

Automatic Sentence Structure Annotation for Spoken Language Processing

Dustin Lundring Hillard

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

University of Washington

2008

Program Authorized to Offer Degree: Electrical Engineering

University of Washington
Graduate School

This is to certify that I have examined this copy of a doctoral dissertation by

Dustin Lundring Hillard

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Chair of the Supervisory Committee:

Mari Ostendorf

Reading Committee:

Mari Ostendorf

Jeff Bilmes

Andreas Stolcke

Date:

In presenting this dissertation in partial fulfillment of the requirements for the doctoral degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of this dissertation is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for copying or reproduction of this dissertation may be referred to Proquest Information and Learning, 300 North Zeeb Road, Ann Arbor, MI 48106-1346, 1-800-521-0600, to whom the author has granted "the right to reproduce and sell (a) copies of the manuscript in microform and/or (b) printed copies of the manuscript made from microform."

Signature_____

Date_____

University of Washington

Abstract

Automatic Sentence Structure Annotation
for Spoken Language Processing

Dustin Lunding Hillard

Chair of the Supervisory Committee:

Professor Mari Ostendorf

Electrical Engineering

Increasing amounts of easily available electronic data are precipitating a need for automatic processing that can aid humans in digesting large amounts of data. Speech and video are becoming an increasingly significant portion of on-line information, from news and television broadcasts, to oral histories, on-line lectures, or user generated content. Automatic processing of audio and video sources requires automatic speech recognition (ASR) in order to provide transcripts. Typical ASR generates only words, without punctuation, capitalization, or further structure. Many techniques available from natural language processing therefore suffer when applied to speech recognition output, because they assume the presence of reliable punctuation and structure. In addition, errors from automatic transcription also degrade the performance of downstream processing such as machine translation, name detection, or information retrieval. We develop approaches for automatically annotating structure in speech, including sentence and sub-sentence segmentation, and then turn towards optimizing ASR and annotation for downstream applications.

The impact of annotation is explored at the sentence and sub-sentence level. We describe our general approach for predicting sentence segmentation and dealing with uncertainty in ASR. A new ASR system combination approach is described that improves ASR more than any previously proposed methods. The impact of automatic segmentation in machine translation is also evaluated, and we find that optimizing segmentation directly for translation improves translation quality, performing as well (or better than) using reference segmentation. Turning to sub-sentence annotation, we

describe approaches for supervised comma detection and unsupervised learning of prosodic structure. The utility of automatic commas is then assessed in the context of information extraction and machine translation. Including commas in information extraction tasks significantly improves performance, especially when punctuation precision and recall are optimized directly for entity and relation extraction. We then also propose approaches for improving translation reordering models with cues from commas and sentence structure.

TABLE OF CONTENTS

	Page
List of Figures	iii
List of Tables	iv
Chapter 1: Introduction	1
1.1 Why is speech different?	1
1.2 Spoken Language Processing	3
1.3 Contributions	4
1.4 Dissertation Overview	5
Chapter 2: Background	7
2.1 Segmenting Speech	7
2.2 Prosodic Event Detection	13
2.3 Spoken Language Processing	14
2.4 Representing Uncertainty in Spoken Language Processing	18
Chapter 3: Accounting for ASR Uncertainty in Sentence Segmentation	23
3.1 Experimental Paradigm	25
3.2 Weighted SU Prediction with N-Best Sentence Hypotheses	27
3.3 Joint Word-SU Lattice Decoding	31
3.4 Conclusions	33
Chapter 4: ASR Uncertainty and System Combination	34
4.1 Compensating for Word Confidence Estimation Bias	34
4.2 ASR System Combination	43
4.3 Conclusions	52
Chapter 5: Improving Sentence Transcription for MT	53
5.1 Sentence Segmentation for Machine Translation	53
5.2 Parsing-based ASR Objectives for Machine Translation	60

5.3	Conclusions	71
Chapter 6:	Automatic detection of sub-sentence structure	72
6.1	Automatic Comma Detection	72
6.2	Unsupervised Learning of Prosodic Structure	78
6.3	Conclusion	84
Chapter 7:	Commas for Tagging	86
7.1	Corpora and Evaluation	86
7.2	Speech Tagging Framework	87
7.3	Part-of-Speech Tagging	89
7.4	Name Tagging	90
7.5	Conclusions	92
Chapter 8:	Commas for Information Extraction	94
8.1	Experimental Setup	94
8.2	Results	96
8.3	Discussion	98
8.4	Conclusion	99
Chapter 9:	Improving MT Reordering with Commas	103
9.1	Soft Boundaries	105
9.2	Maxent models for reordering	108
9.3	Conclusion	113
Chapter 10:	Conclusion	115
10.1	Main Conclusions	115
10.2	General Findings and Impact	117
10.3	Future Research	118
	Bibliography	121

LIST OF FIGURES

Figure Number	Page
3.1 DET curve illustrating SU detection error trade-offs for 1-best (blue line) vs. reference (red line) decoding for English CTS.	24
3.2 Confusion network for a single hypothesis.	28
3.3 Confusion network for a merged hypothesis.	29
3.4 DET curve illustrating SU detection error trade-offs for pruned 1-best (solid line) vs. confusion network (dashed line) decoding.	31
3.5 The trade-off in SU error rate and WER in lattice decoding on CTS.	32
4.1 Relative frequency that a hypothesized word is correct as a function of the predicted posterior in a CN.	36
4.2 Various feature sets using one SVM	40
4.3 Comparison of bias plots for original posteriors, compensation using the best single SVM, and compensation using the best dual SVM with thresholding.	41
4.4 DET curve for detecting slots with missing words.	42
6.1 Comma prediction for Mandarin TDT4 ASR words with reference sentence boundaries and three different modeling approaches.	75
8.1 IE performance on entities and relations when period and comma thresholds are varied from 0 to 1 (from left to right and bottom to top). Contours are displayed every 0.2 point drop from the highest score (artifacts are created by undersampling). The punctuation-optimal thresholds are indicated by dotted lines, the entity-optimal thresholds by dash-dot lines, and the relation-optimal thresholds by dashed lines.	101
8.2 Examples where noun phrase assignment is ambiguous due to a missed sentence boundary (1) or comma (2). Even if semantically unlikely, the assignment is usually syntactically correct. Similarly, inserting a punctuation mark in the middle of a noun phrase will result in a split.	101
8.3 Percentage of reference entity mention extents split by inserting commas or periods at their respective decision thresholds.	102
9.1 Example of $c(j)$ and $r(j)$ reordering penalties. p_b is the probability (confidence) of a comma boundary, and p_{nb} is the probability of no comma boundary ($1 - p_b$)	107

LIST OF TABLES

Table Number	Page
3.1 SU error rates for single best vs. confusion nets using N-best list pruning.	30
4.1 Corpus statistics for the EPPS English task of the 2006 TC-STAR Evaluation Campaign.	48
4.2 <i>WER[%] results for single systems.</i>	48
4.3 WER results for development data with different feature classes.	49
4.4 <i>WER[%] results for development data with manual segmentation, and using cross-validation for iROVER. All methods use the same alignment but different voting functions: majority-, confidence-, classifier-based-, and oracle-voting.</i>	50
4.5 <i>WER[%] results for evaluation data. All methods use the same alignment but different voting functions: majority-, confidence-, classifier-based-, and oracle-voting.</i>	51
5.1 Corpus statistics for the bilingual training data of the Chinese-to-English and Arabic-to-English MT systems (GALE large data track).	57
5.2 Segmentation and translation results [%] for different sentence segmentation settings on the Chinese-to-English task. The best results for each score are highlighted in boldface.	58
5.3 Segmentation and translation results [%] for different sentence segmentation settings on the Arabic-to-English task. The best results for each score are highlighted in boldface.	60
5.4 Correlation between two ASR scores (CER and SParseval) and two MT scores (HTER and TER) for Broadcast News. (TER score also provided for comparison with ASR scores.)	64
5.5 Correlation between two ASR scores (CER and SParseval) and two MT scores (HTER and TER) for Broadcast Conversations. (TER score also provided for comparison with ASR scores.)	65
5.6 ASR scores and MT scores on dev07-bn.	69
5.7 ASR scores and MT scores on dev07-bn (true sentences) for two ASR objectives: character error rate (CER) and SParseval F-score.	70
5.8 ASR scores and MT scores on dev07-bc (true sentences) for two ASR objectives: character error rate (CER) and SParseval F-score.	71

6.1	Confusion table counts for comma and caesura prediction on the Mandarin TDT4 held out set, using a .5 comma threshold and reference SUs.	74
6.2	Results for comma detection on Mandarin ASR transcripts with different thresholds for comma posteriors on GALE Y1 BN data. The far left column contains the threshold used in automatic SU detection, while the far right column contains the threshold used for comma detection under each SU setting.	77
6.3	Results for comma detection on English ASR transcripts with different thresholds for comma posteriors (reference sentence boundaries). Optimal F-score threshold of 0.68 selected on dev set.	78
6.4	Comparison of Unsupervised Emphasis Clustering Approaches	81
6.5	Emphasis Classification with Boostexter	81
6.6	Break Clustering Confusion Table	82
6.7	Unsupervised 2 Class Prosodic Break Clustering	83
6.8	Break Classification with Boostexter	83
7.1	POS tagging performance on various training/test conditions using the Viterbi tagger.	91
7.2	Named entity tagging performance on news text under different punctuation conditions.	92
8.1	Baseline IE performance for entities and relations on the test set. Various conditions are presented which include: reference words (Ref), machine generated words (ASR), reference punctuation (with and without commas) and fixed length punctuation.	97
8.2	IE performance when punctuation is self-optimized (Punc.) or optimized in order to improve entities (Ent.) and relations (Rel.). Period and comma decision thresholds (thr_p , thr_c) are chosen in order to maximize performance on the development set and used blindly on the test set. Punctuation F -measure is reported in F_p and F_c for periods and commas, respectively. Comma F -measure is reported using reference sentence boundaries.	98
9.1	Commas as reordering boundaries in a hand aligned section of the Chinese Treebank	104
9.2	Comma and translation results [%] for the different SU and soft boundary settings on the Chinese-to-English task.	108
9.3	Example of improved MT quality by using automatically predicted commas as soft boundaries (Chinese-to-English task).	109
9.4	Two-class phrase orientation prediction errors on Chinese-to-English newswire translations, where “LDC only” is the parallel corpora from LDC, and “with UN” includes the UN parallel corpora.	111

