich Text Metadata Extraction (RT04) task evaluation, several teams competed directly to best predict extra-lexical features such as utterance boundaries and simple disfluencies like filler words and reparandum regions using the same training and test data

\(^1\)Though there have been no direct studies of this phenomenon, the problem is assumed to be a result of the relative infrequency as well as inconsistency of appearance of these structures, and also due to the fact that (even enriched) context-free parsers cannot capture edits of a context-sensitive nature.
and the same evaluation criteria. Systems predicted this metadata for both direct output from automatic speech recognition (ASR) systems and speech transcripts.

Results from this evaluation, measured as a edit word error rate (EWER, Equation 3.1), are in Table 5.1. Clearly, the task of “edit word” detection performed far better when predicted on transcripts rather than ASR output alone. The top-performing system, both on ASR output and using manual transcriptions, is described in Section 5.1.1.

5.1.1 Edit detection via noisy channel TAG models

Stochastic approaches for simple disfluency detection use features such as lexical form, acoustic cues, and rule-based knowledge. Most recent state-of-the-art methods for edit region detection such as (Johnson and Charniak, 2004; Liu et al., 2004; Zhang and Weng, 2005; Honal and Schultz, 2005) use a text-based noisy channel approach. In a noisy channel model we assume that an unknown but fluent string $F$ has passed through a disfluency-adding channel to produce the observed disfluent string $D$, and we then aim to recover the most likely input string $\hat{F}$, defined in Equation 5.3 as

$$\hat{F} = \arg \max_F P(F|D)$$

$$= \arg \max_F P(D|F)P(F)$$

where $P(F)$ represents a language model defining a probability distribution over fluent “source” strings $F$, and $P(D|F)$ is the channel model defining a conditional probability distribution of observed sentences $D$ which may contain the types of construction errors
described in the previous subsection.

The Johnson and Charniak (2004) approach, referred to in this dissertation as JC04, combines the noisy channel paradigm with a tree-adjoining grammar (TAG) (Joshi and Schabes, 1997) to capture approximately repeated elements. The TAG approach models the crossed word dependencies observed when the reparandum incorporates the same or very similar words in roughly the same word order, which Johnson and Charniak (2004) refer to as a rough copy (RC). Our version of their system does not use external features for reranking, as used in the related RT04 evaluation system (Johnson et al., 2004), but otherwise appears to produce comparable results to those reported.

While much progress has been made in simple disfluency detection in the last decade, even top-performing systems continue to be ineffective at identifying words in reparanda. To better understand these problems and identify areas for improvement, we used the top-performing JC04 noisy channel TAG edit detector to produce edit detection analyses on the test segment of the Spontaneous Speech Reconstruction (SSR) corpus (Fitzgerald and Jelinek, 2008). Table 5.2 demonstrates the performance of this system for detecting filled pause fillers, discourse marker fillers, and edit words. The results of a more granular analysis compared to a hand-refined reference (see Figure 5.2) are shown in Table 5.3. The reader will recall from Section 3.1 that precision \( P \) is defined as \( P = \frac{|\text{correct}|}{|\text{correct}| + |\text{false}|} \) and recall \( R = \frac{|\text{correct}|}{|\text{correct}| + |\text{miss}|} \). We again denote the harmonic mean of \( P \) and \( R \) as F-score \( (F) \) and calculate it as \( F_1 = \frac{2}{1/P + 1/R} \) (i.e. \( \beta = 1 \)).

\[ \text{As determined in the NIST RT04 EARS Metadata Extraction Task (Fiscus et al., 2004)} \]
CHAPTER 5. RECONSTRUCTION AND RELATED TEXT TRANSFORMATIONS

Figure 5.2: Refined word-level error class labels (compare to Figure 5.1), where ‘–’ denotes a non-error, FL denotes a filler, E generally denotes reparanda, and RC and NC indicate rough copy and non-copy speaker errors, respectively.

Table 5.2: JC04 disfluency detection performance on SSR test data

<table>
<thead>
<tr>
<th>Label</th>
<th>% of words</th>
<th>Precision</th>
<th>Recall</th>
<th>F$_1$-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fillers</td>
<td>5.6%</td>
<td>64%</td>
<td>59%</td>
<td>61%</td>
</tr>
<tr>
<td>Edit (reparandum)</td>
<td>7.8%</td>
<td>85%</td>
<td>68%</td>
<td>76%</td>
</tr>
</tbody>
</table>

Table 5.3: JC04 disfluency detection: rough copies vs. non-copies. A rough copy is defined as a speaker error where the reparandum incorporates the same or very similar words in roughly the same word order as the corresponding repair. A non-copy is a speaker error where the reparandum has no lexical or structural relationship to the repair region following.

<table>
<thead>
<tr>
<th>Label</th>
<th>% of edits</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rough copy (RC) edits</td>
<td>58.8%</td>
<td>84.8%</td>
</tr>
<tr>
<td>Non-copy (NC) edits</td>
<td>41.2%</td>
<td>43.2%</td>
</tr>
<tr>
<td>Total edits</td>
<td>100.0%</td>
<td>67.6%</td>
</tr>
</tbody>
</table>

As expected given the assumptions of their approach, JC04 identifies repetitions and most revisions in the SSR data, but less successfully labels false starts and other speaker self-interruptions which do not have cross-serial correlations. These non-copy (NC) errors (with a recall of only 43.2%), hurt the overall edit detection recall score. Precision (and thus F-score) cannot be calculated for the experiment in Table 5.3; since the JC04 does not explicitly label edits as rough copies or non-copies, we have no way of knowing whether words falsely labeled as edits would have been considered as false RCs or false NCs. This
CHAPTER 5. RECONSTRUCTION AND RELATED TEXT TRANSFORMATIONS

will unfortunately hinder us from using JC04 as a direct baseline comparison in our work targeting false starts; however, we consider these results to be further motivation for our work.

Surveying these results, we conclude that there is still much room for improvement in the field of simple disfluency identification, especially the cases of detecting non-copy reparandum and learning how and where to implement non-deletion reconstruction changes.

5.2 Viewing reconstruction through other NLP techniques

While the simple error detection and cleanup described so far can be approached as a word tagging task, full speech reconstruction as defined in Chapter 1 is essentially a string transformation task with specific lexical and semantic constraints. We survey other natural language processing fields which share string transformation as a goal. For each, standard approaches are described and compared to the constraints of the speech reconstruction task at hand.


5.2.1 Reconstruction by paraphrasing

A paraphrase is an alternate phrasing used to convey identical information. Automatic paraphrase generation research has two primary aims:

- to identify repeated information between documents (e.g. see if two sentences are paraphrases of each other in the context of multi-document summarization or question answering), and
- to produce a greater variety of text styles and wordings for natural language generation (NLG).

The earliest paraphrasing work emphasized synonym replacements on the word-level (Pereira et al., 1993; Lin, 1998) or extracted inference rules (such as “X wrote Y” implies “X is the author of Y”) (Lin and Pantel, 2001), neither of which appear promising for speech reconstruction. Other approaches, however, consider paraphrasing on a more robust level.

The sentence compression model

Sentence compression paraphrasing is intended as a subcomponent of document summarization: the aim is text reduction on the sentence level. In this model, sentences are shortened by detecting less essential elements and deleting them, which at least on the surface is a good match for the goals of speech reconstruction (in terms of simple disfluency removal). Sentence compression in (Knight and Marcu, 2000; Turner and Charniak, 2005) is accomplished by training a noisy channel transformation model on short “source” parse
CHAPTER 5. RECONSTRUCTION AND RELATED TEXT TRANSFORMATIONS

“target” (a)  
S  
NP  
|  
|  
NNP  
john  
|  
VP  
|  
|  
VBD  
saw  
|  
|  
NP  
mary  
|  
|  
IN  
in  
|  
|  
DT  
the  
|  
|  
NN  
park

“source” (b)  
S  
NP  
|  
|  
NNP  
john  
|  
VP  
|  
|  
VBD  
saw  
|  
|  
NP  
mary

Figure 5.3: Sentence compression paraphrasing example. Tree (a) is the parse of the original (target) tree \( t \), and tree (b) is the parse of its reduction \( s \) (the source in the noisy channel model). The relevant transform rule between the two is \( P(S \rightarrow NP \ VP \ PP \mid S \rightarrow NP \ VP) \).

trees \( s \in S \) and the long, original “target” trees \( t \in T \):

\[
s = \arg \max_s P(s \mid t) \tag{5.1}
\]

\[
= \arg \max_s \frac{P(s) * P(t \mid s)}{P(t)} \tag{5.2}
\]

\[
= \arg \max_s P_{\text{tree}}(s) * P_{\text{expand,tree}}(t \mid s) \tag{5.3}
\]

where the equality between Equation 5.2 and Equation 5.3 is possible given that \( P(t) \) does not impact \( \arg \max_s \). In this manner, the model learns specific context-free rules which appear in a reduced form in the source tree as compared to the target tree (ex. \( P(S \rightarrow NP \ VP \ PP \mid S \rightarrow NP \ VP) \))

While the summarization output produced by this model received strong feedback in human evaluations, the primary drawback for using this method for spontaneous speech reconstruction (besides the fact that only deletions are possible) is its reliance on accurate parse trees to learn transfer rules. Though accurate parsing of spontaneous speech is an

---

3This transformation can also written \( S \rightarrow [NP \ VP \ PP]/[NP \ VP] \), as used in Section 4 syntactic comparisons such as those in Table 4.3.
active area of research (Harper et al., 2005), initial comparisons between automatic parses of both original and reconstructed trees not only demonstrated many parsing errors, but showed that the difference between the automatic parses is rarely the clean removal of subtrees from the original string’s parse. Given this observation, paraphrasing via parse compressions does not appear promising for the reconstruction problem at hand.

**Paraphrase by string alignments**

Another common approach to paraphrasing involves corpus-based extraction of similar sentences which are then aligned to produce candidate paraphrases on the phrase or sentence level (Barzilay and McKeown, 2001; Ibrahim et al., 2003; Power and Scott, 2005; Wan et al., 2005; Barzilay and Lee, 2003). In Barzilay and McKeown (2001) this alignment is done in an approach similar to alignments for SMT given multiple translations of the same novel, identifying noun phrases (NPs) and verb phrases (VPs) via a chunker, and then seeking similar context between matching pairs.

The work described in Barzilay and Lee (2003) expanded this approach, using unsupervised Multiple Sequence Alignment (MSA) methods between different but related news articles to produce a lattice (compact graph-based representation) of sentences within a common cluster. The lattice compresses commonalities between the given strings, and the *backbone nodes* (shared by more than 50% of the cluster’s sentences) serve to identify regions of variability as arguments. Lattices with many common arguments become matching lattices, and paraphrases are produced by substituting the argument values of the
original sentence aligned to a lattice into the backbone slots of a matching lattice.

While training these models for robust transformation is highly data intensive, such approaches may become valuable especially in terms of correcting missing function words and auxiliaries, substituting for morphology errors and colloquial expressions, and possibly for the reconstructive shifting of certain arguments (when fronted) or adjuncts.

**Generation from predicate-argument structures**

One approach to paraphrasing which has particular appeal to speech reconstruction is the idea of generating paraphrases directly from an underlying predicate-argument (shallow semantic) structure, which may in the future be extractable from spontaneous speech text using labels like those described in Section 4.1.2.

While Kozlowski et al. (2003) does not fully implement this concept, it develops an operational prototype. Lexico-grammatic resources (D-Tree Substitution Grammars, or DSG) map elementary semantic structures (predicate-argument set) to sets of syntactic realizations. Given some unrealized input, the appropriate DSG matching the given predicate and satisfying restrictions is found. Arguments and then modifiers are recursively realized from the previous mappings until the entire semantic expression has been fleshed out.

### 5.2.2 Reconstruction by summarization

Summarization on the document and multi-document levels has many similarities to paraphrasing, especially sentence compression methods. The goal is typically to abbreviate
CHAPTER 5. RECONSTRUCTION AND RELATED TEXT TRANSFORMATIONS

Language formalization:

V: BUT that’s not the point i WANTED to draw attention to
O: HOWEVER that IS not the point i WANT to draw attention to

Paraphrasing:

V: there are now 83 lisbon directives which the european parliament has passed
O: the european parliament has passed 83 lisbon directives

Sentence compression/extractive summarization (multi-line):

V: + president that’s a question for you president thank you uh president thank you
O: mister president - - - - - - - - - - - - - - - - - - - - - - - - - -

V: very much may i first of all just say that in light of the constitutional referendums...
O: - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - in light of the constitutional referendums...

Combination disfluency correction, formalization, and extractive summarization (multi-line)

V: AND [THE]+ the kind generosity of my friend DOWN THERE in the socialist group +
O: THEREFORE - the kind generosity of my friend - - in the socialist group in

V: reducing it to half AN hour is + + not HALVING + + + THIRDING IT
O: reducing it to half AND hour is cutting it not TO HALF but to one THIRD -

V: IF THAT’S A WORD + + + +
O: - - - - - - - of what it should be

Figure 5.4: Extractive summarization: Comparing verbatim (V) and official (O) transcripts from the European Parliamentary Proceedings
a long document into a shorter document with minimal content loss, or to combine many similar documents into a single document containing the relevant points from each. There are generally three types of document summarization systems:

- **Extractive summarizers** such as Barzilay and Elhadad (1997) drop non-important syntactic and discourse constituents (sentence compression) – highly dependent on accurate parsing of text.

- **Headline generators**, not relevant to this discussion.

- **Sentence compression systems**, such as those discussed in Section 5.2.1, which are applied to each sentence in a document to produce a shorter version of that document.

The work described in Daumé III and Marcu (2002) attempts to combine elements of the above three approaches. The syntactic structure of each sentence is derived and basic discourse structure is inferred. A hierarchical model of text production is used to drop inessential syntactic and discourse constituents, generating “coherent, grammatical document compressions of arbitrary length”.

To a certain extent, extractive summarization approaches could have use in conversation-level reconstruction, such as speech reconstruction of meetings or public lectures. In addition to backchannel acknowledgements (which serve only to affirm attention rather than to contribute content), conversations and human speech in general contains a level of redundancy not present in written text. Extractive summarization could potentially help to target inessential utterances within a conversation, while features like those in
CHAPTER 5. RECONSTRUCTION AND RELATED TEXT TRANSFORMATIONS

Wrede and Shriberg (2003) (which uses acoustic features like F0 and energy to target “hot spots”, or points of heated discussion, in meeting corpora). could identify highly relevant segments.