9.5	Four-class phrase orientation prediction errors on Chinese-to-English newswire translations, where “LDC only” is the parallel corpora from LDC, and “with UN” includes the UN parallel corpora.	112
9.6	Phrase orientation prediction error on Chinese-to-English newswire translations, testing with automatic commas (LDC corpora only).	112
9.7	MT BLEU score [%] on Chinese-to-English broadcast news reference words, with and without boundary features in phrase orientation prediction.	113
9.8	Example of improved MT quality by using commas from text as boundary features in phrase orientation modeling (Chinese-to-English task).	114

ACKNOWLEDGMENTS

This dissertation is the product of years of work, which in various respects has been nurtured by many colleagues (who I now count as friends) and also in no small part by my family. I certainly owe a significant and primary debt of gratitude to my advisor, Mari Ostendorf. With her untiring interest in my research she inspired me to continue throughout graduate school. She has provided countless insights that have contributed to the success of this work. She also made the process enjoyable with a sustaining balance of flexibility and encouragement that allowed me to thrive.

I would also like to thank the members of my committee: Jeff Bilmes, Andreas Stolcke, Fei Xia, and Efthimis Efthimiadis. Their time and comments are much appreciated and have improved the quality of this thesis.

I could not have asked for a better research environment than SSLI lab. Over my years at the lab I have learned much from more students than I can name here. I am particularly thankful for the friendship and collaborations I have with Jeremy Kahn. Conversations with him have helped bring better understanding, and always better code. Arindam Mandal helped me to learn the SRI recognizer, not an insignificant task. Sarah Schwarm Petersen was helpful throughout, and introduced me to the SRILM toolkit, and Ivan Bulyko further taught me about language modeling. Ozgur Cetin was willing to answer any question I had about speech recognition or machine learning and Scott Otterson has provided helpful advice in particular with prosodic break classification. I have also benefited from conversations with and insights from Andrei Alexandrescu, Chris Bartels, Becky Bates, Costas Boulis, Kevin Duh, Karim Filali, Sangyun Hahn, Xin Lei, Jon Malkin, Alex Marin, Sheila Reynolds, and Amar Subramanya. Finally Lee Damon provided a stable and productive computing environment (which can not be undervalued) and a valued friendship.

This work consists of many collaborations, and without the interest and effort of my

colleagues, most of this thesis would not have been possible. Throughout my research, colleagues at SRI and ICSI have always provided quick and helpful advice. Yang Liu was instrumental early in my research career in providing much help to me as I learned about sentence segmentation. Her willingness to help me understand her system and also in conveying her research experience was extremely helpful as I began my research. Andreas Stolcke also provided much advice and help, as well as the superb SRILM toolkit that has been a foundation to much of my research software. Elizabeth Shriberg was also kind in teaching me about her tools to generate prosodic features. Later, Dilek Hakkani-Tur, Mathew Magimai-Doss, and Benoit Favre were helpful collaborators and provided their time and expertise, in addition to the ICSI sentence segmentation system. Finally, Wen Wang was extremely helpful in providing tools and expertise for training language models.

I spent the spring and summer of 2006 in Aachen, Germany at I6, courteously hosted by Hermann Ney. Many of the students there were kind and I learned a significant amount about machine translation and also understood speech recognitions from new perspectives. In particular, Evgeny Matusov has been an ideal collaborator and friend. Without his help in running experiments and implementing new decoding approaches, most of the translation work in this dissertation would not have been possible. Arne Mauser, Richard Zens, Jia Xu, and Yuqi Zhang also provided assistance as I learned about their MT system and ran experiments. In addition, I also benefited from spontaneous discussions with Björn Hoffmeister that evolved into exciting research and a quality friendship. Sasan Hasan, Oliver Bender, Thomas Deselaers, Stefan Hahn, Jonas Löff, Gregor Leusch, Daniel Stein, Christian Plahl, Christian Golan, and Ralf Schüller were welcoming and helpful throughout my stay.

I have also benefited from great collaborations with Mary Harper and Zhongqiang Huang at Maryland, as well as Heng Ji and Ralph Grishman at NYU.

Foremost, the support and endurance of my family has been critical to my success. To my wife Julianna, I cannot convey the depth of gratitude which I feel for her continuing support, encouragement, and capacity to love. Many months of long hours and late nights undoubtedly created sometimes difficult living conditions, but she persevered and supported

me throughout. She has grown with me throughout my time at the University of Washington, and most of what I accomplish is better because of how she challenges and supports me.

To my mom and dad I credit my capacity to learn and interest in the world. My interest in becoming an engineer started with the example set by my dad, and was fostered by his limitless willingness (and even encouragement) to answer my sometimes unending questions about how things work. I owe my confidence in myself and my thoughts to the unquestionable support and love from my mom, who was always there to encourage me in hard times and celebrate with me in good times. My sister has put up with me like only a sibling could, and I love her very much for the kind and thoughtful person that she is. Julianna's parents have accepted me as their own child, an action that speaks volumes of their compassion. Their abiding interest in my research (and even editing of this document) has provided encouragement throughout. Finally my aunts, uncles, and cousins have been wonderfully supportive (even as they may have wondered why I stayed in school so long).

This large body of people have made this work possible. I feel privileged to count them all as friends and family.

Chapter 1

INTRODUCTION

Increasing Internet connectivity across the world is influencing the way that people and businesses accomplish their goals. Simple tasks of daily life, such as finding a movie time, looking up a weather forecast, or getting directions to a location, can now be easily accomplished on-line (or even from mobile devices). Increasingly, people also look to on-line information sources for news, answers to questions, or advice on what to buy. Facilitating these tasks is now a critical component of improving productivity in the daily working and personal lives of an increasing portion of the world.

While the primary form of information on the Internet has been textual, audio and video sources are rapidly becoming a prominent portion of on-line content. Large bodies of recorded speech such as broadcast news, oral histories, lectures, and user-generated content are now easily available and continuing to grow. The ability to automatically process and search through content in spoken documents from these sources is an important area of research. Developing approaches for automatically distilling information from spoken documents requires new techniques to deal with the differences between speech and text. This thesis will investigate methods for improving spoken language processing of speech and also propose approaches for addressing the unique problems that arise when working with spoken (as opposed to written) documents.

1.1 Why is speech different?

Successful approaches from text processing all assume an accurate word sequence and structure that is not available when working with audio and video sources, because usually a transcript is not available. In order to efficiently process large amounts of information from these richer media, automatic speech recognition is required to provide transcripts. It would not be feasible, for example, to have humans transcribe all of the available on-line news audio and video, or even all of the videos

on YouTube (especially in real time). As large vocabulary speech recognition has improved over recent years it has become increasingly interesting to explore language processing on speech sources (extending beyond the traditional domains of textual sources). The quality of automatic recognition has reached levels that make possible further analysis of recognition transcripts. Once a reasonably reliable recognition transcript has been obtained, then techniques from natural language processing of text, such as name detection, or translation from a source to target language, can be applied.

Typical speech recognition systems usually output the most likely string of words given the audio, determined by the combination of an acoustic model and a language model. This output has significant differences from what is typically assumed of textual data because it normally lacks any sentence boundaries, casing, other punctuation, or higher level structure such as paragraphs. The style of spoken documents is also different from text, in part because people use different cognitive processes when speaking as compared to writing. Differences arise from the phenomena of conversations (such as interruptions, mistakes, and corrections), more informal word choices and phrasing, and differing signals of intent and emphasis. In addition, the recognizer makes transcription errors when compared to reference transcripts. These errors can be as low as 3% for domains such as Broadcast News (BN), but can reach error rates of 30% and higher for domains such as recorded meetings with multiple speakers, or in noisy environments.

When approaches from Natural Language Processing (NLP) on text are applied to errorful speech recognition output, performance significantly degrades when compared to performance on human transcripts. A large portion of performance loss can be attributed to transcription errors, but NLP methods still lag behind performance on written text when they are applied to reference transcriptions that do not contain punctuation or case information. In addition, differences in style between written and spoken language introduce further challenges. Improving performance of NLP tasks for speech domains requires addressing all of these aspects: errors in transcription, lack of punctuation, sentence structure, and style mismatch.

The benefit of working with speech is that there are additional information sources when compared to text. In spoken language, speakers convey more than what is captured by the simple word string. Spoken communication uses prosody: the combination of pitch inflections, word durations, energy, pauses, and speaking rate. Prosody communicates additional information about structure, emotion, and intent. Detecting and modeling this additional stream of information allows for a re-

construction of the missing punctuation and structure in speech transcripts, as well as additional sources of information that are not well captured in written text, such as emphasis or doubt. When video is available, a further set of cues such as gestures, facial expressions, and body language are also available (but we will focus on only speech). Augmenting speech transcripts with this additional information, as well as tighter coupling with downstream NLP systems, can lead towards performance on speech tasks that approach the results obtained when working with well formatted text.

1.2 Spoken Language Processing

Natural language processing research has developed many useful approaches for automatic processing of text. Extending these approaches to enable similar processing of speech will allow the same powerful techniques to be applied to the growing amount of available audio and video data.

While a simple first approach to spoken language processing might only apply speech recognition and then use the generated words as input to an NLP system, that approach ignores much of the information available in the acoustic signal. Further annotation of the speech recognition output can provide additional structure, such as sentence boundaries and punctuation, which are often useful to NLP tasks. Additionally, annotating the words with prosodic phrases and emphasis could provide additional structural cues to downstream tasks. In order to incorporate multiple sources of information, spoken language processing approaches typically involve a pipeline of component systems that work together to produce the final desired output.