One genre of speech data where extractive summarization methods could be of use for speech reconstruction is for the automatic cleaning of political proceedings. In corpora like the European Parliamentary Proceedings (Mostefa et al., 2006), verbatim transcripts are available and correspond to the official, cleaned versions of the proceedings. While this might appear to be ideal parallel text for reconstruction training, we find that speeches are often prepared and the politicians speaking may be considered professional speakers; altogether, such speakers made very few errors. More commonly found in the alignment between the verbatim and official transcripts is “formalization” of the text, or selectively deleting or replacing words, phrases, or sentences which are not immediately relevant to the official business of the governing body. Examples of these alignments can be seen in Figure 5.4.

Though summarization approaches do have potential relevance for some aspects of spontaneous speech cleaning, these methods do not address the dominant issue of reconstruction in most speech data: string transformation to correct speaker errors.

5.2.3 Reconstruction by machine translation techniques

Spontaneous speech reconstruction can be viewed as the translation from an errorful spontaneous “language” to clean English (or appropriate language). The most common ap-
proach for language translation, or statistical machine translation (SMT), is a noisy channel model.

Noisy channel model (described in Equation 5.3) approaches to language engineering methods typically seek the fluent but unknown “source” string most likely to have produced the observed string. It is a common approach in SMT literature (Brown et al., 1993), where for the foreign string $F$ and its English translation $\hat{E} = \arg\max_{E} P(E)P(F|E)$, the probability $P(E)$ is determined by a language model and $P(F|E)$ is a separate translation model from a potential English string back to the observed foreign words. Since reconstruction can be viewed as translating from poorly constructed English (“foreign”) to fluent and grammatical English, and as language models for grammatical speech are plentiful, applying the approach to speech reconstruction is well-motivated here.

A language model (LM) estimates the probability of a word sequence. Typically it is predicted as the product of probabilities of each word given the words seen before it (as seen from 5.4 to 5.5 below). A common approximation for this is a n-gram language model (Equation 5.6). An n-gram LM approximates the context of a given word by conditioning only on the last $n$ words $W_{i-n+1:i-1}$.

$$P(W) = P(W_1^N)$$

$$= \prod_{i=1}^{N} P(W_i|W_1^{i-1})$$

$$= \prod_{i=1}^{N} P(W_i|W_{i-n+1}^{i-1})$$

Additional back-off or interpolation is generally added to this string probability model for
Table 5.4: BLEU score (Eqn. 3.6) and Exact SU Match accuracy (Eqn. 3.11) for Moses SMT-based reconstruction output compared to original and inter-reference baselines.

<table>
<thead>
<tr>
<th>Data</th>
<th>Test data</th>
<th>BLEU</th>
<th>SU Match</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annotated SSR only</td>
<td>Original:References</td>
<td>62.3</td>
<td>10.53%</td>
</tr>
<tr>
<td></td>
<td>Moses SMT output:References</td>
<td><strong>71.3</strong></td>
<td><strong>18.73%</strong></td>
</tr>
<tr>
<td></td>
<td>Inter-reference (Table 3.2)</td>
<td>88.6</td>
<td>62.85%</td>
</tr>
</tbody>
</table>

the purpose of smoothing probabilities to compensate for producible data unseen in training text (see Chen and Goodman (1998) for an overview of common approaches for smoothing language models).

Typically in SMT the transformation from source words to target words $P(F|E)$ must be learned via automatically obtained word-to-word alignments. We might learn, for example, that “water” in English often corresponds to “agua” in Spanish, with alignments crossing between “cold water” and “agua fría”. For the task of speech reconstruction, it is advantageous that only rarely does lexical form alter between the original string and the reconstruction (i.e. our source and target “languages” have a shared vocabulary and simple translation lexicon). Additionally, we have the advantage of manually produced (versus automatically derived) word-to-word alignments during the annotation process, which could prove beneficial during the training of these models.

In SMT, these transformations typically come in the form of word-level or phrase-level substitution, with an additional word reordering model used less frequently. Unlike the speech reconstruction task, many-to-none alignments are rare.

To understand the potential of SMT for reconstruction methods, we use the high-
CHAPTER 5. RECONSTRUCTION AND RELATED TEXT TRANSFORMATIONS

performing and open-source Moses toolkit (Koehn et al., 2007). Moses takes as input a language model \( P(E) \) and a translation model \( P(F|E) \) (word alignment probabilities plus phrase translation probability tables). The language mode \( P(E) \) used is an interpolation of two language models (optimized on heldout data) – one built from English subtitles from the Broadcast News (BN) corpora (Graff et al., 1997) and the second built from the SWBD (Godfrey et al., 1992) CTS corpus with annotated \text{EDITED} nodes removed (\( \lambda_{BN} = 0.797 \) and \( \lambda_{SWBD} = 0.203 \)). Word alignment and translation probabilities \( P(F|E) \) are extracted from the occurrence frequencies in the SSR annotation alignments. Results are shown in Table 5.4 and typical output can be seen in Figure 5.5. Note that compared to results in Tables 3.1 and 3.2, the BLEU reported here is only on the SSR test subcorpus, not the entire SSR corpus. Like that data, however, the training and testing for this experiment is only conducted on the manually annotated (prefiltered) portion of the corpus (and not the unaannotated “clean sentences”) with two references per utterance.

Altogether, while some gain in BLEU was achieved over BLEU when making no changes at all, the low Exact SU Match accuracy of 18.73\% demonstrates that many errors are left unaddressed or mis-corrected by the SMT method. The rules learned are predominantly comprised of deleting filled pauses and discourse markers, and correcting common repetitions like “i don’t i don’t” and “and and”. Examples like #4 and 5 show that a weakness of an SMT reconstruction approach is its strong dependence on the templates created from aligned substrings in the training data. The transformational rules learned are unable to generalize to repeated patterns. In example #5, it appears that the rule “[it it’s it ‘s” → “it ‘s]
CHAPTER 5. RECONSTRUCTION AND RELATED TEXT TRANSFORMATIONS

Good reconstructions:

1-ORIG: I DO N’T i do n’t know if you can do that
   SMT : * ** *** i do n’t know if you can do that

2-ORIG: i said first of all i GOT to find a sitter
   SMT : i said first of all i HAVE to find a sitter

3-ORIG: and you ’re not missing NOTHING because i WAS LIKE i have n’t been out in a while
   SMT : and you ’re not missing ANYTHING because i *** SAID i have n’t been out in a while

Incomplete corrections:

4-ORIG: AND he ‘s AND he ‘s older than i am a little you know about nine years older
   SMT : *** he ‘s *** he ‘s older than i am a little you know about nine years older

5-ORIG: IT IT ‘S it ‘s it ‘s really hard to say what ‘s going to happen in these shows
   SMT : ** ** ** it ‘s it ‘s really hard to say what ‘s going to happen in these shows

Transformations altering meaning:

6-ORIG: UH but A- all the rest of her family i think they would eat that butcher shop meat
   SMT : ** but NOT all the rest of her family i think they would eat that butcher shop meat

7-ORIG: UM UH having listened to YOU actually having S- that sort of triggered that
   SMT : ** AND having listened to *** actually having THERE that sort of triggered that

8-ORIG: but UH even though they ’re reality YOU KNOW that it ’s not real
   SMT : but ** even though they ’re reality *** **** that it ’s not real

Figure 5.5: Reconstruction by machine translation: example output when an SMT sys-

tem (Moses) is trained on aligned verbatim-reconstructed data and tested on the SSR test
corpus. Evaluation of overall results is shown in Table 5.4.
was seen in data, but the out-of-the-box system cannot generalize this transformation to
the test string “it it's it's it's”, even by applying templates in series (e.g. [“it it's it's it's” → “it it's
it's” → “it's”]). Another problem is that translation templates in the Moses system cannot be
aligned to nothing. Thus, for example, the rule [“um” → ϵ] can never be learned and the
system must instead try to learn [“um w”→ w] for every unique token w which follows “um”
in the corpus. Overall, Moses attempted to transform little beyond a few speaker errors
like discourse markers and commonly repeated words, but due to the template restrictions
mentioned, was not able to clean even these consistently.

Some of these problems may be due to a comparative lack of parallel training data
compared to what is used in most SMT tasks – typically 100k sentence pairs as opposed
to the 17,000 pairs used in this task – though as mentioned we have a much larger body
of well-constructed English resources for language modeling and manually aligned words
which could reduce the impact of sparsity on the translation model. Even so, spontaneous
speech reconstruction by SMT will likely require bootstrapping methods to supplement our
training, and diligent checks to ensure that models avoid overfitting and handle sparsity.

Spontaneous speech reconstruction varies from statistical machine translation in many
ways, including the following:

- Reconstructive changes tend to emphasize deletions, which in standard SMT can
  only be generated by many-to-one alignment template mappings.
- Reconstruction involves far fewer substitutions than SMT (most lexical items take
  the same form in both source and target), and accordingly the default “translation”
CHAPTER 5. RECONSTRUCTION AND RELATED TEXT TRANSFORMATIONS

for any given word is itself.

- Reordering rules are less consistent than those in SMT (which can take advantage of language-specific rules like “[adjective] [noun]” → “[noun] [adjective]” when translating between English and Romance languages like Spanish).

A noisy-channel transformation model for speech reconstruction, such as one that is SMT-based, will need to alter its parametrization to handle these differences. Initially however, effort would be more productively focused on developing a reconstruction approach more directly relevant to the error-correction task.

5.3 Automatic speech recognition (ASR) and speech reconstruction

An ideal speech reconstruction system would perform directly on ASR output, alleviating the need for manual transcripts. In Chapter 1, the scope of this work was defined to focus on reconstructing speech transcripts rather than ASR output for now (and Table 5.1 results help to justify why). However, reconstruction on ASR output in the future would be desirable both for speaker error reduction and possibly direct ASR error correction. To anticipate potential gains in ASR error correction through reconstruction, we further investigate the type and frequency of errors made by a state-of-the-art speech recognition system decoding spontaneous speech.
In this section we attempt to quantify the potential ASR error correction gain if reconstruction methods could successfully be applied to speech recognition lattices.

Here we consider ASR output using the combined IBM-SRI system on the development partition $\text{DEv2}$ of the Fisher conversational telephone speech corpus. Due to inconsistent sentence boundary markings between transcripts, ASR output, and simple disfluency analysis, we first had to resegment the Fisher data (see Section 2.3.1). After finding the Levenshtein alignment between the IBM-SRI ASR output and the reference transcript, we studied the types of errors made and the apparent potential for fixing these problems. A summary of the experiments conducted and the results found is in Table 5.5.

To measure the potential of speech reconstruction of lattices to fix speech errors, we

<table>
<thead>
<tr>
<th>Comparison</th>
<th>WER</th>
<th># ref wds</th>
<th># ins</th>
<th># del</th>
<th># subs</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall:</td>
<td>11.7%</td>
<td>37425</td>
<td>636</td>
<td>1329</td>
<td>2475</td>
<td>-</td>
</tr>
<tr>
<td>Found in lattice</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Function words</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Content words</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OOV (content) words</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Function in lattice</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Content in lattice</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Ref/ASR markup:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edit/edit region</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edit/miss region</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non-edits/edit</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non-edits/non-edit</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Function edit words</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Content edit words</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.5: ASR errors: content, function, and out-of-vocabulary (OOV) word analysis of ASR lattices on Fisher dev04 set. (*Baseline WER for edit experiment was 13.0 due to normalization issues.*)
searched through those lattices for each missed reference word to see which of those words could be found in the lattice of hypothesized output. We found that a large majority of deletions but just over half of substitutions are hypothesized in the original lattice. Overall, we approximate that the lattice oracle (best possible) WER could be as low as 5.3%, a strong motivation to continue rescoring work through reconstruction or other means. Of course, we also note that this oracle path may not exist in all cases.

While 68% of the reference words are identified as function words (a.k.a. they appeared on a standard 76-word stop-list), function words accounted for 83% of deletions but only 60% of substitutions. 27% of missed content words are out of vocabulary (OOV) and thus unrecoverable through reranking alone. Moreover, only 43% of missed content words are found in the lattices, a negative result as we expect that function words will be far more easily induced from text via rules than missing or incorrect content word counterparts (see Sections 3.3.1-3.3.3 for examples of this).

Finally, we analyzed what difference, if any, simple disfluency cleaning would have on our WER, and whether more could be learned by studying these edit regions. We carried out simple cleaning on both reference and hypothesis text (using reference and automatically obtained edit markup, respectively, to simulate what we’d expect from a real word test). We found that, within reference edit boundaries on the reference text, there is a 25.4% WER, far higher than the 11.8% overall WER. However, these errors also had a major impact on the accuracy of the automatic edit detector. Regions where the automatic system missed the edit had a 32.0% WER, and regions where the automatic system gave false edits
CHAPTER 5. RECONSTRUCTION AND RELATED TEXT TRANSFORMATIONS

(though small) had a very high 50.3% WER. This is further evidence that edit detection on ASR output is currently very weak, and brings up the question of whether the two data sets (transcripts and ASR output) should use the same edit detection techniques and if the ASR-specific system should perhaps be less dependent on lexical items. While compelling, these results may be impacted by alignment issues and should be further verified in the future.

5.4 Summary and conclusions

In this chapter, a preamble to automatic approaches for speech reconstruction, we approached analysis of related applied methods in three distinct ways.

In Section 5.1, we described previous approaches for automatic detection of filler words and reparanda, highlighting failures of systems detecting directly on automatic speech recognition output, and specifically emphasizing the “JC04” noisy channel tree-adjoining grammar model (Johnson and Charniak, 2004).

Section 5.2 looked ahead to what will be required for full reconstruction including word reorderings, substitutions, and other complex string transformations. We reviewed standard approaches to paraphrasing, summarization, and machine translation, and used an out-of-box statistical machine translation, trained on parallel SSR corpus data to attempt speech reconstruction.

The final section of the chapter, Section 5.3, looked not at error detection or string transformation approaches, but instead surveyed the output of a state-of-the-art automatic
CHAPTER 5. RECONSTRUCTION AND RELATED TEXT TRANSFORMATIONS

speech reconstruction system on conversational telephone speech data. The errors made were quantified and separated into content vs. function words, in- vs. out-of-vocabulary, and errors occurring within and outside of reparandum regions. It was found that only 43% of missed content words, considered to be those most likely to be repaired during successful reconstruction, appear in the lattices of a top automatic speech recognition system lattice, indicating that to impact ASR itself, reconstruction may need to be considered at an earlier stage of recognition, or may only be useful at an improved state of speech recognition in general.