For example, identifying names in an English news broadcast can be a very useful step in further understanding a news broadcast, but in order to perform high quality name detection, multiple processing stages are important: automatic speech recognition (ASR) to obtain the words, automatic punctuation prediction to annotate the words with structure, and finally named entity detection to identify which words are names. Each processing stage can generate errors; if the stages are just connected by simply passing on their single best output, cascading errors can lead to poor final performance. Tighter coupling across individual components can aid in alleviating the effect of compounding errors; an important aspect of this is communicating uncertainty between systems. Allowing the speech recognizer to pass on uncertainty in hypotheses can let the information ex-

traction weight the words it uses in detecting names in identifying relations. Also, passing on uncertainty in sentence segmentation can allow for downstream systems to optimize decisions on segmentation for the needs of the task.

Speech translation involves a similar pipeline approach which depends on speech recognition, sentence segmentation, and then finally machine translation. A baseline system would first generate a single best word sequence from the speech recognizer, predict sentence boundaries on that word sequence, and then translate to the target language. Again, errors from speech recognition and sentence segmentation both degrade performance of translation, so developing approaches that integrate the component systems can allow communication across the pipeline to reduce errors.

1.3 Contributions

The main goal of the dissertation is to develop general approaches for improving spoken language processing, investigated in the context of three main tasks: tagging, information extraction, and machine translation. While a possible approach is to simply run automatic speech recognition and then treat the task as if working with text, this ignores the possible errors introduced in recognition and also loses the additional information available in the acoustics of the speech. The two primary contributions of this dissertation aim to improve spoken language processing with:

- improved annotation of sentence and sub-sentence structure for speech, and
- tighter coupling between speech recognition, segmentation, and downstream applications

The contributions are evaluated for both sentence level and sub-sentence annotations.

At the sentence level, the key contribution in annotating structure is improving features for sentence segmentation models explicitly for machine translation. We investigate tighter coupling between ASR and sentence segmentation, finding that there is not a big benefit from considering ASR uncertainty in joint ASR and sentence segmentation. This study identifies a problem in bias of ASR confidence estimates, which we address with a new confidence estimation approach that extends to a multi-hypothesis representation. In addition we leverage our confidence estimation approach to develop a new ASR system combination approach that leads to reductions in word errors. We look at tighter coupling between ASR and downstream language processing by changing

the objective function for ASR to a parsing-based criteria that gives more weight to words that are syntactically important. Finally, we look at tighter coupling between segmentation and MT through optimizing the sentence segmentation threshold for translation performance.

Contributions in sub-sentence annotation are focused primarily on automatic comma prediction. We also investigate unsupervised clustering of words on acoustic features to discover prosodic structure, but performance for comma detection is more robust. We present results in Mandarin and English comma prediction, including representation of two types of commas for Mandarin, with findings that one type is sufficient for tagging and information extraction. We then investigate the use of automatic punctuation in downstream processing. Automatic comma annotation is incorporated into Mandarin part-of-speech tagging and named entity detection. We explore tighter coupling between punctuation and information extraction through optimizing sentence boundary and comma thresholds for entity and relation extraction on English. Finally, we investigate approaches for utilizing commas as soft boundaries in translation, developing a penalty for reorderings that cross comma boundaries and an approach that directly uses commas as features in phrase orientation modeling.

The impact of automatic sentence segmentation and comma prediction is assessed in multiple downstream tasks in order to evaluate across a range of possible uses and draw conclusions that could apply across more than one specific application. In particular, we investigate approaches that tune segmentation for the end application rather than as an independent segmentation model, which provide a simple but effective means of improving integration of segmental models. Our goal is to identify directions for improving automatic processing of speech, particularly with regard to sub-sentence segmentation, which is an aspect of speech processing that has not yet received significant attention.

1.4 Dissertation Overview

The dissertation is composed of two main parts, which follow a review of related background work presented in Chapter 2. The first part investigates sentence-level annotation and its impact on speech translation. Chapter 3 describes an approach for sentence segmentation that utilizes multiple recognition hypotheses. Chapter 4 develops methods for improving confidence estimation and then leverages the approach to develop a new method for ASR system combination. Chapter 5 describes

approaches for improving machine translation by with tighter coupling to sentence segmentation by optimizing sentence segmentation for translation. Then an approach for optimizing ASR for translation is also investigated with an alternative parsing-based optimization criterion.

The second part investigates sub-sentence structure and its impact on three downstream tasks. Chapter 6 describes an approach for automatic comma prediction and also explores unsupervised methods for detecting sub-sentence prosodic structure. Chapters 7 and 8 then assess automatic comma annotations and optimize punctuation for information extraction. Chapter 9 introduces two methods for using comma and parse information to improve reordering models. Finally, Chapter 10 concludes with a summary of the results and a discussion of future work.

Chapter 2

BACKGROUND

To provide context for the thesis research, this chapter reviews related work in segmenting speech in Section 2.1, and Section 2.2 describes automatic detection of prosodic structure. Section 2.3 describes spoken language processing tasks, specifically information extraction (IE) and machine translation (MT) for speech, the two tasks that we will consider in assessing the impact of segmentation. Then the implications of uncertainty in ASR and methods for coping with uncertainty in spoken language processing are discussed in Section 2.4, which motivate our experiments on integration of speech transcription and language processing.

2.1 *Segmenting Speech*

Typical speech recognition produces a string of text, but additional structure or segmentation is often desired by automatic processes (or humans) that are consumers of speech recognition output. In this section we describe possible types of segmentation, review prior work, and then describe the baseline models used in our research.

2.1.1 Types of Segmentation

Speech documents can be segmented at various granularities, ranging from small sub-sentence phrases up to whole stories or topics. Segmentation is generally taken for granted in language processing of text, because text typically has punctuation which naturally breaks the text into sentences and also has orthographic text mark up cues to paragraphs and document (story) boundaries. Spoken documents, such as news broadcasts, often have multiple stories or topics, so automatic segmentation can be an important task to provide coherent units for further processing. Given a single story or topic, approaches such as language model adaptation can be helpful for better automatic understanding. In retrieval tasks, automatic story segmentation may determine the unit that is returned to queries, or help users navigate large audio/video collections. Automatic speaker diarization is

another aspect of annotating spoken documents that conveys information about document structure, and what words are spoken by what speaker. Attributing words to speakers can significantly improve readability, and also allows analysis of speaker roles and contributions.

Sentence level segmentation is critical to most speech understanding applications, which often assume sentences as the basic unit of processing. Most speech understanding tasks were originally developed for text, and therefore assumed the presence of sentence boundaries. Even though the amount of spoken language material available is increasing, most spoken language processing models still train (at least in part) on textual data because there are often orders of magnitude more training data available from text sources. Therefore, automatic segmentation of speech is an important direction, and tied to reducing the mismatch between speech and text. While the convention for sentence boundaries are relatively clear for text, new conventions are required for capturing the characteristics of segmentation in spontaneous speech. Much of our work centers around sentence level segmentation, so we turn to a more detailed description of human annotation procedures for sentences.

Sentences in speech differ significantly from written text because spoken style often departs from written conventions, especially for conversational speech. A recent labeling strategy for speech data uses sentence-like units called SUs, for Sentential Units or sometimes Slash Units. An SU roughly corresponds to a sentence, except that SUs are for the most part defined as units that include only one independent main clause, and they may sometimes be incomplete (as when a speaker is interrupted and does not complete their sentence). A complete annotation guideline for SUs is available [157], which we refer to as the “V6” standard. While some corpora have been carefully labeled with the “V6” standard, it is much slower (and therefore costly) than alternative labeling strategies that have less clearly specified definitions of sentences. In order to reduce costs, many speech corpora are labeled by letting the transcribers use their own intuitive definition of sentences (more data can then be created at a lower cost, but with greater variability in sentence annotation). Our initial work on sentence segmentation focuses on detecting the SUs of the “V6” standard, but later work focuses more on the impact of sentence segmentation in spoken language processing (and less on actual sentence segmentation accuracy).

Another difference between spoken and written documents is that speech recognition does not typically produce sub-sentence punctuation. Automatic punctuation prediction, for instance sen-

tence boundaries and commas, can help in providing additional sentence structure and also lessen the differences between speech recognition output and textual sources. The following section describes approaches for automatic prediction of segmentation in speech.

2.1.2 Prior Work in Segmentation

Sentence segmentation generally predicts the most likely sequence of events, E , given a word sequence W and prosodic features, F , as in Equation 2.1.

$$\hat{E} = \underset{E}{\operatorname{argmax}} p(E|W, F) = \underset{E}{\operatorname{argmax}} p(E, W)p(F|W, E) \quad (2.1)$$

The modeling can be factored into two parts, as in Equation 2.1. The joint word and event sequence, $P(W, E)$, models events with lexical features (n-gram history), such as with the hidden event language model (HELM) of [155]. Prosody also provides important cues. Prosodic features capture pause, duration, pitch, energy, and various normalizations (including speaker normalization). The prosody component, $P(F|W, E)$ can be approximated by $P(F|E)$ because the prosodic features are mostly independent of the word identity. The normalized posteriors from a classifier on prosodic features $P(E|F)/P(E)$, such as a decision tree [19], is usually substituted for $P(F|E)$ by ignoring sequence dependencies in F . These two components are then combined, as in Equation 2.1, with an Hidden Markov Model (HMM) style approach to decoding [140]. Other early work in segmenting dialogs used prosodic features in neural nets to predict dialog units (which are similar to sentences) [74, 97].

Further modeling techniques have improved on the baseline approaches. The HELM framework was improved by interpolating multiple language models in [89]. Neural nets were used to predict sentence segmentation of Arabic speech in [149]. Both neural nets and finite-state models were investigated for prosodic and lexical feature combination in [28]. Approaches that directly model $P(E|W, F)$, such as Maximum Entropy models were explored in [55], and Conditional Random Fields (CRFs) were found to outperform other previous methods in [90]. Some of the best results so far have used a combination of multiple modeling approaches. Various approaches for combining results from multiple prosodic classifiers were investigated in [33], but CRFs alone are nearly as good [90].

Semi-supervised learning with co-training was investigated in [43]. They use the natural division of lexical and prosodic features to partition their feature space, and compare a baseline approach to two types of co-training and to self-training. They find that selecting training samples based on disagreement between the lexical and prosodic features was somewhat better than choosing based on agreement. Their final co-training model found improvements over the baseline for settings where only a small set of labeled data is available, but less so when more than 1,000 labeled points are available.

Punctuation For intra-sentence punctuation insertion, [12] use lexical information in the form of tri-gram language models. Decision trees with linguistically sophisticated features for enriching natural language generation output are used in [184], and they obtain better results than using just n-gram language models. The maximum entropy model of [55] also predicted commas with both lexical and acoustic features.

In [25], speech recognition and punctuation prediction were combined so that the recognizer generated punctuation based on acoustic and language model features. A similar approach was combined with a speech recognition system in [68], where the ASR system found improvement in punctuation prediction when rescoring using a decision tree for prosodic features, but it also introduced a small increase in word error rate.

2.1.3 Baseline Sentence Segmentation Models

The experiments of this dissertation build from various baseline sentence segmentation models, trained for different tasks. The three main configurations are described below.

SU segmentation for English CTS

The sentence segmentation approach for Conversational Telephone Speech (CTS) in Chapter 3 takes as input recognition alignments from either reference or recognized (1-best) words, and combines lexical and prosodic information using multiple models [84]. Prosodic features include about 100 features reflecting pause, duration, pitch, energy, and speaker change information. The prosody model is a decision tree classifier that generates the posterior probability of an SU boundary at each inter-word boundary given the prosodic features. Trees are trained from sampled training data in

order to make the model sensitive to features of the minority SU class using bagging techniques to reduce the variability due to a single tree. Language models include word and class n-grams. The prosody and language model are combined using an HMM. Additional approaches in this framework combine prosody tree predictions with language cues using other modeling frameworks, such as maxent models and CRFs.

ICSI+ multilingual sentence segmentation tools

Chapters 5 7, and 8 use the ICSI+ English, Mandarin, and Arabic sentence segmentation tools of [187] for broadcast news sentence boundary detection. As for the English CTS system, sentence boundary detection is treated as a binary classification problem, where every word boundary can be one of two classes: sentence boundary or non-sentence boundary. The classifier uses a combination of 5-gram HELMs and a boosting classifier [135] that combines speaker, prosodic, and lexical cues. The HELM classifier exploits lexical information and word sequence dependencies, whereas the boosting classifier exploits lexical cues (word trigrams) in combination with prosodic and speaker change information (extracted from the ICSI diarization system). The posterior estimates from the outputs of the two classifiers are interpolated using weights optimized on a held-out data set.

RWTH sentence segmentation for MT

The approaches to boundary detection described so far select boundaries by choosing only those positions for which the posterior probability of a boundary exceeds a certain threshold. This means that although the average segmentation granularity can be controlled, the length of a segment may take any value from one to several hundred words (unless some additional length heuristics are imposed). This may pose a problem for machine translation. Many statistical machine translation algorithms are either inefficient or not applicable if the length of the input sentence (in words) exceeds a certain threshold L (e.g. more than 60 words). Also, if a segment is too short (e.g. less than 3-4 words), important context information can be lost.

The RWTH sentence segmentation algorithm [102] was developed especially for the needs of machine translation. We utilize the approach in Chapter 5 in our comparison of segmentation approaches for MT. It also uses the concept of hidden events to represent the segment boundaries. A

decision to place a segment boundary is made based on a log-linear combination of language model and prosodic features, as modeled by Equation 2.2. In training, a downhill simplex algorithm is used to select the optimal λ_m s for sentence boundary prediction.

$$\hat{E} = \frac{\exp\left(\sum_{m=1}^M \lambda_m h_m(E, W)\right)}{\sum_{E'} \exp\left(\sum_{m=1}^M \lambda_m h_m(E', W)\right)} \quad (2.2)$$

with different features h_m (such as language model probabilities and pause lengths) and scaling factors λ_m for those features. In decoding, Equation 2.3 is used for the decision rule.

$$\hat{E} = \operatorname{argmax}_E \sum_{m=1}^M \lambda_m h_m(E, W) \quad (2.3)$$

In contrast to the ICSI+ approach, the RWTH approach optimizes over the length of each segment and adds an explicit segment length model (conditioned on the previous segment boundary) in the HMM-style search. Such an approach makes it possible to restrict the minimum and maximum length of a segment, while still producing meaningful SUs that pass all the relevant context information on to the phrase-based MT system. The minimum and maximum length of a segment can then be restricted (e. g. 4 and 60 words, respectively). As a result, the decision criterion and the HMM-style search has to be modified to include explicit optimization over the previous segment boundary.

The main features used by the algorithm are a 4-gram hidden-event LM, a normalized pause duration feature, and an explicit sentence length probability distribution learned from the training data. A segment penalty is also used to additionally control the segmentation granularity. The scaling factors in the log-linear combination of these and other models are tuned on a development set.

Other features can be included in the log-linear model. In particular, for a hypothesized boundary, the posterior probability from the ICSI+ model can be used as an additional feature to improve the RWTH approach.

2.2 *Prosodic Event Detection*

While prosody can be represented by continuous prosodic features (such as pauses, pitch, energy, duration, and speaking rate) which are used in the segmentation models described above, prosody can also be annotated symbolically as phrases and emphasis. Prosody conveys additional structure in speech beyond simple commas and periods. Instead of just annotating speech with commas and periods, a symbolic representation of acoustic cues from prosody can provide additional information about the structure of spoken communication.

The ToBI annotation standard for prosodic breaks and emphasis in English describes multiple prosodic phrase levels as well as tones in speech emphasis [145]. In the ToBI system there are five break levels (0-4), and a “p” diacritic to indicate disfluent boundaries. Break level 4 corresponds to an intonational phrase and break level 3 to an intermediate phrase. Another annotation strategy that more closely matches syntactic structure was developed for the VerbMobile project [11]. That system was also used to improve recognition and translation in a dialog system [39]. Prosodic boundaries are highly correlated with syntactic structure; in [36], at least 65% of automatically generated syntactic boundaries (from the chunk parser of [4]) coincided with prosodic boundaries. Incorporating prosodic features into a discriminative parse reranker improved parse accuracy in [64]. In [45], various levels of prosodic information are incorporated in speech recognition, and [96] use prosody in speech summarization.

2.2.1 *Models Used in Supervised Learning*

Approaches for detecting prosodic events typically train classifiers on prosodic labels from human annotators. There are three main tasks that have been studied: accent prediction, prosodic break prediction, and boundary tone recognition. While there are six possible accent tones in ToBI, some are infrequent so tone are often grouped in to a smaller number. For example, accent tone prediction can be approached as a four class problem: unaccented, high, low, and downstepped high. Supervised learning approaches achieve accuracies approaching 90% for tone classification on a single speaker in the Boston University Radio News corpus [119, 161, 9]. Prosodic break detection often groups annotated breaks into a smaller number of classes, such as: no prosodic break, minor break, major break, or disfluent break. Supervised learning approaches exceed 90% accuracy for a single speaker

in the Radio News corpus [176, 26] and achieve about 75% accuracy for a speaker-independent model [110] on the Switchboard corpus [120].

2.2.2 *Unsupervised Learning*

Human annotation of prosody is expensive, and therefore limited resources are available for prosodically labeled data. Models that require little or no hand labeled data are then important for modeling prosodic events. Recent unsupervised clustering approaches have found accuracies that approach that of supervised classification for emphasis [81], which uses k-means and asymmetric k-lines clustering into four classes. The collapsed two class problem of accent vs. no-accent and boundary vs. no-boundary can achieve even higher accuracies, with fuzzy k-means or Gaussian mixtures [8].

2.3 *Spoken Language Processing*

In general, spoken language processing adapts natural language processing applications for using speech input, rather than text. A wide range of tasks from information retrieval to summarization to machine translation are encompassed. Here we focus on information extraction (name and relation detection) and machine translation, because these are the tasks where we will evaluate the impact of automatic speech segmentation.

2.3.1 *Information Extraction*

Information Extraction (IE) aims at finding semantically defined entities in documents and characterizing relations between them. This is a critical step in better automatic understanding of documents. IE outputs are used as features in various tasks like machine translation, question answering, summarization, and information distillation [80].

Information extraction technology is benchmarked by the Automatic Content Extraction (ACE) series of evaluations, which include several separate tasks. *Entity Detection and Tracking*, which involves the identification of all entities in seven semantic classes (people, organizations, geo-political entities [locations with governments], other locations, facilities, vehicles, and weapons) which are mentioned in a document. In practice this involves finding mentions of entities (names, noun phrases, or pronouns), and then grouping mentions which refer to the same entity (co-reference).

Relation Detection and Characterization involves finding specified types of semantic relations between pairs of entities. For the 2004 evaluations, there were 7 types of relations and 23 subtypes, including a located-in relation, employment relations, a citizen-of relation, and a subsidiary-of relation. Entity and relation scores include weighted penalties for missing items, spurious items, and feature errors in corresponding items; details are given in the ACE 2004 Evaluation Plan.¹

The Nymble system from BBN developed an HMM approach for information extraction (IE) on text [16], and also applied it to speech [105]. Their approach is similar to the HELM segmentation model, choosing the maximum likelihood names sequence by modeling the joint sequence of words and names, as in:

$$\hat{E} = \operatorname{argmax}_{NE} p(NE|W) = \operatorname{argmax}_{NE} p(W|NE)p(NE) \quad (2.4)$$

Standard IE HMMs are extended and explicit error modeling for speech recognition is developed in [121]. Discriminatively trained language models for spoken language understanding are used to improve name detection in [54]. Investigations in [94] found that removing punctuation, in particular commas, has a significant negative impact on information extraction performance. They found that automatic sentence segmentation was almost as good as reference segmentation, but that removing other punctuation resulted in large performance drops.