The approaches discussion will motivate us as we pursue our own automatic approaches for reconstruction and correction of errorful spontaneous speech.
Chapter 6

Automatic identification and repair of simple errors

This chapter presents a conditional random field-based approach for identifying speaker-produced disfluencies (i.e. where and if they occur) in spontaneous speech transcripts (Fitzgerald et al., 2009). We emphasize false start regions, which are often missed in current disfluency identification approaches (Section 5.1.1) as they lack lexical or structural similarity to the speech immediately following. We find that combining lexical, syntactic, and language model-related features with the output of a state-of-the-art disfluency identification system improves overall word-level identification of these and other errors. Improvements are reinforced under a stricter evaluation metric requiring exact matches between cleaned sentences annotator-produced reconstructions, and altogether show promise for general reconstruction efforts.
CHAPTER 6. AUTOMATIC IDENTIFICATION AND REPAIR OF SIMPLE ERRORS

he that’s g- uh that’s a relief

NC RC RC RC FL - - - -

Figure 6.1: Refined word-level error class labels, where – denotes a non-error, FL denotes a filler, E generally denotes reparanda, and RC and NC indicate rough copy and non-copy speaker errors, respectively.

While full speech reconstruction would likely require a range of string transformations and potentially deep syntactic and semantic analysis of the errorful text, in this work we will first attempt to resolve less complex errors, corrected by deletion alone, in a given manually-transcribed utterance.

We build on efforts from Johnson et al. (2004), aiming to improve overall recall – especially of false start or non-copy errors – while concurrently maintaining or improving precision.

6.1 Conditional random fields

Figure 6.2: Illustration of a conditional random field. For this work, x represents observable inputs for each word as described in Section 6.3 and y represents the error class of each word (Section 6.4).

Conditional random fields (Lafferty et al., 2001), or CRFs, are undirected graphical models whose prediction of a sequence of hidden variables Y is globally conditioned on
a given observation sequence $X$, as shown in Figure 6.2. Each observed state $x_i \in X$ is composed of the corresponding word $w_i$ and a set of additional features $F_i$, detailed in Section 6.3.

The conditional probability of this model can be represented as

$$p_\Lambda(Y|X) = \frac{1}{Z_\Lambda(X)} \exp\left(\sum_k \lambda_k F_k(X, Y)\right)$$  \hspace{1cm} (6.1)

where $Z_\Lambda(X)$ is a global normalization factor and $\Lambda = (\lambda_1 \ldots \lambda_K)$ are model parameters related to each feature function $F_k(X, Y)$.

CRFs have been widely applied to tasks in natural language processing, especially those involving tagging words with labels such as part-of-speech tagging and shallow parsing (Sha and Pereira, 2003), as well as sentence boundary detection (Liu et al., 2005, 2004). These models have the advantage that they model sequential context (like Hidden Markov Models (HMMs)) but are discriminative rather than generative and have a less restricted feature set. Additionally, as compared to HMMs, CRFs offer conditional (versus joint) likelihood, and directly maximizes posterior label probabilities $P(E|O)$.

We use the GRMM package (Sutton, 2006) to implement our CRF models, each using a zero-mean Gaussian prior to reduce over-fitting our model. No feature reduction is employed, except where indicated.
6.2 Experimental data

We conducted our experiments on the full Spontaneous Speech Reconstruction (SSR) corpus, a medium-sized set of disfluency annotations atop Fisher conversational telephone speech (CTS) data (Cieri et al., 2004) described in Chapter 2.

As reconstructions are sometimes non-deterministic (illustrated in examples from Section 1.2.1), the SSR provides two manual reconstructions for each utterance in the data. We use these dual annotations to learn complementary approaches in training and to allow for more accurate evaluation.

We recall that the SSR corpus does not explicitly label all reparandum-like regions, as defined in Section 1.2.1, but only those which annotators selected to delete (for example, type 5 errors). Thus, for these experiments we must implicitly attempt to replicate annotator decisions regarding whether or not to delete reparandum regions when labeling them as such. Fortunately, we expect this to have a negligible effect here as we will emphasize utterances which do not require more complex reconstructions in the work described in this chapter.

The output of the JC04 model ((Johnson and Charniak, 2004), described in Section 5.1.1), is included as a feature and used as an approximate baseline in the following experiments. The training of the TAG model within this system requires a very specific data format, so this system is trained not with SSR but with Switchboard (SWBD) (Godfrey et al., 1992) data as described in Johnson and Charniak (2004). Key differences in these corpora, besides the form of their annotations, include:
CHAPTER 6. AUTOMATIC IDENTIFICATION AND REPAIR OF SIMPLE ERRORS

- SSR aims to correct speech output, while SWBD edit annotation aims to identify reparandum structures specifically. Thus, as mentioned, SSR only marks those reparanda which annotators believe must be deleted to generate a grammatical and content-preserving reconstruction.

- SSR considers some phenomena such as leading conjunctions (“and i did” → “i did”) to be fillers, while SWBD does not.

- SSR includes more complex error identification and correction, though these effects should be negligible in the experimental setup presented herein.

While we hope to adapt the trained JC04 model to SSR data in the future, for now these differences in task, evaluation, and training data will prevent direct comparison between JC04 and newly produced results.

6.3 Feature functions

We aim to train our CRF model with sets of features with orthogonal analyses of the errorful text, integrating knowledge from multiple sources. While we anticipate that repetitions and other rough copies will be identified primarily by lexical and local context features, this will not necessarily help for false starts with little or no lexical overlap between reparandum and repair. To catch these errors, we add both language model features (trained with the SRILM toolkit (Stolcke, 2002) on SWBD data with EDITED reparandum nodes removed), and syntactic features to our model. We also included the output of the
JC04 system – which had generally high precision on the SSR data (see Section 5.1.1) – in the hopes of building on these results.

Altogether, the following features $F$ were extracted for each observation $x_i$.

- **Lexical features**, including
  - the lexical item and part-of-speech (POS) for tokens $t_i$ and $t_{i+1}$,
  - distance from previous token to the next matching word/POS,
  - whether previous token is partial word and the distance to the next word with same start, and
  - the token’s (normalized) position within the sentence.

  The motivation for this last feature stems from the fact that many simple mistakes have been found to occur at or near the beginning of a given SU. As found in Bock (1982), “the syntactic structure of utterances appears to be sensitive to the accessibility of lexical information, with phrases containing more accessible information occurring earlier in sentences.”

- **JC04-edit**: whether previous, next, or current word is identified by the JC04 system as an edit and/or a filler (fillers are classified as described in Johnson et al. (2004)).

- **Language model (LM) features**: the unigram log probability of the next word (or POS) token $p(t)$, the token log probability conditioned on its multi-token history $h$
(p(t|h))^1, and the log ratio of the two (log \( \frac{p(t|h)}{p(t)} \)) to serve as an approximation for mutual information between the token and its history, as defined below.

\[
I(t; h) = \sum_{h,t} p(h, t) \log \frac{p(h, t)}{p(h)p(t)} \\
= \sum_{h,t} p(h, t) \left[ \log \frac{p(t|h)}{p(t)} \right] \quad (6.2)
\]

This aims to capture unexpected n-grams produced by the juxtaposition of the reparandum and the repair. The mutual information feature aims to identify when common words are seen in uncommon context (or, alternatively, penalize rare n-grams normalized for rare words).

- **Non-terminal (NT) ancestors**: Given an automatically produced parse of the utterance (using the Charniak (1999) parser trained on Switchboard (SWBD) (Godfrey et al., 1992) CTS data), we determine for each word all NT phrases just completed (if any), all NT phrases about to start to its right (if any), and all NT constituents for which the word is included.

Ferreira and Bailey (2004) and others have found that false starts and repeats tend to end at certain points of phrases, which was also found to be generally true for the annotated data (as described in Section 3.3.5, where the final POS and phrase position for the end of each speaker error was studied).

Note that the syntactic and POS features we used are extracted from the output of an automatic parser. While we do not expect the parser to always be accurate, especially when

---

1In our model, word historys \( h \) encompassed the previous two words (a 3-gram model) and POS history encompassed the previous four POS labels (a 5-gram model)
CHAPTER 6. AUTOMATIC IDENTIFICATION AND REPAIR OF SIMPLE ERRORS

parsing errorful text, we hope that the parser will at least be consistent in the types of structures it assigns to particular error phenomena. We use these features in the hope of taking advantage of that consistency.

6.3.1 Prosodic features

One set of features notably absent from the analysis presented here are those derived from acoustics. As described in Section 1.3.1 and reinforced by related results in Table 5.1, this work emphasizes gains possible using textual information from the transcripts without acoustic analysis, a task which humans are proficient at and which allows us to gauge the information available in transcripts alone.

However, it is obvious that acoustics and prosody (the sound patterns of language, focusing especially on intonation, stress, and syllable timing (Ferreira and Bailey, 2004)) play a major role in human comprehension of errorfully produced spontaneous speech. In studies reported in Ferreira et al. (2004); Fox Tree and Meijer (2000); Aylett (2005), it is found that adding pauses to speech has direct impact on comprehension rates, and Fox Tree and Clark (1997) report that pronouncing the word “the” as THIY instead of THUH often serves as a signal to the listener that the speaker is about to pause or make a correction.

Several studies, including Ferreira and Bailey (2004); Levelt and Cutler (1983), have observed that reparanda are not prosodically marked, but rather it is the repair portion of a disfluency that is spoken with distinctive acoustic features. Thus the human processing
“parser” receives no prosodic warning of a disfluency, and therefore as the reparandum is processed it appears only to be a standard linguistic sequence. This combination of cues is received temporally after the reparandum has concluded, and therefore a certain amount of backtracking, both in human parsers and during automatic reconstruction, must be used to accurately identify and correct these errors.

Hirschberg et al. (1999) find that utterances misrecognized during automatic speech recognition (ASR) are correlated with prosody (specifically f0 excursion, loudness, long prior pause, and longer duration), both between and within speakers. All these are also features of hyper-articulated speech, but the observation remains true even when dominant examples of hyper-articulated speech are excluded.

Some work done where acoustics and prosodic features are incorporated into error detection or handling include the following.

- In Nakatani and Hirschberg (1993), acoustic and prosodic information is specifically targeted as potential cues for instances of self-repair. Though correlations were determined, little improvement in detecting errorful words is achieved.

- Liu et al. (2005); Ostendorf et al. (2004) include as features in their automatic disfluency detection systems binned posteriors (based on decision tree models) of prosodic events such as duration, pitch, and energy.

- In the speech parsing system described in Kahn et al. (2005), prosodic information is incorporated into the *parse selection/reranking* process (not directly into edit detection), along with non-local syntactic information, leading to improved parsing
CHAPTER 6. AUTOMATIC IDENTIFICATION AND REPAIR OF SIMPLE ERRORS

accuracy on accurate transcripts of conversational speech.

The authors provide motivation for using ToBI (prosodic) labels instead of direct acoustic features, saying the following:

These events are predicted from a combination of continuous acoustic correlates, rather than using the acoustic features directly, because the intermediate representation simplifies training with high-level (sparse) structures. Simplify modeling the interdependent set of continuous-valued acoustic cues related to prosody. However, also as in speech recognition, we use posterior probabilities of these events as features rather than making hard decisions about presence vs. absence of a constituent boundary.

6.4 Experimental setup

In these experiments, we attempt to label the following word-boundary classes as annotated in SSR corpus:

- fillers (FL), including filled pauses and discourse markers (\(\sim 5.6\%\) of words)
- rough copy (RC) edit (reparandum incorporates the same or very similar words in roughly the same word order, including repetitions and some revisions) (\(\sim 4.6\%\) of words)
- non-copy (NC) edit (a speaker error where the reparandum has no lexical or structural relationship to the repair region following, as seen in restart fragments and some revisions) (\(\sim 3.2\%\) of words)

Other labels annotated as described in Section 2.6.1 (such as insertions and word reorderings), have been ignored for these error tagging experiments.
CHAPTER 6. AUTOMATIC IDENTIFICATION AND REPAIR OF SIMPLE ERRORS

<table>
<thead>
<tr>
<th>Setup</th>
<th>Train data</th>
<th>Test data</th>
<th>Classes trained by model</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Full train</td>
<td>Full test</td>
<td>FL + RC + NC</td>
</tr>
<tr>
<td>2</td>
<td>Full train</td>
<td>Full test</td>
<td>{RC, NC}, FL+{RC, NC}</td>
</tr>
<tr>
<td>3</td>
<td>Errorful SUs</td>
<td>Errorful SUs</td>
<td>FL + RC + NC</td>
</tr>
<tr>
<td>4</td>
<td>Errorful SUs</td>
<td>Full test</td>
<td>FL + RC + NC</td>
</tr>
<tr>
<td>5</td>
<td>Errorful SUs</td>
<td>Errorful SUs</td>
<td>{RC, NC}, FL+{RC, NC}</td>
</tr>
<tr>
<td>6</td>
<td>Errorful SUs</td>
<td>Full test</td>
<td>{RC, NC}, FL+{RC, NC}</td>
</tr>
</tbody>
</table>

Table 6.1: Overview of experimental setups for word-level error predictions

We approach our training of CRFs in several ways, detailed in Table 6.1. In half of our experiments (#1, 3, and 4), we trained a single model to predict all three annotated classes (as defined at the beginning of Section 6.5), and in the other half (#2, 5, and 6), we trained the model to predict NCs only, NCs and FLs, RCs only, or RCs and FLs (as FLs often serve as interregnum, we predicted that these would be a valuable cue for other edits).

We varied the subcorpus utterances used in training. In some experiments (#1 and 2) we trained with the entire training set\(^2\), including sentences without speaker errors, and in others (#3-6) we trained only on those sentences containing the relevant deletion errors (and no additionally complex errors) to produce a densely errorful training set. Likewise, in some experiments we produced output only for those test sentences which we knew to contain simple errors (#3 and 5). This was meant to emulate the ideal condition where we could perfectly predict which sentences contain errors before identifying where exactly those errors occurred.

\(^2\)Using both annotated SSR reference reconstructions for each utterance
CHAPTER 6. AUTOMATIC IDENTIFICATION AND REPAIR OF SIMPLE ERRORS

The JC04-edit feature was included to help us build on previous efforts for error classification. To confirm that the model is not simply replicating these results and is indeed learning on its own with the other features detailed, we also trained models without this JC04-edit feature. Similarly, we ran models trained without and only with each group of features described in Section 6.3 (e.g. lexical, JC04, language model, and NT bounds features) in order to better estimate the impact of each.

6.5 Word-level evaluation of error point ID

We first evaluate edit detection accuracy on a per-word basis. To evaluate our progress identifying word-level error classes, we calculate precision, recall and $F_1$-scores for each labeled class $c$ in each experimental scenario, as defined in Section 3.1. As usual, these metrics are calculated as ratios of correct, false, and missed predictions. However, to take advantage of the double reconstruction annotations provided in SSR (and more importantly, in recognition of the occasional ambiguities of reconstruction) we modified these calculations as shown below.

$$\text{corr}(c) = \sum_{i:c_{wi} = c} \delta(c_{wi} = c_{g1,i} \text{ or } c_{wi} = c_{g2,i})$$  \hspace{1cm} (6.3)$$

$$\text{false}(c) = \sum_{i:c_{wi} = c} \delta(c_{wi} \neq c_{g1,i} \text{ and } c_{wi} \neq c_{g2,i})$$ \hspace{1cm} (6.4)$$

$$\text{miss}(c) = \sum_{i:c_{g1,i} = c} \delta(c_{wi} \neq c_{g1,i})$$ \hspace{1cm} (6.5)$$

$$P = \frac{\text{corr}(c)}{\text{corr}(c) + \text{false}(c)} \hspace{1cm} R = \frac{\text{corr}(c)}{\text{corr}(c) + \text{miss}(c)}$$ \hspace{1cm} (6.6)$$

151
where \( c_{w_i} \) is the hypothesized class for \( w_i \), \( c_{g_1,i} \) and \( c_{g_2,i} \) are the two reference classes, and \( \delta(\text{expression}) = 1 \) whenever the expression is true, and zero otherwise.

<table>
<thead>
<tr>
<th>Setup</th>
<th>Class labeled</th>
<th>Training</th>
<th>Testing</th>
<th>FL</th>
<th>RC</th>
<th>NC</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1 FP+RC+NC</td>
<td>All train</td>
<td>All test</td>
<td><strong>71.0</strong></td>
<td>80.3</td>
<td>47.4</td>
<td></td>
</tr>
<tr>
<td>#2 NC</td>
<td>All train</td>
<td>All test</td>
<td>-</td>
<td>-</td>
<td><strong>42.5</strong></td>
<td></td>
</tr>
<tr>
<td>#2 NC+FL</td>
<td>All train</td>
<td>All test</td>
<td>70.8</td>
<td>-</td>
<td><strong>47.5</strong></td>
<td></td>
</tr>
<tr>
<td>#2 RC</td>
<td>All train</td>
<td>All test</td>
<td>-</td>
<td><strong>84.2</strong></td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>#2 RC+FL</td>
<td>All train</td>
<td>All test</td>
<td>67.8</td>
<td><strong>84.7</strong></td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

Train and test on errorful SUs

<table>
<thead>
<tr>
<th>Setup</th>
<th>Class labeled</th>
<th>Training</th>
<th>Testing</th>
<th>FL</th>
<th>RC</th>
<th>NC</th>
</tr>
</thead>
<tbody>
<tr>
<td>#3 FP+RC+NC</td>
<td>Errorful only</td>
<td>Errorful only</td>
<td>Errorful only</td>
<td><strong>91.6</strong></td>
<td>84.1</td>
<td>52.2</td>
</tr>
<tr>
<td>#4 NC</td>
<td>Errorful only</td>
<td>All test</td>
<td>44.1</td>
<td>69.3</td>
<td>31.6</td>
<td></td>
</tr>
<tr>
<td>#5 NC+FL</td>
<td>Errorful only</td>
<td>Errorful only</td>
<td>Errorful only</td>
<td>-</td>
<td>-</td>
<td><strong>73.8</strong></td>
</tr>
<tr>
<td>#6 RC</td>
<td>Errorful only</td>
<td>All test</td>
<td>-</td>
<td>-</td>
<td>39.2</td>
<td></td>
</tr>
<tr>
<td>#5 RC+FL</td>
<td>Errorful only</td>
<td>Errorful only</td>
<td>Errorful only</td>
<td>90.7</td>
<td>-</td>
<td>69.8</td>
</tr>
<tr>
<td>#6 RC+FL</td>
<td>Errorful only</td>
<td>All test</td>
<td>50.1</td>
<td>-</td>
<td>38.5</td>
<td></td>
</tr>
</tbody>
</table>

Table 6.2: Word-level error prediction \( F_1 \)-score results, part 2. The first column identifies which data setup was used for each experiment. The highest performing result for each class in the first set of experiments has been highlighted.

Analysis of word-level label evaluation

Experimental results can be seen in Tables 6.2 and 6.3. Table 6.2 shows the impact of training models for individual features and of constraining training data to contain only those utterances known to contain errors. It also demonstrates the potential impact on error classification after prefiltering test data to those SUs with errors. Table 6.3 demonstrates the contribution of each group of features to our CRF models.
CHAPTER 6. AUTOMATIC IDENTIFICATION AND REPAIR OF SIMPLE ERRORS

<table>
<thead>
<tr>
<th>Features</th>
<th>FL</th>
<th>RC</th>
<th>NC</th>
</tr>
</thead>
<tbody>
<tr>
<td>JC04 only</td>
<td>56.6</td>
<td>69.9-81.9</td>
<td>1.6-21.0</td>
</tr>
<tr>
<td>lexical only</td>
<td>56.5</td>
<td>72.7</td>
<td>33.4</td>
</tr>
<tr>
<td>LM only</td>
<td>0.0</td>
<td>15.0</td>
<td>0.0</td>
</tr>
<tr>
<td>NT bounds only</td>
<td>44.1</td>
<td>35.9</td>
<td>11.5</td>
</tr>
<tr>
<td>All but JC04</td>
<td>58.5</td>
<td>79.3</td>
<td>33.1</td>
</tr>
<tr>
<td>All but lexical</td>
<td>66.9</td>
<td>76.0</td>
<td>19.6</td>
</tr>
<tr>
<td>All but LM</td>
<td>67.9</td>
<td>83.1</td>
<td>41.0</td>
</tr>
<tr>
<td>All but NT bounds</td>
<td>61.8</td>
<td>79.4</td>
<td>33.6</td>
</tr>
<tr>
<td>All</td>
<td>71.0</td>
<td>80.3</td>
<td>47.4</td>
</tr>
</tbody>
</table>

Table 6.3: **Word-level error prediction F$_1$-score results: Feature variation**. All models were trained with experimental setup #1 and with the set of features identified.

Our results demonstrate the impact of varying our training data and the number of label classes trained for. We see in Table 6.2 from setup #5 experiments that training and testing on error-containing utterances led to a dramatic improvement in F$_1$-score. On the other hand, our results for experiments using setup #6 (where training data was filtered to contain errorful data but test data was fully preserved) are consistently worse than those of either setup #2 (where both train and test data was untouched) or setup #5 (where both train and test data were prefiltered). The output appears to suffer from sample bias, as the prior of an error occurring in training is much higher than in testing. This demonstrates that a densely errorful training set alone cannot improve our results when testing data conditions do not match training data conditions. However, efforts to identify errorful sentences before determining where errors occur in those sentences may be worthwhile in preventing false positives in error-less utterances.

We next consider the impact of the four feature groups on our prediction results. The
CHAPTER 6. AUTOMATIC IDENTIFICATION AND REPAIR OF SIMPLE ERRORS

CRF model appears competitive even without the advantage of building on JC04 results, as seen in Table 6.3. Interestingly and encouragingly, the NT bounds features which indicate the linguistic phrase structures beginning and ending at each word according to an automatic parse were also found to be highly contribututive for both fillers and non-copy identification. We believe that further pursuit of syntactic features, especially those which can take advantage of the context-free weakness of statistical parsers like Charniak (1999) will be promising in future research.

It was unexpected that NC classification would be so sensitive to the loss of lexical features while RC labeling was generally resilient to the dropping of any feature group. We hypothesize that for rough copies, the information lost from the removal of the lexical items might have been compensated for by the JC04 features as JC04 performed most strongly on this error type. This should be further investigated in the future.

6.6 Sentence-level evaluation of error point ID

Depending on the downstream task of speech reconstruction, it could be imperative not only to identify many of the errors in a given spoken utterance, but indeed to identify all errors (and only those errors), yielding the exact cleaned sentence that a human annotator might provide.

In these experiments we apply simple cleanup (as defined in Section 1.2.1) to both JC04

---

JC04 results are shown as a range for the reasons given in Section 5.1.1: since JC04 does not on its own predict whether an “edit” is a rough copy or non-copy, it is impossible to calculate precision and thus F\textsubscript{1} score precisely. Instead, here we show the resultant F\textsubscript{1} for the best case and worst case precision range.
CHAPTER 6. AUTOMATIC IDENTIFICATION AND REPAIR OF SIMPLE ERRORS

<table>
<thead>
<tr>
<th>Setup</th>
<th>Classes deleted</th>
<th># SUs</th>
<th># SUs which match gold</th>
<th>% accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>only filled pauses</td>
<td>2288</td>
<td>1800</td>
<td>78.7%</td>
</tr>
<tr>
<td>JC04-1</td>
<td>E+FL</td>
<td>2288</td>
<td>1858</td>
<td>81.2%</td>
</tr>
<tr>
<td>CRF-#1</td>
<td>RC, NC, and FL</td>
<td>2288</td>
<td>1922</td>
<td>84.0%</td>
</tr>
<tr>
<td>CRF-#2</td>
<td>( \bigcup {RC,NC} )</td>
<td>2288</td>
<td>1901</td>
<td>83.1%</td>
</tr>
<tr>
<td>Baseline</td>
<td>only filled pauses</td>
<td>281</td>
<td>5</td>
<td>1.8%</td>
</tr>
<tr>
<td>JC04-2</td>
<td>E+FL</td>
<td>281</td>
<td>126</td>
<td>44.8%</td>
</tr>
<tr>
<td>CRF-#3</td>
<td>RC, NC, and FL</td>
<td>281</td>
<td>156</td>
<td>55.5%</td>
</tr>
<tr>
<td>CRF-#5</td>
<td>( \bigcup {RC,NC} )</td>
<td>281</td>
<td>132</td>
<td>47.0%</td>
</tr>
</tbody>
</table>

Table 6.4: **Word-level error predictions: Exact Sentence Match Results.** JC04-2 was run only on test sentences known to contain some error to match the conditions of Setup #3 and #5 (from Table 6.1). For the baselines, we delete only filled pause filler words like “eh” and “um”.

output and the predicted output for each experimental setup in Table 6.1, deleting words when their right boundary class is a filled pause, rough copy or non-copy.

Taking advantage of the dual annotations for each sentence in the SSR corpus, we can report both double-reference exact match evaluation, as defined in Equation 3.11 in Section 3.1. Thus, we judge that if a hypothesized cleaned sentence exactly matches *either* reference sentence cleaned in the same manner, we count the cleaned utterance as correct and otherwise assign no credit, and we find the average over all sentences judged.

**Analysis of sentence level evaluation**

We see the outcome of this set of experiments in Table 6.4. While the unfiltered test sets JC04-1 and setups #1 and 2 appear to have much higher sentence-level cleanup accuracy than the other experiments (JC04-2, #3, and 5), we recall that this is naturally also due to the fact that the majority of these sentences should not be cleaned at all, besides occasional
CHAPTER 6. AUTOMATIC IDENTIFICATION AND REPAIR OF SIMPLE ERRORS

minor filled pause deletions. Looking specifically on cleanup results for sentences known to contain at least one error, we see, once again, that our system outperforms our baseline JC04 system at this task.

6.7 Discussion

Our first goal in this work was to focus on an area of disfluency detection currently weak in other state-of-the-art speaker error detection systems – false starts – while producing comparable classification on repetition and revision speaker errors. Secondly, we attempted to quantify how far deleting identified edits (both RC and NC) and filled pauses could bring us to full reconstruction of these sentences.

We’ve shown in Section 6.5 that by training and testing on data prefiltered to include only utterances with errors, we can dramatically improve our results, not only by improving identification of errors but presumably by reducing the risk of falsely predicting errors. In Chapter 7, we further investigate to understand how well we can automatically identify errorful spoken utterances in a corpus.
Chapter 7

Integrating sentence- and word-level error identification

In the experiments conducted and described in Chapter 6, the potential benefit of conducting word-level reconstruction of simple errors only on those sentences known to have errors was quantified. In the work described in this chapter, we explore new approaches for automatically identifying speaker construction errors on the utterance level, and learn the impact that this initial step has on word- and sentence-level accuracy, compared to that shown in Sections 6.5 and 6.6.
CHAPTER 7. INTEGRATING ERROR IDENTIFICATION METHODS

7.1 Identifying poor constructions

It is to our advantage to automatically identify (prior to reconstruction) poorly con-
structed sentences, defined as being ungrammatical, incomplete, or missing necessary sen-
tence boundaries. In addition to optimizing resources before annotating a densely errorful
training set, we see in Sections 6.5 and 6.6 that accurately extracting ill-formed sentences
prior to sub-sentential error correction helps to minimize the risk of information loss posed
by unnecessarily and incorrectly reconstructing well-formed text.

To evaluate the efforts described, we manually labeled each SU $s$ in the SSR test set $S$
(including those not annotated in Chapter 2) as well-formed or poorly-formed, forming the
set of poorly constructed SUs $P \subset S$, $|P| = 531$ and $|S| = 2288$ utterances.

7.1.1 Maximum entropy prediction models

In this section, we build on the preliminary sentence-level error identification described
in Section 2.4. We expand our identification features and combine those features into a
single framework using a maximum entropy model (implemented with the MEGA Model
toolkit (Daumé III, 2004)).

7.1.2 SU-level error features

Six feature types are presented in this section.

- Features #1 and #2 were included in a similar though less exhaustive effort in Section
CHAPTER 7. INTEGRATING ERROR IDENTIFICATION METHODS

2.4 and are explained in further detail below.

- Feature types #3 and #4 both extract features from automatic parses produced by the given sentence. We anticipate that these parses will contain errors given the known construction errors the parser will encounter, and also acknowledge that the usefulness of these features may be parser-specific. However, as we believe that a state-of-the-art parser will treat similar construction errors consistently if not accurately, we believe there will be predictive power in these features.

- Feature type #5 investigates the relationship between an SU’s error condition and the number of words it contains.

- Feature type #6 serves to discount the probability of assigning a backchannel acknowledgement SU as an error instance.

**Feature #1 (JC04): Consider only sentences with JC04 detected edit regions**

This approach, described also in Section 2.4, takes advantage of the JC04 disfluency detection approach (Johnson and Charniak, 2004) defined in Section 5.1.1. We run the detection system on text, and consider any sentence with at least one reparandum to have a “poor” indicator. As speakers repairing their speech once are often under a higher cognitive load and thus more likely to make more serious speech errors (Bard et al., 2001), this seems to be a reasonable first approach for finding deeper problems.

The feature, used alone, performed well, with 97.6% of sentences containing automatically identified reparandum words appearing in the gold standard list of poor constructions.
CHAPTER 7. INTEGRATING ERROR IDENTIFICATION METHODS

(precision), and 67.6% of those pre-identified poor constructions being caught (recall). F$_1$ score is 79.9.

**Feature #2 (HPSG): Using deep linguistic parsers to confirm well-formedness**

Statistical context-free parsers such as Charniak (1999) are highly robust and due to smoothing can assign non-zero probability syntactic structure even for text and part-of-speech sequences never seen in training. However, sometimes it is better to get no output than highly errorful output. While hand-built, rule-based parsers can produce extremely accurate and context-sensitive syntactic structures, they are also brittle and do not adapt well to never before seen input. For this task we used that inflexibility to our advantage.