NYU IE System Our experiments in Chapter 7 and Chapter 8, which investigate the impact of segmentation on IE, utilize the NYU IE system. Names are identified and classified using an HMM-based name tagger trained on several years of ACE data. Noun groups are identified using a maximum-entropy-based chunker trained on part of the Penn TreeBank and then semantically classified using statistics from the ACE training corpora. Coreference is rule based, with separate rules for name, nominal, and pronominal anaphors. For relation detection, a distance metric is defined based on each relation’s argument type and heads, as well as the words appearing between the arguments. Relations in new sentences are then identified using a nearest-neighbor procedure. The name model was trained on 800K words, the nominal classifier on 600K words, and the relation model on about 90K words of ACE training data. More details about the NYU IE system can be found in [42].

¹<http://www.itl.nist.gov/iaui/894.01/tests/ace/ace04/doc/ace04-evalplan-v7.pdf>

2.3.2 Machine Translation

Our experiments in Chapter 5 and Chapter 9 analyze the impact of sentence and sub-sentence segmentation on translation using the RWTH Aachen translation system. In statistical machine translation, a target language translation $e_1^I = e_1 \dots e_i \dots e_I$ for the source language sentence $f_1^J = f_1 \dots f_j \dots f_J$ is found by maximizing the posterior probability $Pr(e_1^I | f_1^J)$. This probability is modeled directly using a log-linear combination of several models. The best translation is found with the following decision rule:

$$\hat{e}_1^I = \operatorname{argmax}_{I, e_1^I} \left\{ \sum_{m=1}^M \lambda_m h_m(e_1^I, f_1^J) \right\} \quad (2.5)$$

The model scaling factors λ_m for the features h_m are trained with respect to the final translation quality measured by an error criterion [115]. The posterior probability $Pr(e_1^I | f_1^J)$ is modeled directly using a weighted log-linear combination of various models [116]: an n -gram language model, a phrase translation model [180], and a word-based translation model. The latter two models are used in both directions: $p(f|e)$ and $p(e|f)$. Additionally, a word penalty and phrase penalty can be included. Other features can include phrase count thresholds and a phrase reordering model [103].

MT quality is somewhat difficult to assess, but two error measures are prominently used. BLEU [122] is an n -gram overlap measure (with a length penalty). TER [147] is similar to WER in ASR, except that word phrases can be reordered with a cost of only one (rather than a cost equal to the length of the phrase, as would be the case in typical WER). In addition, because system segmentation does not usually match reference segmentation, we used the tool of [101] to determine the alignment with reference translations based on the word error rate and, using this alignment, to re-segment the translation output to match the number of reference segments.

Speech Translation Translation of speech incorporates additional areas of research that are not found in translation of text. The input to the translation system (i.e. the words and sentence structure) must be automatically recognized in speech. Previous research on speech translation has applied automatic speech recognition (ASR) to obtain the words, followed by techniques from machine translation for text. Many early speech-to-speech translation systems were dialog systems for constrained domains such as air travel planning or conference registration. Due to the nature of

the tasks, the systems worked on isolated utterances of just a few words ([5] [169], [107]). Later systems ([111], [109], [79], [160]), can deal with larger vocabulary speech input, but in general still focus on dialog tasks that have fairly constrained domains and relatively clear turn-taking that provides isolated utterances for translation.

Some more recent dialog systems [114], [75] developed frameworks for determining semantic and prosodic boundaries to provide segmentation for the translation engine. Their work predicted possible segmentations and then parsed over the hypothesized boundaries to determine the most effective segmentation, but the domain was still a task-driven, turn-based dialog with relatively defined segments.

However, many additional interesting translation domains do not have isolated utterances (i.e. broadcast news), and they also have a much less constrained dialog (i.e. not a specific task-driven interaction). Without clear notions of a sentence to provide to the translation engine, additional research is required to determine structure that can be helpful for translation. A study on Arabic-to-English broadcast news translation found better translation quality when automatic segmentation was tuned to produce longer sentences, emphasizing the importance of optimizing sentence segmentation directly for translation performance [98]. Speech translation for European Parliamentary meetings is described in [158], and the effect of WER on MT quality is measured. They find that WER has a roughly linear relationship to both BLEU and TER. A study of English-to-Iraqi, and Iraqi-to-English translation found a piece-wise linear relationship between WER and BLEU (with only small improvements in translation after a critical level of WER performance is achieved) [134]. They also found that human judgments were correlated to BLEU, human judgments were more sensitive to changes in WER.

Punctuation for MT Besides producing correct words in translation, systems should also produce correctly punctuated output. Better translation performance is achieved in [102, 77] for speech translation when target language punctuation is inserted by the translation system, which is accomplished by training with no punctuation on the source side and reference punctuation on the target side. That approach outperforms alternatives that predict punctuation on the source or target side independently.

2.4 *Representing Uncertainty in Spoken Language Processing*

Most spoken language processing systems involve a pipeline of multiple components, where each component makes errorful automatic predictions. When errors are made early on in the pipeline, it is often difficult for later stages to recover, especially when only simple hard decisions are passed from stage to stage. For many problems, the individual modules in the pipeline are themselves so complex, that it is not practical to combine them for joint decision making. One approach for mitigating the effect of errors in early processing components is to allow for communicating uncertainty in predictions. Speech recognition can annotate words with confidences so that low confidence words can be treated accordingly, and N-Best lists or lattices can pass on multiple recognition hypothesis rather than just the single best hypothesis. Likewise, confidences can be provided for predicted structure, or multiple structure hypotheses.

This section describes approaches for representing multiple ASR hypotheses, determining ASR word confidences, and combining hypotheses from multiple ASR systems. Finally, approaches for handling uncertainty in spoken language processing are reviewed.

2.4.1 *Representing Multiple ASR hypotheses*

Most speech recognition systems are designed to select the best word hypothesis from a larger hypothesis space that contains multiple recognition hypotheses. Various representation approaches have been developed that express increasing richness. N-best lists simply enumerate multiple possible recognition outputs per segment and typically are highly redundant because usually only a few words differ between hypotheses. Word lattices are a more concise representation that can represent multiple paths. A degenerate case could actually encode an N-best list, if no path interacts with another path. A more typical word lattice would only have each unique word (associated with a particular time range) present once (or somewhat duplicated if the lattice is expanded for unique n-gram context). Word graphs are a less rich lattice-like representation that do not encode word times, but only possible word sequences. A confusion network [95] is a simplified and compacted representation of a word lattice or N-best list, where the complexity of the lattice or list representation is reduced to a form that maintains all possible paths (and more), transforming the space to a series of slots that each have word hypotheses (and null arcs) and associated posterior probabilities. Where a

lattice can represent independent multi-word paths, confusion networks reduce the expressive power such that there is always a path from any word to any adjacent word. An intermediate approach simplifies a full lattice to a “pinched lattice” by forcing all lattice paths through certain nodes, thereby limiting the length of independent paths [34]. A confusion network is a type of pinched lattice that has been “pinched” at every node.

Each representation also incorporates component scores. Lattices and n-best lists typically store multiple types of scores per word, such as acoustic, language model, and pronunciation scores, as well as a combined posterior. Confusion networks usually combine scores into a single posterior probability that represents the system confidence for each word.

2.4.2 ASR Word Confidences

Usually the word posteriors resulting from the recognition process are over confident, in part because of modeling error and in part because (in order to be computationally tractable) recognizers prune the set of hypotheses that are considered and the word likelihoods are thus normalized by some subset of the hypothesis space (rather than the total hypothesis space). When the pruned hypothesis space is richer, then better posteriors are obtained, as illustrated by the improvements from using word graphs rather than N-best lists in [173], where confidences are based on frame-level analysis, rather than word level. To account for bias in word confidence, previous work has trained models to warp the confidence of the top word hypothesis to obtain a more accurate measure, using secondary predictors such as neural networks [171], generalized linear models [146], and in [162] a comparison of boosting, SVMs, and decision trees. Another approach directly optimizes the ASR acoustic and language model scores for improving word confidences [148]. Alternatively, the use of a background model to cover the pruned space is proposed in [82]. In all cases the models are trained using a target of one when a training instance is the correct word and a target of zero when a training instance is an incorrect word. Confidence prediction is usually evaluated in terms of normalized cross entropy (as described in Section 4.1.3), which has the advantage of providing a single measure of performance but obscures the bias in the estimates. Reducing word confidence bias should provide downstream processing with better estimates to assess the usefulness of particular words in further modeling.

2.4.3 ASR System Combination

State-of-the-art speech recognition systems usually combine multiple different recognition outputs in order to obtain an improved overall result. Previous work in speech recognition system combination has produced significant improvements over the results possible with just a single system. The most popular, and often best performing method is ROVER [38], which selects the word that the most systems agree on at a particular location (majority voting). An extended version of ROVER also weights system votes by the word confidence produced by the system (confidence voting).

Further improvements have been achieved by including multiple system alternatives, with methods such as Confusion Network Combination (CNC) [35], or N-Best ROVER [152], which is a special case of CNC. Alternatively, the combination can be performed at the frame level (min-fWER) [53].

Another system combination approach [183] used two stages of neural networks to select a system at each word, with features that capture word frequency, posteriors at the frame, word, and utterance level, LM back-off mode, and system accuracy. They obtained consistent but small improvements over ROVER: between 0.7 and 1.7% relative gains for systems with about 30% WER. In other related work, a language model was used to break tie votes and improve ROVER in [136].

Recent work found that which system combination method works best depends on the number of systems being combined [52]. When only two systems are available, approaches considering multiple alternatives per system were better, but as the number of systems increased, the standard ROVER with confidence scores was more robust and sometimes even better than CNC or min-fWER combination.

2.4.4 Uncertainty in Spoken Language Processing

Approaches that communicate uncertainty from ASR or segmentation to spoken language processing have been shown to improve results over simpler techniques that make hard decisions. The remainder of the section reviews related work in uncertainty and spoken language processing that motivates the research of the dissertation with regard to tighter integration of ASR, segmentation, and spoken language processing.

ASR Uncertainty Speech recognition can be improved by selecting hypotheses from an N-best list using discriminative reranking techniques based on n-gram features [132] or syntactic features [31, 104]. Rescoring speech recognition lattices with a parsing LM also gave WER reductions in [23]. This work motivates our approach of leveraging multiple ASR hypotheses to improve sentence segmentation.

Accounting for uncertainty in speech recognition output has provided improvements in a variety of subsequent language processing tasks. Gains have been obtained in spoken document retrieval by using lattices (or simplified lattices), rather than just 1-best transcriptions [133, 24, 27]. Likewise, parsing results are improved on speech if multiple hypotheses are considered [63], and also as in Mandarin name detection when using an N-best list [182]. Improvements have been found for broadcast news speech translation when using N-Best lists and lattices [6, 100]. The MOSES toolkit has also implemented translation for confusion networks [72], which has been used in [138, 14]. The fact that multiple word hypotheses are helpful argues for improving the confidences of those hypotheses and also representing multiple segmentation hypotheses.

Segmentation Uncertainty Accounting for uncertainty in segmentation has given better performance in speech recognition, parsing, and speech translation. In [150], uncertainty in segmentation was used in language model rescoring in order to have sentence-like units (matching the training data) rather than the pause-based units typically used in speech recognition. Small reductions in WER were achieved. The effect of automatic segmentation on parsing was shown in [65], and parsing improvements were found when reranking over multiple possible segmentations in [44]. Previous work on communicating segmentation uncertainty has also led to improvements in speech-to-speech translation systems. The work of [108] explored incremental parses to determine the appropriate dialog act segmentation, but no prosodic cues were included. A similar approach searched multiple possible parses and segmentations of a spoken utterance and included prosodic cues [114].

A primary new direction for our work is the departure from tasks which typically have clear boundaries (i.e. speaker turns) at fairly short intervals. These constraints have allowed previous work to focus on searching many possible segmentations over reasonably short word streams (an average of 25 in one of the longer cases). Continuous speech domains, such as broadcast news, will have much longer continuous speech portions without clear break points. Searches over all possible

parses and segmentations (as in most previous work) would become computationally intractable for such long pieces of continuous speech.

Chapter 3

ACCOUNTING FOR ASR UNCERTAINTY IN SENTENCE SEGMENTATION

Motivated by previous work that found improvements in ASR and spoken language processing by leveraging multiple ASR hypotheses, in this chapter we develop approaches for integrating ASR and sentence segmentation. A key difficulty in processing automatic speech recognition output is the uncertainty of the hypothesized word strings. Recognition output is typically a single best hypothesis that contains transcriptions errors. These word errors contribute false information to further processing, leading to higher error rates for language processing of speech, when compared to text. In addition, word errors lead to higher error rates for sentence segmentation, which can compound the problems for language processing. When the performance of the speech recognition system is constant, one possibility for addressing the problem of word errors is to use information about hypothesis uncertainty from the recognizer. In this chapter, we extend existing methods for automatic sentence boundary detection by leveraging multiple recognizer hypotheses in order to provide robustness to speech recognition errors.

Multiple hypotheses are helpful because the single best recognizer output still has many errors even for state-of-the-art systems. For conversational telephone speech (CTS) word error rates can be from 15-20%. These errors limit the effectiveness of sentence boundary prediction because they introduce incorrect words to the word stream. Comparing the performance of reference and ASR based sentence segmentation over a range of operating points via a decision-error trade-off (DET) curve (Figure 3.1) shows the generally increased miss and false alarm probability when working with recognition output for English CTS. In addition, the miss probability asymptotically approaches 10%, because some SUs cannot be detected when all the words in a short SU are ASR deletion errors. The optimal point for the system on reference words has 26.7% error (19% deletions) and the system on 1-best ASR words has 41.2% error (with 26% deletions).

Results from other evaluations have shown that sentence boundary detection error rates on a baseline system increased by 50% relative for CTS when moving from the reference to the ASR

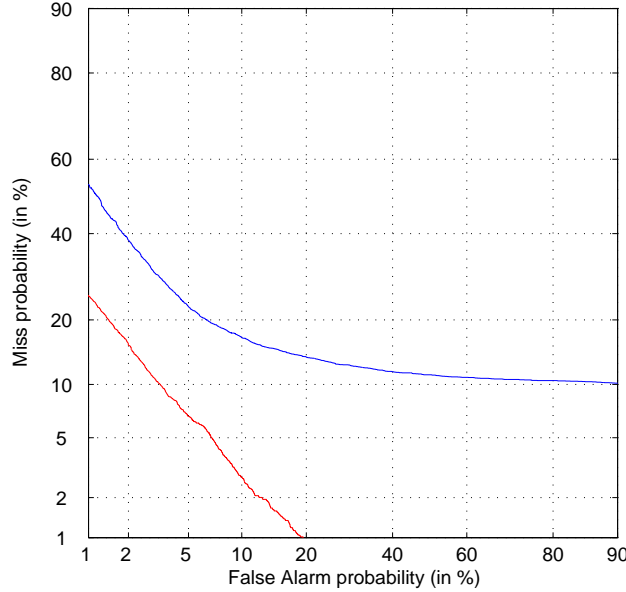


Figure 3.1: DET curve illustrating SU detection error trade-offs for 1-best (blue line) vs. reference (red line) decoding for English CTS.

condition [83]. The large increases in SU detection error rate in moving from reference to recognizer transcripts motivates an approach that reduces the mistakes introduced by word recognition errors. Although the best recognizer output is optimized to reduce word error rate, multiple word hypotheses may together reinforce alternative (more accurate) SU predictions. Or, there may be multiple ASR hypotheses with the same word error rate but leading to different sentence segmentations, some better than others. Thus, we investigate two approaches to improving sentence segmentation: combining the sentence segmentation predictions without changing the 1-best word hypothesis, and joint decoding of words and sentence segmentation in the recognition lattice.

The baseline approach, as described in Section 2.1.2, models sentence segmentation over just the first-best word hypothesis, W , as in:

$$\hat{E} = \operatorname{argmax}_E p(E|W, F) \quad (3.1)$$

Our first proposed approach chooses sentence segmentation by combining across multiple word hypotheses, W , weighted by the probability, $p(W|x)$, of each word sequence given the acoustics,

x , as provided by the speech recognizer, as in:

$$\hat{E} = \operatorname{argmax}_E \sum_W p(E|W, F)p(W|x) \quad (3.2)$$

Our second approach selects the joint sentence segmentation and word sequence, incorporating both the segmentation model, $p(E|W, F)$, and the speech recognition model $P(W|x)$, as in:

$$\hat{E}, \hat{W} = \operatorname{argmax}_{E, W} p(E|W, F)p(W|x) \quad (3.3)$$

Our sentence segmentation methods build on a baseline system and task domain reviewed in Section 3.1. Section 3.2 describes our approach and experiments for predictions based on multiple recognizer hypotheses using confusion networks, and Section 3.3 develops and tests a joint word and SU decoding strategy. Finally, Section 3.4 summarizes our conclusions.

This chapter is joint work with Yang Liu at ICSI, Andreas Stolcke and Elizabeth Shriberg at SRI/ICSI, Mari Ostendorf at University of Washington, and Mary Harper at University of Maryland. Confusion network combination for SU prediction was published at HLT 2004 [51], and joint word and SU decoding was published in the September 2006 volume of the IEEE Transactions on Audio, Speech and Language Processing [88].

3.1 *Experimental Paradigm*

In this work, we focus only on detecting SUs (as defined by the LDC “V6” standard) and do not differentiate among the different sentence subtypes (e.g. statement, question, etc.) that were labeled in human annotation. The next sections describe our baseline system, corpora, and evaluation framework.

3.1.1 *Baseline System*

The automatic speech recognition system used was an updated version of that used by SRI in the Fall 2004 RT evaluations [113], with a WER of 18.6% on the English Conversational Telephone Speech (CTS) evaluation test set. The system performs multiple recognition and adaptation passes and eventually produces up to 2000-best hypotheses per waveform segment, which are then rescored with a number of knowledge sources, such as higher-order language models, pronunciation scores, and duration models. For best results, the system combines decoding output from multiple front

ends, each producing a separate N-best list. All N-best lists for the same waveform segment are then combined into a single word confusion network [95] from which the hypothesis with lowest expected word error is extracted. In our baseline SU system, the single best word stream thus obtained is then used as the basis for SU detection.

Our baseline SU system builds on work in sentence boundary detection using lexical and prosodic features [84], as described in Section 2.1.3. For this work we include the HMM and maxent model, but not the CRF.

3.1.2 Corpora

The conversational telephone speech was collected by LDC, who recorded conversations about a pre-specified topic between pairs of speakers who were randomly selected and typically not acquainted with each other. Conversations lasted about five minutes, and were recorded at sampling rate of 8kHz. Conversations labeled with “V6” style SUs (as described in Section 2.1) provide 40 hours of training, 6 hours of development, and 3 hours evaluation test data drawn from both the Switchboard and Fisher conversational telephone speech (CTS) corpora. Additional training data is available from 2003, but the human annotators used an older guideline (“V5”), which had a significantly different definition of SUs, so that data is not used in training for our system.

The system is evaluated for CTS using training, development and test data annotated according to the “V6” standard. The test data is that used in the DARPA Rich Transcription (RT) Fall 2004 evaluations, which has about 5000 SUs.