**Head-driven Phrase Structure Grammar** (HPSG) is a deep-syntax phrase structure grammar which produces rich, non-context-free syntactic analyses of input sentences based on a collection of carefully constructed rules and lexical item structures (Pollard and Sag, 1994; Wahlster, 2000). Each utterance is deep parsed using the PET parser produced by the inter-institutional DELPH-IN group$^1$. The manually compiled English Resource Grammar (ERG) rules have previously been extended for the Verbmobil (Wahlster, 2000) project to allow for the parsing of basic conversational elements such as SUs with no verb or basic backchannel acknowledgements like “last thursday” or “sure”, but still produces strict HPSG parses based on these rules.

We use the binary result of whether or not each sentence is parsable by the HPSG

$^1$The DEep Linguistic Processing with HPSG INitiative (see http://www.delph-in.net/)
CHAPTER 7. INTEGRATING ERROR IDENTIFICATION METHODS

English grammar (Flickinger, 2002) as binary indicator functions in our models.

There has been some work on producing partial parses for utterances for which a full HPSG analysis is not deemed possible by the grammar (Zhang et al., 2007). This work has shown early promise for identifying coherent substrings within errorful SUs given subjective analysis; as this technology improves, HPSG may offer informative subsentential features for word-level error analysis such as that described in Chapter 6.

*Feature #3 (Rules): Unseen phrase rule expansions*

In Section 4.4, we defined context-free grammars and phrase rule expansions. Phrase-based parses are composed of a recursive sequence of non-terminal (NT) rule expansions, such as those detailed for the example parse shown in Figure 7.1. These rules are learned from training data such as the LDC Switchboard treebank, where telephone conversation transcripts were manually parsed. In statistical parsers such as Charniak (1999), new structures are generated based on the relative frequencies of such rules in the training treebank, conditioned on the terminal words and some local context, and the most probable parse (roughly the joint probability of its rule expansions) is selected.

Because parsers are often required to produce output for words and contexts never seen in the training corpus, smoothing is required. Charniak (1999) accomplishes this in part through a *Markov grammar* which works top-down, expanding rules to the left and right of an expansion “head” $M$ of a given rule. The non-terminal (NT) $M$ is first predicted from the parent $P$, then – in order – $L_1$ through $L_m$ (stopping symbol ‘#’) and $R_1$ through $R_n$
(again ‘#’), as shown in Equation 7.1.

\[
\text{parent } P \rightarrow \#L_m \ldots L_1 M R_1 \ldots R_n \#
\]  \hspace{1cm} (7.1)

In this manner, it is possible to produce rules never before seen in the training treebank. While this may happen even for grammatical sentences with rare elements, our fourth SU-level error prediction feature indicates whether the automatic parse for a given SU includes an expansion never seen in the training treebank (in this case, rules seen in the Switchboard treebank after EDITED nodes have been removed).

**Feature #4 (C-comm): Unseen rule c-commanding NTs**

In X’ theory (Chomsky, 1970), lexical categories such as nouns and verbs are often modified by a specifier (such as the DT “a” modifying the NN “lot” in the NP phrase in Figure 7.1 or an auxiliary verb for a verb in a verb phrase (VBZ for VP) and a complement (such as the object of a verb NP for VBG in the phrase VP). In each of these cases, one non-terminal (NT) tree node \( A \) has the following relationship with a second NT \( P \):

- Neither node \( A \) dominates \( P \) nor node \( P \) dominates \( A \), which is to say that neither is directly above the other in the parse tree, and
- Node \( A \) immediately precedes \( P \) in the tree (precedence is represented graphically in left-to-right order in the tree).

Given these relationships, we say that \( A \) locally c-commands \( P \) and its descendants. We
Figure 7.1: The automatically generated parse (a) for an errorful sentence-like unit (SU), with accompanying rule expansions (b) and local c-commands (c). Non-terminal indices such as NP₂ are for reader clarification only and are not considered in the feature extraction process.
further extend this definition to say that, if node \( A \) is the only child of node \( \hat{A} \) (a unary expansion) and \( \hat{A} \) locally c-commands \( P \), then \( A \) locally c-commands \( P \) (so both \([S_{BAR} \rightarrow S]\) and \([S \rightarrow N_{P2} V_{P2}]\) are c-commanded by \( V_{BP} \)). See Figure 7.1 for other examples of non-terminal nodes in c-commanding relationships, and the phrase expansion rule they c-command.

The c-command relationship is fundamental in syntactic theory, and has use in predicting the scope of pronoun antecedents, etc. In this case, however, we use it to describe two nodes which are in a specifier-category or a category-complement relationship. This is valuable to us because it takes advantage of a weakness of statistical parsers: the context used to condition the probability of a given rule expansion generally does not reach beyond dominance relationships, and thus parsers rarely penalize for the juxtaposition of \( A \) c-commanding \( P \) and its children as long as they have previously seen NT type \( A \) preceding NT type \( P \). Thus, we can use the children of a parent node \( P \) as a way to enrich a NT type \( P \) and make it more informative.

For example, in Figure 7.1, the rule \([S \rightarrow N_{P2} V_{P2}]\) is routinely seen in the manual parses of the SWBD treebank, as is \([V_{P1} \rightarrow V_{BP} S_{BAR}]\). However, it is highly unusual for \( V_{BP} \) to immediately precede \( S_{BAR} \) or \( S \) when this rule expands to \( N_{P2} V_{P2} \). So, not only does \( S_{BAR}/S \) complement \( V_{BP} \), but a very specific type of \([S_{BAR}/S \rightarrow N_{P} V_{P}]\) is the complement of \( V_{BP} \). This infrequency serves as an indication of deeper structural errors.

We include in our maximum entropy model a feature indicating whether a given parse includes a c-command relationship not seen in training data.
Feature #5 (Length): Threshold sentences based on length

Empirical observation indicates that long sentences are more likely to contain speaker errors, while very short sentences tend to be backchannel acknowledgments like “yeah” or “I know” which are not considered errorful. Oviatt (1995) quantifies this, determining that the disfluency rate in human-computer dialog increases roughly linearly with the number of words in an utterance. The length-based feature value for each sentence therefore is defined to be the number of word tokens in that sentence.

Feature #6 (Backchannel): Consider backchannel acknowledgements to be non-errors

A backchannel acknowledgement is a short sentence-like unit (SU) which is produced to indicate that the speaker is still paying attention to the other speaker, without requesting attention or adding new content to the dialog. These SUs include “uh-huh”, “sure”, or any combination of backchannel acknowledgements with fillers (ex. “sure uh uh-huh”).

To assign this feature, fifty-two common backchannel acknowledgement tokens are compiled, shown in Figure 7.2. The indicator feature is one (1) if the SU in question is some combination of these backchannel acknowledgements, and zero (0) otherwise.

7.1.3 SU-level error identification results

Maximum entropy classification results using combinations of the features defined in Section 7.1.2 are summarized in Table 7.1.

We first observe the performance of each feature type in isolation in our maximum
CHAPTER 7. INTEGRATING ERROR IDENTIFICATION METHODS

<table>
<thead>
<tr>
<th>Features included</th>
</tr>
</thead>
<tbody>
<tr>
<td>absolutely</td>
</tr>
<tr>
<td>ach</td>
</tr>
<tr>
<td>almost</td>
</tr>
<tr>
<td>alright</td>
</tr>
<tr>
<td>agreed</td>
</tr>
<tr>
<td>all right</td>
</tr>
<tr>
<td>apparently</td>
</tr>
<tr>
<td>ah</td>
</tr>
<tr>
<td>and</td>
</tr>
<tr>
<td>basically</td>
</tr>
<tr>
<td>but</td>
</tr>
<tr>
<td>cool</td>
</tr>
<tr>
<td>correct</td>
</tr>
</tbody>
</table>

Figure 7.2: List of backchannel acknowledgements and fillers, whose appearance or combinations are used as features in SU-level error detection.

<table>
<thead>
<tr>
<th>Setup</th>
<th>Features included</th>
<th>F&lt;sub&gt;1&lt;/sub&gt;-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>a) Individual features</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>√</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>√</td>
</tr>
<tr>
<td>5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>b) All features combined</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>c) All-but-one</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>X</td>
<td>√</td>
</tr>
<tr>
<td>9</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>10</td>
<td>√</td>
<td>X</td>
</tr>
<tr>
<td>11</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>12</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>13</td>
<td>√</td>
<td>√</td>
</tr>
</tbody>
</table>

Table 7.1: Comparison of poor construction identification features, tested on the SSR test corpus.
CHAPTER 7. INTEGRATING ERROR IDENTIFICATION METHODS

entropy framework (Figure 7.1a). The top-performing individual feature is the JC04 edit indicator, which is not surprising as this is the one feature whose existence was designed specifically to predict speaker errors. Following JC04 in individual performance are the HPSG parsability feature, length feature, and unseen c-command rule presence feature. Backchannel acknowledgements had no predictive power on their own, as they were primarily meant to negatively impact the probability of selecting an SU as errorful which we know has a small prior (in our test set, $\frac{P}{S} = \frac{531}{2288} = 0.23$, as described at the beginning of Section 7.1).

Combining all rules together (Figure 7.1b), we note an $F_1$-score gain of 3.4 as compared to the top individual feature JC04. (JC04 has a precision of 97.6, recall of 67.6, and F of 79.9; the combined feature model has a precision of 93.0, a recall of 75.3, and an F of 83.3, so unsurprisingly our gain primarily comes from increased error recall).

In order to understand the contribution of an individual feature, it helps not only to see the prediction results conditioned only on that feature, but the loss in accuracy seen when only that feature is removed from the set. We see in Figure 7.1c that, though the c-command prediction feature was only moderately accurate predicting SU errors alone, it has the second largest impact after JC04 (an F-score loss of 2.1) when removed from the set of features. Length, on the other hand, while moderately powerful as a single indicator, had negligible impact on classification accuracy when removed from the feature set. This indicates that the relationship between errorful sentences and length can be explained away by the other features in our set.
CHAPTER 7. INTEGRATING ERROR IDENTIFICATION METHODS

We also note that the combination of all features excluding JC04 is competitive with JC04 itself. Additional complementary features seem likely to further compete with the JC04 prediction feature.

7.2 Combining efforts

In Chapter 6, we saw that the predictive power of our CRF model could greatly improve, from an F-score of 84.7 to as high as 88.7 for rough copy (RC) errors and from an F-score of 47.5 to as high as 73.8 for non-copy (NC) errors.

Now that we have built a model to predict construction errors on the utterance level, we combine the two approaches to analyze the improvement possible for word-level identification, measured again by precision, recall, and F-score, and for SU-level correction, as measured by the Exact SU Match metric (Equation 3.11).

7.2.1 Word-level evaluation of error point identification, post SU identification

Just as in Section 6.5, we first evaluate edit detection accuracy on those test SUs predicted to be errorful on a per-word basis. To evaluate our progress identifying word-level error classes, we calculate precision, recall and F_1-scores (using the modified definitions from Section 6.5) for each labeled class c.
Table 7.2: Error predictions, post-SU identification: F₁-score results. Automatically identified “SUs for testing” were determined via the maximum entropy classification model described earlier in this chapter, and feature set #7 from Table 7.1. Filler (FL), rough copy error (RC) and non-copy error (NC) results are given in terms of word-level F₁-score. **Bold** numbers indicate the highest performance post-automatic filter for each of the three classes. *Italicized* numbers indicate experiments where no filtering outperformed automatic filtering (for RC errors).

### Analysis of word-level label evaluation, post SU identification

Word level F₁-score results (harmonic mean of precision and recall) for error region identification are shown in Table 7.2.

By first automatically selecting testing data (as described in Section 7.1, with a sentence-level F-score of 83.3), we see some gain in F-score for all three error classes, though much potential improvement remains based on the oracle gain (“Gold errors” testing data) determined in Chapter 6.

We see that, unlike in the experiments where all data was used for testing and training,
CHAPTER 7. INTEGRATING ERROR IDENTIFICATION METHODS

the best NC and RC detection performance given the automatically selected testing data was achieved when training a CRF model to detect each class separately and not in conjunction with filler word detection FL. Once again, training RC and NC models separately instead of in a joint FL+RC+NC model yielded higher accuracy.

We notice also that the F-score for RC identification is lower when automatically filtering the test data. There are two likely causes. The most likely issue is that the automatic SU-error classifier filtered out some SUs with true RC errors which had previously been correctly identified, reducing the overall precision ratio as well as recall (i.e. we no longer receive credit for some easier errors once caught). A second, related possibility is that the errorful SUs identified by the Section 7.1 method had a higher density of errors that the current CRF word-level classification model is unable to identify (i.e. the more difficult errors are now a higher relative percentage of the errors we need to catch). While the former possibility seems more likely, both causes should be investigated in future work.

The F-score gain in NC identification from 42.5 to 54.6 came primarily from a gain in precision (since in the original model, many non-errorful SUs were mistakenly determined to include errors).

Though capturing approximately 55% of the non-copy NC errors (for SUs likely to have errors) is an improvement, this remains a challenging and unsolved task which should be investigated further in the future.
7.2.2 Sentence-level evaluation of error point identification and region deletion, post SU identification

<table>
<thead>
<tr>
<th>Setup</th>
<th>Classed deleted</th>
<th>Testing</th>
<th># SUs (filt/unfilt)</th>
<th># SUs that match ref</th>
<th>% accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline-1</td>
<td>only filled pauses</td>
<td>All data</td>
<td>2288</td>
<td>1800</td>
<td>78.7%</td>
</tr>
<tr>
<td>JC04-1</td>
<td>E+FL</td>
<td>All data</td>
<td>2288</td>
<td>1858</td>
<td>81.2%</td>
</tr>
<tr>
<td>MaxEnt/CRF-1</td>
<td>FL+RC+NC</td>
<td>All data</td>
<td>2288</td>
<td>1922</td>
<td>84.0%</td>
</tr>
<tr>
<td>Baseline-2</td>
<td>only filled pauses</td>
<td>Auto ID’d</td>
<td>430</td>
<td>84</td>
<td>19.5%</td>
</tr>
<tr>
<td>JC04-2</td>
<td>E+FL</td>
<td>Auto ID’d</td>
<td>430</td>
<td>187</td>
<td>43.5%</td>
</tr>
<tr>
<td>MaxEnt/CRF-2</td>
<td>FL+RC+NC</td>
<td>Auto ID’d</td>
<td>430</td>
<td>223</td>
<td>51.9%</td>
</tr>
<tr>
<td>Baseline-3</td>
<td>only filled pauses</td>
<td>Gold errors</td>
<td>281</td>
<td>5</td>
<td>1.8%</td>
</tr>
<tr>
<td>JC04-3</td>
<td>E+FL</td>
<td>Gold errors</td>
<td>281</td>
<td>126</td>
<td>44.8%</td>
</tr>
<tr>
<td>MaxEnt/CRF-3</td>
<td>FL+RC+NC</td>
<td>Gold errors</td>
<td>281</td>
<td>156</td>
<td>55.5%</td>
</tr>
</tbody>
</table>

Table 7.3: **Error predictions, post-SU identification: Exact Sentence Match Results.**
For the baseline, we delete only filled pause filler words like “eh” and “um”. For JC04 output, we deleted any word assigned the class E or FL. Finally, for the MaxEnt/CRF models, we used the jointly trained FL+RC+NC CRF model and deleted all words assigned any of the three classes.