3.1.3 Evaluation

Errors are measured by a slot error rate similar to the WER metric utilized by the speech recognition community, i.e. dividing the total number of inserted and deleted SUs by the total number of reference SUs. (Substitution errors are included only when subtype is scored, and our focus is on simple SU detection because subtype is determined in a subsequent detection stage, i.e. after the N-best SU detection.) When recognition output is used, the words generally do not align perfectly with the reference transcription and hence the SU boundary predictions will require some alignment procedure to match to the reference transcript. Here, the alignment is based on the minimum word

error alignment of the reference and hypothesized word strings, and the minimum SU error alignment if the WER is equal for multiple alignments. We report numbers computed with the md-eval (v18) scoring tool from NIST, as well as provide DET curves showing error trade-offs.

3.2 *Weighted SU Prediction with N-Best Sentence Hypotheses*

This section describes our experiments in sentence segmentation with multiple ASR hypotheses. First we develop our approach then present our experimental results.

3.2.1 Approach

We first describe feature extraction for multiple ASR hypotheses, then SU detection for each hypothesis, and finally hypothesis combination.

Feature Extraction

Prediction of SUs using multiple hypotheses requires prosodic feature extraction for each hypothesis. This in turn requires a forced alignment of each hypothesis to obtain phone times (because our N-Best lists do not include timing information). Thousands of hypotheses are output by the recognizer, but we prune to a smaller set to reduce the cost of running forced alignments and prosodic feature extraction. The recognizer outputs an N-best list of hypotheses and assigns a posterior probability to each hypothesis, which is normalized to sum to 1 over all hypotheses. We collect hypotheses from the N-best list for each acoustic segment up to 98% of the posterior mass (or to a maximum count of 1500), which gives essentially the same WER when compared to the full unpruned N-Best list.

Next, forced alignment and prosodic feature extraction are run for all segments in this pruned set of hypotheses. Statistics for prosodic feature normalization (such as speaker and turn F0 mean) are collected from the single best hypothesis. For lexical features that span the boundaries of the recognizer segmentation there is the potential to consider all of the adjacent N-best hypotheses. Since this quickly becomes very costly, we extend the current hypothesis to the left and right using the 1-best hypothesis from neighboring segments.

SU Detection for Individual Hypotheses

After obtaining the prosodic and lexical features, the HMM and maxent systems predict sentence boundaries for each word sequence hypothesis independently. For each hypothesis, an SU prediction is made at all word boundaries, resulting in a posterior probability from each classifier for SU and no_SU at each boundary. The same models are used as in the 1-best predictions – no parameters were re-optimized for the N-best framework. Given independent predictions for the individual hypotheses, we then build a system to incorporate the multiple predictions into a single hypothesis, as described next.

Combining Hypotheses via Confusion Networks

The prediction results for an individual hypothesis are represented in a confusion network that consists of a series of word slots, each followed by a slot with SU and no_SU arcs, as shown in Figure 3.2. (This representation is a somewhat unusual form because the word slots have only a single hypothesis.) The words in the individual hypotheses have probability one, and each arc with an SU or no_SU token has a confidence (posterior probability) assigned from the HMM or maxent model. The overall network has a score associated with its N-best hypothesis-level posterior probability, scaled by a weight corresponding to the goodness of the ASR system that generated that hypothesis.

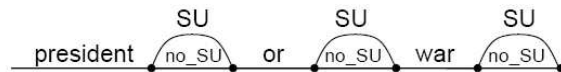


Figure 3.2: Confusion network for a single hypothesis.

The confusion networks for each hypothesis are then merged with the SRI Language Modeling Toolkit [151] to create a single confusion network for an overall hypothesis. This confusion network is derived from an alignment of the confusion networks of each individual hypothesis, where the alignment is constrained so that the SU events and words cannot align to each other. The resulting network, as shown in Figure 3.3, contains two types of slots. Slots with word hypotheses from the N-best list and slots with the SU/no_SU arcs (but with new confidence estimates combined across the hypotheses in the N-Best list). The confidences assigned to each token in the new confusion network

are a weighted linear combination of the probabilities from individual hypotheses that align to each other, compiled from the entire hypothesis list, where the weights are the scaled hypothesis-level posteriors.

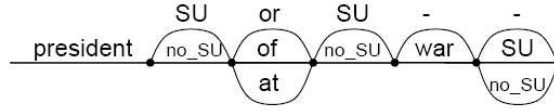


Figure 3.3: Confusion network for a merged hypothesis.

A combined confusion network is produced for each model type: HMM and maxent. Then, to obtain a final confusion network, the same alignment and combination is applied across the results from each modeling approach. The HMM and maxent confusion networks are each given equal weight for this final combination.

Finally, the best decision in each confusion network slot is selected by choosing the words and boundary events with the highest probability. Here, the words and SUs are selected independently, so that we obtain the same words as would be selected without inserting the SU tokens (and therefore guarantee no degradation in WER). The key improvement is that the SU detection is now a result of detection across all recognizer hypotheses, which reduces the impact of word errors from the top hypothesis.

3.2.2 Experiments

Table 3.1 shows the results in terms of slot error rate on the RT04 CTS Evaluation Set. The middle column indicates the performance on a single hypothesis, with the words derived from the pruned set of N-best hypotheses. The right column indicates the performance of the system using multiple hypotheses merged with confusion networks. There is an improvement of 0.7% absolute in the SU detection score, which is significant at $p < .08$ using a matched pair test on segments defined by large pauses [49]. This improvement also translates directly to an improvement in the subtype classification score, i.e. when substitution errors are included the error rate is 52.2% for the pruned 1-best system and 51.5% for the confusion network system. However, examining the performance of the two systems over a range of operating points via a decision-error trade-off (DET) curve

(Figure 3.4) shows that there is not a consistent advantage to the N-best system.

Table 3.1: SU error rates for single best vs. confusion nets using N-best list pruning.

	SU error rate	
	Single Best	Conf. Nets
CTS HMM	42.0%	41.9%
CTS Maxent	42.4%	43.2%
CTS HMM+Maxent	41.2%	40.5%

The small improvements obtained here are in contrast to earlier work [51], which showed somewhat greater benefit with the use of confusion networks. There have been some changes to the task definition (V5 vs. V6 specification) and there are always statistical variations across different test sets, but more importantly the SU detection system and ASR system improved. One other possible reason for the smaller benefit in the current work is that the N-best framework cannot take advantage of a new turn labeling strategy used in the 1-best system (because the turn based features were generated based on the word times of the 1-best hypothesis). We also note that the introduction of DET-based error reporting [49] allows us to see that there may be greater or lesser advantages at different operating points.

Though gains were somewhat larger in our previous work [51], they were still relatively small. At that time, we hypothesized that N-best pruning was limiting performance, but here we find that the WER is similar with and without pruning so that cannot explain the negative results. Since improvement in WER clearly has a significant impact on SU detection, and since the N-best oracle error rate is significantly better than the 1-best WER (roughly 30-40% lower), we hypothesize that either the MDE confidence needs to be more tightly linked to the word hypothesis (such as the joint word/SU detection of the next section) and/or that improved word confidence estimates are needed.

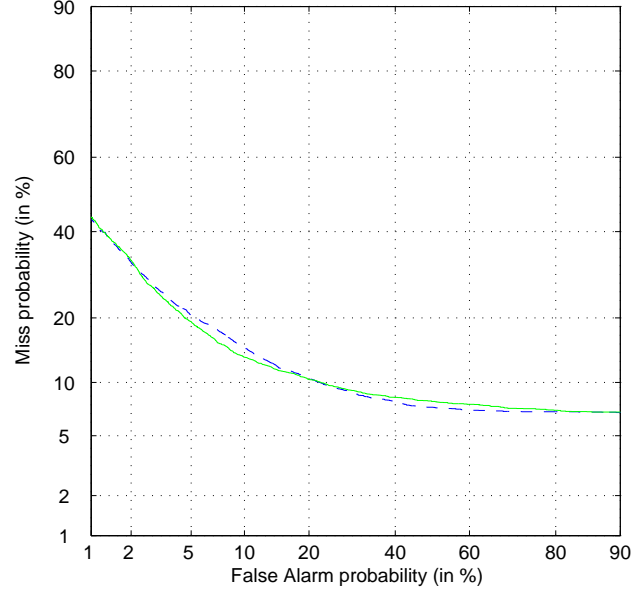


Figure 3.4: DET curve illustrating SU detection error trade-offs for pruned 1-best (solid line) vs. confusion network (dashed line) decoding.

3.3 Joint Word-SU Lattice Decoding

To consider a larger number of word hypotheses and with the hope of positively impacting word recognition performance, we also investigated lattice-based joint decoding of words and SUs. The next sections describe our approach and experimental results.

3.3.1 Approach

To simplify the implementation, only the HMM modeling approach is used. First, prosodic features are extracted over all words in the lattice, and then prosodic posteriors from the decision trees are attached to each possible boundary position (after each word). The probabilities for each of the SU LMs in the HMM model are also associated with each arc in the lattice. Running the forward-backward algorithm on the lattice, with the same score combination weights as in independent recognition and SU decoding, we obtain SU and non-SU posteriors for each word boundary in the lattice.

The resulting log posteriors are used in joint word and SU decoding, with a tunable weight for the SU scores. The use of SU posteriors (that sum to one) rather than the separate SU-related scores is important for ensuring that word posteriors (as computed by the forward-backward algorithm) are equivalent to the posteriors from the lattice without inserted SU events.

3.3.2 Experiments

The plot in Figure 3.5 illustrates the trade-off in SU error and WER as the weight on the SU posterior is varied in ASR decoding, for the CTS task. As the weight of the SU score is increased relative to the other ASR scores (e.g., acoustic, LM, pronunciation), the SU error decreases, but it eventually causes a degradation in WER. Unfortunately, only small gains are obtained before causing a degradation in WER. To better understand this result, we analyzed confidence estimates for SUs and words to determine whether the SU detection scores are causing problems. As it turned out, the SU posteriors are relatively good estimates of confidence of an SU, but the word confidences are optimistically biased and overestimate the probability that a word hypothesis is correct. We conjecture that this bias could be one reason for the lack of benefit to SU performance from considering different recognition hypotheses. The fact that SUs provide no benefit to recognition performance may be due to their relatively high error compared to recognition error rate, but it may also be the case that a more sophisticated integration of SUs and language modeling is needed.