As in Section 6.6, in these experiments we apply simple cleanup to both JC04 output and the predicted output for each of our experimental setups. In this case, “experimental setup” refers to the various feature sets used to train the SU-level error identification model (as seen in Table 7.1).

We again report double-reference exact match evaluation, as defined in Equation 3.11 in Section 3.1 and repeated below.

\[
SU\text{\_match} = \frac{1}{S} \sum_{s \in S} \max_{r \in R} \delta(s, r)
\]  

(7.2)

Thus, we judge that if a hypothesized cleaned sentence exactly matches either reference
sentence cleaned in the same manner, we count the cleaned utterance as correct and otherwise assign no credit, and we find the average over all sentences judged.

Analysis of sentence level evaluation, post SU identification

Results from this second evaluation of rough copy and non-copy error reconstruction can be seen in Table 7.3.

As seen in word-level identification results (Table 7.2), automatically selecting a subset of testing data upon which to apply simple cleanup reconstruction does not perform at the accuracy shown to be possible given an oracle filtering. While measuring improvement is difficult (here, non-filtered data is incomparable to filtered test data results as a majority of these sentences require no major deletions at all), we note once again that our methods (MaxEnt/CRF-x) outperform the baseline of deleting nothing but filled pauses like “eh” and “um”, as well as the state-of-the-art baseline JC04.

7.3 Summary and conclusions

The work in this chapter was an extension of the results in Chapter 6, which showed that automatically determining which utterances contain errors before attempting to identify and delete fillers and reparanda has the potential to increase accuracy significantly.

In Section 7.1, we built a maximum entropy classification model to assign binary error classes to spontaneous speech utterances. Six features – JC04, HPSG, unseen rules, unseen
CHAPTER 7. INTEGRATING ERROR IDENTIFICATION METHODS

c-command relationships, utterance length, and backchannel acknowledgement composition – were considered, and the combined model achieved an improved precision of 93.0, a recall of 75.3, and an $F_1$-score of 83.3.

We then, in Section 7.2, cascaded the sentence-level error identification system output into the Chapter 6 word-level error identification and simple cleanup system. This combination lead to non-copy error identification with an $F_1$-score of 54, up from 47 in the experiments conducted on all data instead of data identified to be errorful, while maintaining accuracy for rough copy errors and increasing filler detection accuracy as well.
Chapter 8

Conclusion

This final chapter summarizes the overall contributions described in this dissertation. Components of these contributions include analysis of speech phenomena, data, production theory extensions, automatic recognition and correction of some speaker-generated errors in spontaneously produced speech.

8.1 Summary

We have motivated and presented a framework for speech reconstruction, a novel problem in natural language processing. Conversational speech text often contains speaker-initiated corrections, disfluencies, self-interruptions, missing, redundant, and misplaced arguments, and other ungrammatical phenomena, making it difficult to process speech in many natural language processing applications. Speech reconstruction, in turn, aims to
correct these ungrammatical events to produce fluent and grammatical text which still preserves the speaker’s intended meaning.

This dissertation begins in Chapter 1 by motivating speech reconstruction work, naming and illustrating fifteen newly defined classes of speaker errors, and defining the scope and terminology for our work.

Chapters 2 through 4 covers motivation for the creation of the Spontaneous Speech Reconstruction (SSR) corpus and corpus annotation process, along with detailed analysis of both surface and underlying characteristics identified in these manually reconstructed utterances from spontaneous conversational telephone speech. A summary of this segment of the work appears in Section 8.1.1 below.

In the latter half of the dissertation – chapters 5 through 7 – we review automatic approaches for string transformations, aimed towards speech reconstruction. We then implement systems of our own for identification and repair of filler and reparandum errors. Findings and results are summarized below in Section 8.1.2.

8.1.1 Comprehensive error analysis

In Chapter 2 we have discussed the creation of the Spontaneous Speech Reconstruction (SSR) corpus. For this set of 21k conversational telephone speech utterances, a third of the utterances were filtered as likely to be errorful and were then manually reconstructed on three levels. The final corpus has been used for analysis of error phenomenon and training and evaluation of automatic error detection system.
CHAPTER 8. CONCLUSION

Analysis in Chapter 3 has sought to use the SSR corpus to quantify the range, complexity, and frequencies of speaker errors common in spontaneous speech given a set of manual reconstruction annotations. This empirical overview indicates the errors which are most frequent and problematic, including deletions (most prevalently repetitions/revisions, fillers, and restart fragments) and phrase reorderings. These insights have later been helpful in selecting the subset of speech errors emphasized in initial reconstruction efforts and designing relevant features for their identification.

The surface transformations recorded in our reconstruction annotations reflect underlying influences in psycholinguistics and speech production models of spontaneous speakers. Chapter 4 consists of an investigation of naturally occurring speaker errors with manual semantico-syntactic analysis. Quantifying syntactic overlap between the automatic (and manual) parses of the verbatim and reconstructed string, along with generalizing the relationship between reconstructive changes and semantic labels projected onto the original string, has yielded additional insight into the impact of spoken language on semantic structure and context and into the way these features may be used in future reconstruction efforts.

8.1.2 Applied approaches for error detection and correction

Chapter 5 surveys past work in disfluency detection, and additionally considered several string transformation tasks and their potential relationship/impact on reconstruction efforts.
CHAPTER 8. CONCLUSION

Currently applied approaches for automatic paraphrasing, for example, consist primarily of compression approaches – where certain parsed (and potentially disfluent) phrases are selectively removed – and by string alignments and transformations – where, given adequate parallel information, transformation templates may in the future be able to assist in making appropriate substitutions, deletions, and insertions during the reconstruction process. Summarization currently consists primarily of extractive summarization techniques where words are deleted or preserved on a sentence (or sometimes phrase) level. While this may have value for the reconstruction of data like the European Parliamentary Proceedings, there is less potential for the reconstruction of truly spontaneous speech transcripts. Posing reconstruction as a statistical machine translation problem has shown potential, but analysis of out-of-the-box methods for SMT indicates weaknesses of the paradigm which would need to be addressed, such as the lack of many-to-none alignments (i.e. deletions) or the power to generalized disfluency patterns beyond examples seen in training.

Beyond string transformation approaches for tasks related to the goals of speech reconstruction, we also discuss in Chapter 5 the potential gains in automatic speech recognition (ASR) accuracy if spontaneous speech reconstruction were to successfully be applied to speech recognition systems. In other words, we have considered initial output of a recognizer, in lattice form, which could be subjected to a reconstruction-based reranking. We have found that less than half of content words missed in 1-best ASR output may be found elsewhere in the lattices, indicating that no amount of reranking can repair the errors if content-based reconstruction were applied in a lattice reranking framework. Additionally,
ASR errors are more prevalent in regions where speaker errors have occurred, which does not bode well for the direct reconstruction of ASR output in its current state.

Given our accumulated knowledge about the types, frequencies, contexts, and drivers of speaker-generated errors in spontaneous speech, in Chapter 6 we built a system which aimed to automatically identify and correct a subset of the most frequent errors. Using a conditional random field classification model and a variety of lexical, syntactic, and shallow semantic features, we demonstrate improvement in the correction of these errors over a state-of-the-art system. Chapter 7 has extended this approach by first building a maximum entropy classifier to identify errorful sentences and then integrating the two approaches to further improve classification accuracy.

The improvements demonstrated in the experimental results of Chapters 6 and 7 represent an improvement over the previous state of the art. Though no “killer” feature has been identified in the feature analysis for these experiments, a number of promising new features, such as the c-command indicator for errorful sentence-like unit (SU) identification have been proposed and have demonstrated success on which one may build further. Experiments in these chapters suggest that the JC04 features are important, but not independently of the other features. We demonstrated that automatically selecting a subset of SUs upon which to implement reconstruction improves the accuracy of non-copy (restart fragment) reparanda identification and cleaning, though less improvement results from doing the same for rough copy identification.

The primary source of obtained gains, beyond data inspired features and intelligent
CHAPTER 8. CONCLUSION

reconstruction targeting, is considered to be the enriched SSR reconstruction corpus itself, and the error class separation it allows for both training and testing of non-copy reparandum errors (as compared to rough copy errors).

8.2 Future work

Our research has concentrated on identifying and analyzing the reconstruction problems found in spontaneous text, learning to automatically detect these problem regions, and producing a richly annotated corpus of reconstructed data in preparation for supervised training of statistical classifiers.

In the future we hope researchers will apply this knowledge and these new resources to implement and evaluate a series of reconstruction approaches on both strings and syntactic trees.

8.2.1 Applications for the Spontaneous Speech Reconstruction corpus

As suggested throughout the dissertation, the Spontaneous Speech Reconstruction (SSR) corpus was created with the hope that it would be relevant and indeed valuable for research in the years ahead.

Psycholinguists may find additional observations of value in the thoroughly annotated
CHAPTER 8. CONCLUSION

data; these may include error patterns undocumented in this text or even more theoretical questions such as the expressive power required for a transducer to produce reconstructive transformations equivalent to manually produced verbatim-to-reconstructed string alignments such as those in Figure 4.3\textsuperscript{1}.

We believe that Chapter 4 serves as a general proof of concept that the automatic semantic role labeling (SRL) of verbatim speech text may conceivably be produced in the future. This is supported by the similarity of common predicate-argument paths between this data and the PropBank Wall Street Journal annotations (Palmer et al., 2005) and the consistency of other features currently emphasized in automatic SRL work on clean text data. To automatically semantically label speech transcripts, however, is expected to require additional annotated data beyond the 3,626 utterances multiply-annotated for SRL included in the SSR corpus, though it may be adequate for initial adaptation studies.

With this and other future applications in mind, we hope that the reconstruction framework presented and annotation guidelines utilized will themselves be useful for future reconstruction annotation of additional data, both in other spontaneous speech formats such as meetings and lectures, as well as languages beyond English.

8.2.2 Efforts in spontaneous speech reconstruction

Language engineers should expand upon the features used in Chapters 6 and 7 to more directly implement the insights provided by the annotations. While some success and im-

\textsuperscript{1}Thanks to Mark Johnson for bringing up such considerations during discussions of this work.
CHAPTER 8. CONCLUSION

Improvements for the automatic detection and deletion of fillers and reparanda (i.e. “simple cleanup”) have been demonstrated in this work, much remains to be done to adequately address the issues and criteria considered here for full reconstruction of spontaneous speech.

Included features for both the word level and SU-level error detection have only skimmed the surface of potentially powerful features for spontaneous speech reconstruction. There should be continued development of complementary parser-based features (such as those from dependency parsers or even deep syntax parsers such as implementations of Head-driven Phrase Structure Grammar (HPSG; Pollard and Sag (1994); Callmeier (2001)). as well as additional syntactic features based on automatic constituent or context-free grammar based parsers.

Prosodic features, though demonstrated to be unnecessary for at least moderately successful detection of simple errors, do hold a great deal of promise for additional gains. Future investigators should evaluate the gains possible by integrating acoustic information into the features and ideas presented here.

8.3 Conclusions

We have conducted an analysis of the types and frequencies of construction errors in spontaneous speech transcripts, presented methods for automatically identifying poorly constructed sentences, and produced a richly annotated set of manually reconstructed utterances from conversational speech. We have proposed several approaches for reconstruc-
CHAPTER 8. CONCLUSION

tion, borrowing from methods previously used in simple disfluency detection, machine translation, and semantic role labeling.

We anticipate that these methods will yield strong initial results in the area of speech reconstruction (as measured by BLEU, WER, or similarity and fluency measures), and will motivate comprehensive speech reconstruction research efforts in the years ahead.
Appendix I: The Penn Treebank Tag Set

The part-of-speech (POS) tagset used in the Spontaneous Speech Reconstruction (SSR) corpus and referred to in this thesis, described in Marcus et al. (1994). These tags are listed below, and were collected from http://bulba.sdsu.edu/jeanette/thesis/PennTags.html.

**WORD/Terminal Level**

- **CC**: Coordinating conjunction
- **CD**: Cardinal number
- **DT**: Determiner
- **EX**: Existential there
- **FW**: Foreign word
- **IN**: Preposition or subordinating conjunction
- **JJ**: Adjective
- **JJR**: Adjective, comparative
- **JJS**: Adjective, superlative
- **LS**: List item marker
### APPENDIX I: THE PENN TREEBANK TAG SET

<table>
<thead>
<tr>
<th>Tag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MD</td>
<td>Modal</td>
</tr>
<tr>
<td>NN</td>
<td>Noun, singular or mass</td>
</tr>
<tr>
<td>NNS</td>
<td>Noun, plural</td>
</tr>
<tr>
<td>NP</td>
<td>Proper noun, singular</td>
</tr>
<tr>
<td>NPS</td>
<td>Proper noun, plural</td>
</tr>
<tr>
<td>PDT</td>
<td>Predeterminer</td>
</tr>
<tr>
<td>POS</td>
<td>Possessive ending</td>
</tr>
<tr>
<td>PP</td>
<td>Personal pronoun</td>
</tr>
<tr>
<td>PP$</td>
<td>Possessive pronoun</td>
</tr>
<tr>
<td>RB</td>
<td>Adverb</td>
</tr>
<tr>
<td>RBR</td>
<td>Adverb, comparative</td>
</tr>
<tr>
<td>RBS</td>
<td>Adverb, superlative</td>
</tr>
<tr>
<td>RP</td>
<td>Particle</td>
</tr>
<tr>
<td>SYM</td>
<td>Symbol</td>
</tr>
<tr>
<td>TO</td>
<td>to</td>
</tr>
<tr>
<td>UH</td>
<td>Interjection</td>
</tr>
<tr>
<td>VB</td>
<td>Verb, base form</td>
</tr>
<tr>
<td>VBD</td>
<td>Verb, past tense</td>
</tr>
<tr>
<td>VBG</td>
<td>Verb, gerund or present participle</td>
</tr>
<tr>
<td>VBN</td>
<td>Verb, past participle</td>
</tr>
<tr>
<td>VBP</td>
<td>Verb, non-3rd person singular present</td>
</tr>
</tbody>
</table>
APPENDIX I: THE PENN TREEBANK TAG SET

<table>
<thead>
<tr>
<th>Tag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBZ</td>
<td>Verb, 3rd person singular present</td>
</tr>
<tr>
<td>WDT</td>
<td>Wh-determiner</td>
</tr>
<tr>
<td>WP</td>
<td>Wh-pronoun</td>
</tr>
<tr>
<td>WP$</td>
<td>Possessive wh-pronoun</td>
</tr>
<tr>
<td>WRB</td>
<td>Wh-adverb</td>
</tr>
</tbody>
</table>

**Phrase Level**

<table>
<thead>
<tr>
<th>Tag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADJP</td>
<td>Adjective Phrase.</td>
</tr>
<tr>
<td>ADVP</td>
<td>Adverb Phrase.</td>
</tr>
<tr>
<td>CONJP</td>
<td>Conjunction Phrase.</td>
</tr>
<tr>
<td>FRAG</td>
<td>Fragment.</td>
</tr>
<tr>
<td>INTJ</td>
<td>Interjection. Corresponds approximately to the part-of-speech tag UH.</td>
</tr>
<tr>
<td>LST</td>
<td>List marker. Includes surrounding punctuation.</td>
</tr>
<tr>
<td>NAC</td>
<td>Not a Constituent; used to show the scope of certain prenominal modifiers within an NP.</td>
</tr>
<tr>
<td>NP</td>
<td>Noun Phrase.</td>
</tr>
<tr>
<td>NX</td>
<td>Used within certain complex NPs to mark the head of the NP.</td>
</tr>
<tr>
<td>PP</td>
<td>Prepositional Phrase.</td>
</tr>
<tr>
<td>PRN</td>
<td>Parenthetical.</td>
</tr>
<tr>
<td>PRT</td>
<td>Particle. Category for words that should be tagged RP.</td>
</tr>
<tr>
<td>QP</td>
<td>Quantifier Phrase (i.e. complex measure/amount phrase); used within NP.</td>
</tr>
</tbody>
</table>
APPENDIX I: THE PENN TREEBANK TAG SET

RRC  Reduced Relative Clause.
UCP  Unlike Coordinated Phrase.
VP   Verb Phrase.
WHADJP  Wh-adjective Phrase. Adjectival phrase containing a wh-adverb, as in how hot.
WHAVP  Wh-adverb Phrase. Introduces a clause with an NP gap.
WHNP  Wh-noun Phrase. Introduces a clause with an NP gap.
WHPP  Wh-prepositional Phrase. Prepositional phrase containing a wh-noun phrase (such as “of which” or “by whose authority) that either introduces a PP gap or is contained by a WHNP.
X    Unknown, uncertain, or unbracketable. X is often used for bracketing typos.

CLAUSE LEVEL

S    Simple declarative clause
SBAR Clause introduced by a (possibly empty) subordinating conjunction.
SBARQ Direct question introduced by a wh-word or a wh-phrase.
SINV Inverted declarative sentence, i.e. one in which the subject follows the tensed verb or modal.
SQ   Inverted yes/no question, or main clause of a wh-question, following the wh-phrase in SBARQ.
Appendix II: Spontaneous Speech

Reconstruction corpus repartition

Discussion extended from Section 2.7.

To be certain that the data distribution is similar across the Spontaneous Speech Reconstruction (SSR) partitions, we repartitioned the Fisher DEV1, DEV2, and EVAL into three SSR train, development, and test sets: 17,162 training sentences (119,693 words), 2,191 sentences (14,861 words) in the development set, and 2,288 sentences (15,382 words) in the test set. The repartitioning is done by taking

- SSR\_TRAIN: the first 80% of conversations of Fisher DEV1, DEV2, and EVAL
- SSR\_CHECK: the next 10% of conversations of Fisher DEV1, DEV2, and EVAL
- SSR\_TEST: the last 10% of conversations of Fisher DEV1, DEV2, and EVAL

Conversation-level breakdown of this repartition can be seen in Tables 8.1-8.3
APPENDIX II: SPONTANEOUS SPEECH RECONSTRUCTION CORPUS REPARTITION

Table 8.1: Conversations included in resegmented SSR_CHECK and SSR_TEST.

<table>
<thead>
<tr>
<th>Conversation label</th>
<th>Fisher partition</th>
<th>SSR partition</th>
<th>Conversation label</th>
<th>Fisher partition</th>
<th>SSR partition</th>
</tr>
</thead>
<tbody>
<tr>
<td>fsh_117296</td>
<td>DEV2</td>
<td>SSR_CHECK</td>
<td>fsh_117936</td>
<td>DEV2</td>
<td>SSR_TEST</td>
</tr>
<tr>
<td>fsh_117537</td>
<td>DEV2</td>
<td>SSR_CHECK</td>
<td>fsh_118378</td>
<td>DEV2</td>
<td>SSR_TEST</td>
</tr>
<tr>
<td>fsh_117716</td>
<td>DEV2</td>
<td>SSR_CHECK</td>
<td>fsh_118586</td>
<td>DEV2</td>
<td>SSR_TEST</td>
</tr>
<tr>
<td>fsh_118878</td>
<td>EVAL</td>
<td>SSR_CHECK</td>
<td>fsh_119159</td>
<td>DEV2</td>
<td>SSR_TEST</td>
</tr>
<tr>
<td>fsh_118999</td>
<td>EVAL</td>
<td>SSR_CHECK</td>
<td>fsh_119319</td>
<td>EVAL</td>
<td>SSR_TEST</td>
</tr>
<tr>
<td>fsh_119199</td>
<td>EVAL</td>
<td>SSR_CHECK</td>
<td>fsh_119379</td>
<td>EVAL</td>
<td>SSR_TEST</td>
</tr>
<tr>
<td>fsh_119259</td>
<td>EVAL</td>
<td>SSR_CHECK</td>
<td>fsh_61158</td>
<td>EVAL</td>
<td>SSR_TEST</td>
</tr>
<tr>
<td>sw_46565</td>
<td>DEV1</td>
<td>SSR_CHECK</td>
<td>sw_46938</td>
<td>DEV1</td>
<td>SSR_TEST</td>
</tr>
<tr>
<td>sw_46615</td>
<td>DEV1</td>
<td>SSR_CHECK</td>
<td>sw_47038</td>
<td>DEV1</td>
<td>SSR_TEST</td>
</tr>
<tr>
<td>sw_46671</td>
<td>DEV1</td>
<td>SSR_CHECK</td>
<td>sw_47073</td>
<td>DEV1</td>
<td>SSR_TEST</td>
</tr>
<tr>
<td>sw_46677</td>
<td>DEV1</td>
<td>SSR_CHECK</td>
<td>sw_47175</td>
<td>DEV1</td>
<td>SSR_TEST</td>
</tr>
<tr>
<td>sw_46732</td>
<td>DEV1</td>
<td>SSR_CHECK</td>
<td>sw_47282</td>
<td>DEV1</td>
<td>SSR_TEST</td>
</tr>
<tr>
<td>sw_46789</td>
<td>DEV1</td>
<td>SSR_CHECK</td>
<td>sw_47346</td>
<td>DEV1</td>
<td>SSR_TEST</td>
</tr>
<tr>
<td>sw_46868</td>
<td>DEV1</td>
<td>SSR_CHECK</td>
<td>sw_47411</td>
<td>DEV1</td>
<td>SSR_TEST</td>
</tr>
</tbody>
</table>
### APPENDIX II: SPONTANEOUS SPEECH RECONSTRUCTION CORPUS REPARTITION

<table>
<thead>
<tr>
<th>Conversation label</th>
<th>Fisher partition</th>
<th>SSR partition</th>
<th>Conversation label</th>
<th>Fisher partition</th>
<th>SSR partition</th>
</tr>
</thead>
<tbody>
<tr>
<td>fsh_60262</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>sw_45355</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_60354</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>sw_45454</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_60386</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>sw_45481</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_60398</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>sw_45586</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_60416</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>sw_45626</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_60441</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>sw_45654</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_60463</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>sw_45713</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_60477</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>sw_45727</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_60493</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>sw_45819</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_60549</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>sw_45837</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_60568</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>sw_45856</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_60571</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>sw_45973</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_60593</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>sw_46028</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_60613</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>sw_46140</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_60627</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>sw_46168</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_60648</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>sw_46412</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_60650</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>sw_46455</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_60668</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>sw_46512</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_60682</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>fsh_109487</td>
<td>DEV2</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_60720</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>fsh_110103</td>
<td>DEV2</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_60732</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>fsh_110183</td>
<td>DEV2</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_60784</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>fsh_110346</td>
<td>DEV2</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_60797</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>fsh_110484</td>
<td>DEV2</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_60817</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>fsh_110485</td>
<td>DEV2</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_60844</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>fsh_110703</td>
<td>DEV2</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_60862</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>fsh_110747</td>
<td>DEV2</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_60874</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>fsh_111246</td>
<td>DEV2</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_60885</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>fsh_11130</td>
<td>DEV2</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_61039</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>fsh_111956</td>
<td>DEV2</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_61113</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>fsh_111993</td>
<td>DEV2</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_61130</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>fsh_112048</td>
<td>DEV2</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_61148</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>fsh_112707</td>
<td>DEV2</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_61192</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>fsh_113167</td>
<td>DEV2</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_61225</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>fsh_113312</td>
<td>DEV2</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_61228</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>fsh_113585</td>
<td>DEV2</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>sw_45097</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>fsh_113866</td>
<td>DEV2</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>sw_45104</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>fsh_114570</td>
<td>DEV2</td>
<td>SSR_TRAIN</td>
</tr>
</tbody>
</table>

Table 8.2: Conversations included in resegmented SSR_TRAIN subcorpus.
APPENDIX II: SPONTANEOUS SPEECH RECONSTRUCTION CORPUS
REPARTITION

<table>
<thead>
<tr>
<th>Conversation label</th>
<th>Fisher partition</th>
<th>SSR partition</th>
<th>Conversation label</th>
<th>Fisher partition</th>
<th>SSR partition</th>
</tr>
</thead>
<tbody>
<tr>
<td>sw_45142</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>fsh_114830</td>
<td>DEV2</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>sw_45237</td>
<td>DEV1</td>
<td>SSR_TRAIN</td>
<td>fsh_115051</td>
<td>DEV2</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_115069</td>
<td>DEV2</td>
<td>SSR_TRAIN</td>
<td>fsh_114246</td>
<td>EVAL</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_115112</td>
<td>DEV2</td>
<td>SSR_TRAIN</td>
<td>fsh_114429</td>
<td>EVAL</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_115133</td>
<td>DEV2</td>
<td>SSR_TRAIN</td>
<td>fsh_114430</td>
<td>EVAL</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_115136</td>
<td>DEV2</td>
<td>SSR_TRAIN</td>
<td>fsh_114651</td>
<td>EVAL</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_115352</td>
<td>DEV2</td>
<td>SSR_TRAIN</td>
<td>fsh_114849</td>
<td>EVAL</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_115354</td>
<td>DEV2</td>
<td>SSR_TRAIN</td>
<td>fsh_115138</td>
<td>EVAL</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_115679</td>
<td>DEV2</td>
<td>SSR_TRAIN</td>
<td>fsh_115153</td>
<td>EVAL</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_109841</td>
<td>EVAL</td>
<td>SSR_TRAIN</td>
<td>fsh_115231</td>
<td>EVAL</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_110347</td>
<td>EVAL</td>
<td>SSR_TRAIN</td>
<td>fsh_117316</td>
<td>EVAL</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_110863</td>
<td>EVAL</td>
<td>SSR_TRAIN</td>
<td>fsh_117496</td>
<td>EVAL</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_111106</td>
<td>EVAL</td>
<td>SSR_TRAIN</td>
<td>fsh_117635</td>
<td>EVAL</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_111585</td>
<td>EVAL</td>
<td>SSR_TRAIN</td>
<td>fsh_117918</td>
<td>EVAL</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_111905</td>
<td>EVAL</td>
<td>SSR_TRAIN</td>
<td>fsh_117976</td>
<td>EVAL</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_112045</td>
<td>EVAL</td>
<td>SSR_TRAIN</td>
<td>fsh_118177</td>
<td>EVAL</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_112666</td>
<td>EVAL</td>
<td>SSR_TRAIN</td>
<td>fsh_118419</td>
<td>EVAL</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_112825</td>
<td>EVAL</td>
<td>SSR_TRAIN</td>
<td>fsh_118578</td>
<td>EVAL</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_113405</td>
<td>EVAL</td>
<td>SSR_TRAIN</td>
<td>fsh_118698</td>
<td>EVAL</td>
<td>SSR_TRAIN</td>
</tr>
<tr>
<td>fsh_113806</td>
<td>EVAL</td>
<td>SSR_TRAIN</td>
<td>fsh_118778</td>
<td>EVAL</td>
<td>SSR_TRAIN</td>
</tr>
</tbody>
</table>

Table 8.3: More conversations included in resegmented SSR_TRAIN subcorpus.
Appendix III: Examples of interannotator SU segmentation disagreements

Of 6,384 annotated SUs, there were 268 cases when one or both annotators altered the SU boundary placements such that the two cases could not easily be resolved. Our goal in this analysis is to determine a set of common disagreement patterns with a corresponding treatment to resolve the disagreement.

Below are a few of the examples where superimposing original (or taking the union of all) boundaries leads to an issue with word alignments crossing over SU boundaries (not permitted in this data).

**EX. 1: Split vs. Integrate (FSH_60354_B #34)**

Original: “how about any members of your family have they changed anything”

SSR #1: (split)

191
APPENDIX III: EXAMPLES OF INTERANNOTATOR SU SEGMENTATION DISAGREEMENTS

how about any members of your family
have they changed anything

SSR #2: *(combined)*

have any members of your family changed anything,
where “they” is deleted as a coreference to “any members of your family”, which are then
moved into its place

We first consider taking the union of the SU boundaries (adding new boundaries to the
original string). However this violates the stated restrictions as #2 would have alignments
crossing the SU boundary. On the other hand, if we consider only the intersection of the
proposed SU boundaries, the original string would be left unmodified, and thus ill-formed.

We would choose Option 2, but mark #1’s reconst as not completely repaired, thus
giving precedence to #2 in training.

EX. 2: Leave vs. join/integrate *(fsh.60368_B #62)*

Original: at
gorgia tech
gorgia tech

SSR #1: *(no change)*

at
gorgia tech
APPENDIX III: EXAMPLES OF INTERANNOTATOR SU SEGMENTATION DISAGREEMENTS

gorgia tech

SSR #2: (joined segments and deleted repetition)

at gorgia tech

Option 1: take union of SU boundaries (same as orig): impossible because #2 would have alignments crossing an SU boundary

Option 2: take minimum of SU boundaries (mandates removed boundaries): possible, though annotation #1 left unoptimized w/ repeat

EXAMPLE 3: LEAVE VS. JOIN+DELETE (FSH_60441_B #10)

Original: at least that's what

    well [...] if I knew about it I really wouldn't have any other choice

SSR #1: (no bound change)

    at least that's what

    if I knew about it I really wouldn't have any other choice

SSR #2: if I knew about it I really wouldn't have any other choice

    (“at least that's what” joined with next SU, then deleted as a false start and aligned to the first preserved word: “if”)

Option 1: take union of SU boundaries (same as original): impossible by itself because #2’s false start aligns in effect to nothing (possibility: keep all false start words, make own SU, and label SU as ”fragment without content” which in effect means ”delete me”)

193
APPENDIX III: EXAMPLES OF INTERANNOTATOR SU SEGMENTATION DISAGREEMENTS

Option 2: take minimum of SU boundaries: possible, though annotation #1 left unoptimized.

EXAMPLE 4: Movement across SU boundary (fsh_60441_B #30)

Original: especially if it’s something major that you know would be life impacting to i mean not just your brother but your parents you know any other siblings it would be really hard to i mean because you just basically have to i mean you have to make a decision

SSR #1: especially if it’s something major that would be life impacting to not just your brother but your parents and any other siblings

it would be really hard to because you have to make a decision

SSR #2: it would be really hard to make a decision especially if it’s something major that would be life impacting to not just your brother but your parents and any other siblings

These are effectively the same, except in #2 the first clause (“especially... siblings”) is moved rightward to be a complement of the second (“it...decision”)

Option 1: take union of SU boundaries: impossible because #2 would have alignments crossing the SU boundary

Option 2: take minimum of SU boundaries (same as orig): okay
Glossary

automatic speech recognition (ASR) The automatic transcription of speech from audio to written form, or a system which accomplishes this.

backchannel acknowledgement Any utterance not considered incomplete but also not to add new content to the dialogue. Its presence indicates only that the speaker is still paying attention to the other listener.

c-command Given a one non-terminal (NT) tree node $A$ and a second NT $P$, we say that $A$ locally c-commands $P$ and its descendants if (1) neither node $A$ dominates $P$ nor node $P$ dominates $A$ and (2) node $A$ immediately precedes $P$ in the tree.
Glossary

**complement**
A non-terminal tree node locally c-commanded by and modifying some lexical category type X, such as objectival noun phrases modifying verbs or the complementing phrase “who went home” in the noun phrase “students who went home”.

**conditional random fields (CRF)**
Undirected graphical models whose prediction of a sequence of hidden variables Y is globally conditioned on a given observation sequence X.

**context-free grammar (CFG)**
A formal grammar in which every production rule is of the form \([P \rightarrow C]\), where \(P\) is a single non-terminal symbol and \(C\) is a (possibly empty) string of terminals and non-terminals which in turn can be expanded into another string of symbols. *Context-free* refers to the condition that \(P\) can be rewritten into some string \(C\) independently of its surrounding context.

**conversational telephone speech (CTS)**
A particular domain of recorded and transcribed data used for NLP research.
Glossary

**disfluency**  
Any deviation in speech from ideal delivery (Ferreira and Bailey, 2004).

**dominance**  
A tree node $A$ dominates a tree node $B$ if $A$ is directly above $B$ (such as in a parent or grandparent relationship).

**efficacy**  
In the medical field, this term refers to whether intervention can be successful when it is properly implemented under controlled conditions (as opposed to the *effectiveness* of real-world attempts). Likewise, *reconstruction efficacy* reflects best-possible attempts to manually reconstruct utterances into well-formed, grammatical, and content-preserved sentences. Annotators label the final state of each manual reconstruction with its level of grammaticality.
Glossary

**filler (FL)**
Filled pauses like “um”, “ah” and discourse markers like “you know” produced by speakers during spontaneous speech. Also called *abridged repairs* by Heeman and Allen (1999).

**Head-driven Phrase Structure Grammar (HPSG)**
A deep-syntax phrase structure grammar which produces rich, non-context-free syntactic analyses of input sentences based on a collection of manually constructed rules and lexical item structures (Pollard and Sag, 1994).

**hidden Markov model (HMM)**
A statistical model in which the likelihood of any future state in the modeled system is assumed to depend only on the present state (*a Markov assumption*). In this generative model, the values of hidden parameters are calculated based on observable parameters.
Glossary

**interregnum**
In disfluent spontaneous speech, an indication (such as “I mean” or a filled pause like “uh”) given by a speaker to a listener prior to a self-correcting *repair*. See Figure 1.3. Also known as an *editing term* (Heeman and Allen, 1999).

**interruption point (IP)**
The point following a speaker reparandum and preceding an optional interregnum and a speaker self-repair, where a speaker interrupts his or herself to repeat, correct, or otherwise replace the previously stated reparanda. See Figure 1.3. Also called a *moment of interruption* (MOI) (Levelt, 1983) or an *edit signal* (Hin-dle, 1983).

**JC04**
A reference to the simple edit detection approach described in Johnson and Charniak (2004), which is one baseline considered in this work.
### Glossary

**kappa statistic** $\kappa$  
An index comparing the agreement between two sets of categorical decisions against that which might be expected by chance, ranging from +1 (perfect agreement) via 0 (no agreement above that expected by chance) to -1 (complete disagreement). Defined in Equation 3.12.

**language model (LM)**  
Estimates the probability of a word sequence, typically predicted as the product of probabilities of each word or word class given the words or word classes seen previously.

### Linguistic Data

**Consortium (LDC)**  
An open consortium which creates, collects and distributes speech and text databases, hosted at the University of Pennsylvania (http://www.ldc.upenn.edu/).

**n-gram**  
A subsequence of $n \in \mathbb{Z}_+$ items. In an $n$-gram language model, the probability of any given word is approximated to be conditioned only on the $n - 1$ previous words or word classes.
Glossary

**NLP**
The field of natural language processing.

**noisy channel**

**transformation model**
Noisy channel model approaches to natural language processing (described in Equation 5.3) typically seek the fluent but unknown “source” string most likely to have been “distorted” to produce the observed string.

**non-copy (NC) error**
Speaker error where the reparandum has no lexical or structural relationship to the repair region following, as seen in restart fragments and some revisions.

**non-terminal (NT) symbol**
Any symbol which, in a string or tree, can be mapped to production rules for other terminals or non-terminals.

**partial word**
An incompletely articulated word in a speech transcript, also known as a word fragment.

**precedence**
A tree node $A$ precedes a tree node $B$ if $A$ is graphically to the left of (and is not in a dominance relationship with) $B$ in a tree.
<table>
<thead>
<tr>
<th><strong>Glossary</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>prosody</strong></td>
</tr>
<tr>
<td><strong>repair region</strong></td>
</tr>
<tr>
<td><strong>reparandum</strong></td>
</tr>
<tr>
<td><strong>repetition</strong></td>
</tr>
</tbody>
</table>
Glossary

**restart fragment**
In a **restart fragment**, an utterance is aborted and then restarted with a new train of thought. Also called *false starts, fresh starts* (Heeman and Allen, 1999) and *deletions (DE)* (Shriberg, 1994).

**revision**
In a **revision**, the repair phrase alters reparandum words to correct the previously stated thought. Also called *modification repairs* (Heeman and Allen, 1999).

**roleset**
The set of core and optional semantic argument roles associated with a given verb.

**rough copy (RC) error**
Speaker error where the reparandum incorporates the same or very similar words in roughly the same word order as the corresponding repair region.

**semantic role labeling (SRL)**
The task of predicate-argument labeling, also called semantic role labeling or SRL, assigns a simple *who did what to whom when, where, why, how*, etc. structure to sentences.
Glossary

**sentence-like unit (SU)**  A unit of speech which expresses a thought or idea but does not necessarily fit the formal definition of a grammatical sentence despite playing a similar semantic function (Fiscus et al., 2004).

**simple cleanup**  A precursor to full speech reconstruction. In this approach, all detected filler words are deleted, and the reparanda and interregna are deleted while the repair region is left intact.

**simple disfluency**  Speaker-produced errors including *fillers, reparanda,* and *interregna* which can in general be repaired by deletion. Also known as DFs (Shriberg, 1994), etc.

**SimpleMDE data**  Data created for the 2004 NIST Rich Transcription Metadata Extraction (MDE) task to fulfill the goal of providing training data for accurate sentence segmentation and the identification of simple disfluencies and reparandum regions of the type shown in Figure 1.3.

**specifier**  A locally c-commanding node of some lexical category type $X$, such as determiners modifying nouns or subjectival noun phrases modifying verbs.
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>speech reconstruction</strong></td>
<td>A transformation of spontaneous speech input such that the output were to represent, in flawless, fluent, and content-preserving text, the message that the speaker intended to convey.</td>
</tr>
<tr>
<td><strong>Switchboard (SWBD) corpus</strong></td>
<td>A conversational telephone speech (CTS) corpus described in Godfrey et al. (1992).</td>
</tr>
<tr>
<td><strong>tree-adjoining grammar (TAG)</strong></td>
<td>A grammar formalism presented by Joshi and Schabes (1997), similar to context-free grammars but using trees rather than symbols as the basic rewriting expansion units. This formalism models crossed word dependencies, and is used in the Johnson and Charniak (2004) simple disfluency detection work.</td>
</tr>
<tr>
<td><strong>treebank</strong></td>
<td>A set of manually generated syntactic parse trees.</td>
</tr>
</tbody>
</table>
Index

acoustics, 21, 147

automatic speech recognition (ASR), 1, 17, 21, 31, 35, 52, 135, 148


BLEU, 30, 54, 57, 58, 131, 132

c-command, 162, 162–164, 167, 173
dominance, 162
precedence, 162

conditional random fields (CRF), 142, 141–142

experimental results, 152, 153, 155, 171

construction error types, 8–9, 11–20

context-free grammar (CFG) parser, 117
data

Broadcast News, 132

conversational telephone speech (CTS), 2, 11, 20, 28, 31, 32, 143

Fisher (FSH) corpus, 28, 32–35, 38, 39, 47, 69, 143, 187

interview transcripts, 31, 32

lectures, 31, 32

meeting data, 31, 32

parliamentary proceedings, 31, 32, 127, 129

Proposition Bank (PropBank), 80

selection criteria, 31, 32

Simple MDE, 35

SimpleMDE, 28, 34, 40, 69

SSR corpus, see Spontaneous Speech Reconstruction (SSR) corpus

Switchboard (SWBD) corpus, 28, 34,
INDEX

132

Wall Street Journal (WSJ), 80, 98
discourse marker (DM), see filler
edit word error rate (EWER), 52, 118
efficacy, 86
exact SU match metric, 56, 58, 131, 155, 171
Explicit editing terms (EETs), 12
Fβ-score, 37, 55, 121, 152, 153, 166, 169
false start, see restart fragment
filled pause (FP), see filler (FL)
filler (FL), 8, 12, 118, 141, 149, 150, 152, 153, 155, 169–171
Head-driven Phrase Structure Grammar (HPSG), 37, 160, 161, 166, 167, 173, 181
ill-formed SU identification, 36–38, 158, 158–168
experimental results, 37, 166
information extraction, 1

interregnum, 8
interruption point (IP), 8
JC04, 37, 120, 138, 143–145, 153–155, 166, 173
language model (LM), 30, 117, 119, 130–132, 134, 140, 144, 145, 151
metadata extraction (MDE), 28
minTER, 53, 57, 58, 71
minWER, 53, 58
n-gram, 130
natural language generation (NLG), 123
NC, see non-copy (NC) error
NIST RT04 evaluation, 34
noisy channel model, 43, 85, 119, 120, 124, 130, 138
non-copy (NC) error, 121, 141, 149, 150, 152, 153, 155, 169–171
non-terminal (NT) phrase, 146
out of vocabulary (OOV) words, 136, 137
INDEX

paraphrasing, 3, 115, 123–127
  by sentence compression, 123
  by string alignments, 125
  from predicate-argument structures, 126
precision, 37, 55, 121
predicate-argument labeling, see semantic role labeling (SRL)
prosody, 21, 147
question answering (QA), 3, 123
RC, see rough copy (RC) error
recall, 37, 55, 121
reconstruction efficacy, 42, 44, 44–45, 58, 72
repair region, 8
reparandum, 8, 14
  repetition, 8, 121
  restart fragment, 9, 121
  revision, 8, 121
reparandum region, 118
roleset, 80
rough copy (RC) error, 120, 141, 149, 150, 152, 153, 155, 169–171
semantic role labeling (SRL), 46, 80, 80–84
sentence-like unit (SU), 5, 34
simple cleanup, 9, 172
  experimental results, 154, 155, 171
simple disfluency, 8, 37
speech reconstruction evaluation, 23, 52, 57, 71, 118, 121, 131, 151–155, 168, 169, 171
Spontaneous Speech Reconstruction (SSR) corpus, 27, 31, 49, 143, 175, 179
  annotation process, 39–47
  data selection for, 31–34
  extracting ill-formed SUs, 36–38
  motivation for building, 30
  repartitioning of, 187
statistical machine translation (SMT), 1, 3, 54, 115, 129–135
INDEX

summarization, 1, 3, 115, 123, 126–129

translation error rate (TER), 53

minTER, see minTER

tree-adjoining grammar (TAG), 79, 120, 143

word error rate (WER), 53, 136

minWER, see minWER
Bibliography


BIBLIOGRAPHY


Carletta, J. (2007). Unleashing the killer corpus: experiences in creating the multi-
BIBLIOGRAPHY


Cinková, S. and Hajič, J. (2009). An annotation scheme for speech reconstruction on a
BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


Vita

Erin Fitzgerald was born in Pittsburgh, Pennsylvania in 1980. After graduating from W. G. Enloe High School in Raleigh, North Carolina in 1998, she completed a B.S. in Electrical and Computer Engineering and a minor in music performance (Clarinet) from Carnegie Mellon University in Pittsburgh, Pennsylvania in 2002. She then completed her M.S.E. and Ph.D in Electrical and Computer Engineering from the Johns Hopkins University in 2004 and 2009, respectively.

Erin took advantage of several internship opportunities during to her academic career, including the Intel Corporation’s Speech Applications Group in the summer of 2000, Microsoft Corporation’s dotNet Speech Group in the summers of 2001 and 2002, and the BBN Technologies’s Speech & Language Group in the summer of 2004. In addition, she served as a graduate researcher for the Confidence Estimation for Natural Language research team in the 2003 Summer Workshop for Language Research help at Johns Hopkins. She also took advantage of her participation in the NSF Partnership in International Research Education grant by spending several months living and collaborating with researchers at Charles University in Prague, Czech Republic and Universität des Saarlandes in Saarbrücken, Ger-
VITA

Upon the completion of her Ph.D. degree, Erin was selected to serve as a Christine Mirzayan Science and Technology Policy Graduate Fellow at the National Academies in Washington, D.C.