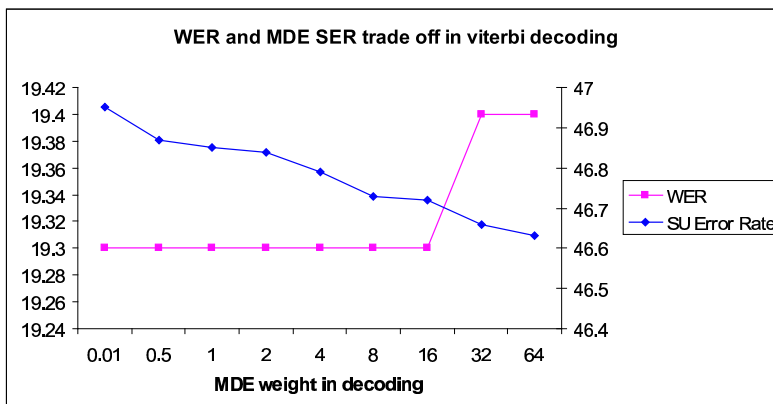


Figure 3.5: The trade-off in SU error rate and WER in lattice decoding on CTS.

3.4 Conclusions

Detecting sentence structure in automatic speech recognition provides important information for automatic language processing and human understanding. Incorporating multiple hypotheses from word recognition output can improve overall detection of SUs in comparison to prediction based on a single hypothesis, although only small gains have been realized so far. In addition, including segmentation in the lattice and jointly decoding words with sentence boundaries found only small changes. We conclude that approaches for improving word confidence estimates and reducing word error rate will provide the most direct path to further improvements in sentence boundary detection, and will also have a direct impact on subsequent language processing. Hence, the next chapter focuses on these problems.

Chapter 4

ASR UNCERTAINTY AND SYSTEM COMBINATION

Chapter 3 found that while incorporating multiple recognition hypotheses in sentence segmentation prediction produced some improvements, gains were relatively small in comparison to the degradation caused by moving from reference to ASR transcripts (where error can increase by 50% relative). In this chapter we first develop an approach for improving word confidence estimates, and then leverage that work to improve ASR system combination and reduce WER.

Section 4.1 describes our new method for reducing bias in confusion network (CN) confidence estimates. We find significantly improved confidences, but limited impact on further processing with multiple hypotheses. Then, instead of attempting to improve word confidences for multiple recognition hypotheses, in Section 4.2 an ASR system combination approach is described that directly reduces errors in the 1-best ASR output. The approach learns a classifier that selects system words based on features generated from ASR lattices and cross system comparisons. The result achieves as good or lower word error rate on our corpus than any previously published system combination method.

The word confidence estimation research is joint work with Mari Ostendorf, and was published at ICASSP 2006 [50]. The ASR system combination research is joint work with Björn Hoffmeister, Ralf Schüller, and Hermann Ney at RWTH Aachen in Germany, and Mari Ostendorf at University of Washington. It was published at HLT 2007 [46].

4.1 *Compensating for Word Confidence Estimation Bias*

This section looks at the problem of confidence estimation at the word network level, where multiple hypotheses from a recognizer are represented in a confusion network. Given features of the network, a support vector machine (SVM) is used to estimate the probability that the correct word is missing from a candidate slot and then other word probabilities are normalized accordingly. The result is a reduction in overall bias of the estimated word posteriors and an improvement in the confidence

estimate for the top word hypothesis in particular.

This work departs from previous work by addressing the issue of bias for multiple word hypotheses generated by the recognizer, not just the top hypothesis. By improving the confidence estimate for all word hypotheses we improve the ability of downstream systems to make use of alternative words and have accurate confidence measures for those words. Our method predicts the probability that the recognizer did not hypothesize the correct word under a particular search pruning condition, and uses this predicted probability to adjust the hypothesized word posteriors. The approach for predicting missed words is similar to the standard approach (which uses a secondary classifier) for predicting when a word is correct, where the difference is that we ask “is the correct word in the list” instead of “is the top word correct”? Experimental results show that this simple predictor reduces the problem of bias in word posterior estimates for the whole network and improves the 1-best confidence estimate significantly as well.

Our approach builds on the confusion network representation of word uncertainty, which is described in Section 2.4.1. Section 4.1.1 illustrates the problem of bias in confusion networks. The method for predicting the missed word probability and thereby adjusting the network posteriors is described in Section 4.1.2. The experimental paradigm and results are described in Section 4.1.3 and 4.1.4, respectively. Contributions and open questions are summarized in Section 4.1.5.

4.1.1 Bias in Confusion Network Posteriors

This work assumes a confusion network representation of recognizer uncertainty. The posterior probabilities of all hypotheses in a slot (including the null arc, if present) are chosen such that they sum to one. This effectively assigns zero probability to the event that the word is not in the lattice or N-best list, which can contribute to (in addition to general modeling error) an optimistic bias of the word posteriors when the recognition hypothesis space is pruned (and therefore confidences are normalized by an smaller, incomplete, probability space).

To illustrate the problem of bias, Figure 4.1 shows a plot of the relative frequency that hypothesized words are correct as a function of their predicted confidence, using data from a conversational speech recognition task. The relative frequencies are computed by binning over different confidence intervals. The distance of the curve from the diagonal line reflects the bias of the estimate. Where

the curve falls below the diagonal, the estimates are “over confident”, e.g. words predicted with a posterior of 0.8 are correct in less than 60% of the cases. Similarly, when the curve is above the diagonal, as for the low posterior cases, the estimate is lower than it should be. This example shows that the confusion network estimates are over confident for the majority of cases, since there are fewer instances at the low confidence regions. As explained above, this bias is due in part to the practice of building a confusion network from a pruned lattice, where the probability mass of alternative hypotheses that are not in the pruned lattice are not accounted for in the confusion network normalization.

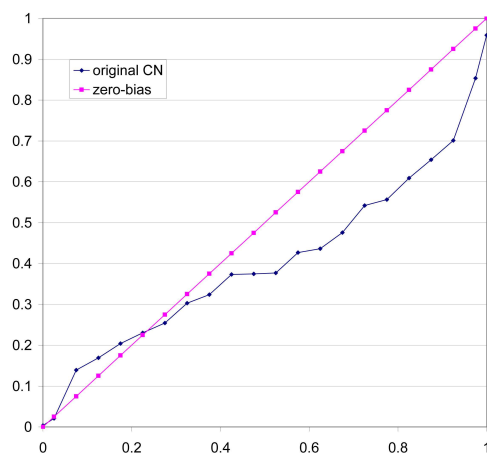


Figure 4.1: Relative frequency that a hypothesized word is correct as a function of the predicted posterior in a CN.

4.1.2 Network Posterior Adjustment

An obvious solution to the problem of over-confidence is to introduce an entry in each slot to account for the event that the correct word is not in the list. The probability of a missed word is predicted independently at each slot in a manner somewhat similar to confidence prediction techniques that combine recognizer posteriors with other features. (An advantage of this approach over using a background model is that the additional features can provide cues beyond what is captured in the recognizer acoustic and language models.) Then, the word and null-arc probabilities are simply renormalized to account for the added probability mass. Thus, the simple prediction of a series of

probabilities of missed words has the effect of adjusting the posteriors of the full network. Also, the addition of a missed word probability may be useful for other tasks, such as OOV detection.

Many different approaches are possible for predicting the missed word probability. Details regarding the features and prediction models used in the experiments conducted here are described next. In order to extract features and train the model, we align reference transcriptions with CNs for a collection of recognizer test data. A separate recognizer test set is used for evaluating the prediction and its impact on the CN posteriors as a whole.

Features

As the “baseline” set, we adopt the features used in previous work [171], which include: the length-normalized position of the word in the sentence; the log length of the sentence; two Boolean features indicating whether the adjacent slot (to the left and right) has the null arc as the most probable word; the posterior probability of the mostly likely word in the current, left and right slots; and the uni-gram word probability from the recognizer language model. In the “extended” set, we include additional feature types: the mean and variance of the posteriors in the current slot, length (in characters) of the top word, and the Boolean features from above extended to the slots two to the left and right of the current slot. In addition, we explore also adding the lexical identity of the top word in the current slot, referred to as the “full” feature set. Boolean features are added for each word, either for the top 1000 most common words, or for all words in the dictionary.

Models

Given a pruned list of words with associated posterior probabilities from a first pass ASR system (and associated features as described above) the problem is to predict the posterior probability of the binary event: whether or not the list is missing the correct word. The prediction could be based on a statistical binary classifier or a regression model. This posterior is a rough estimate of the total probability of the pruned words in the region of the slot. Initially, we trained a neural network with the same approach as used in previous work on confidence prediction [171]. However, although the neural network approach is effective for improving the confidence estimate of the top hypothesis, its performance was poor in predicting missing word probabilities. We then adopted SVM regression

as an approach that was more flexible (i.e. allowed easily for a larger feature set, such as the lexical word feature). The SVM with a linear kernel was not a successful approach, but a Gaussian kernel improved results to a reasonable level.

In both cases, the models are trained by using a target of 1 when the reference word is not present in the slot hypothesis list, and 0 when it is. In our training set, 2,774 of the 34,898 (8%) slots are missing the reference word (and 10%, or 6,535 of the 62,361 testing slots). We use the *SVM^{light}* tools for SVM regression [61]. SVMs have also been used with success in previous work on word error detection [186], which is an alternative to confidence prediction.

Analysis of preliminary experiments indicated some regions that were not being handled well by the classifiers. An SVM trained on all words performed poorly for words that received very high recognizer posteriors (.999 and greater). Since these comprised more than half of the slots and the actual accuracy for these words was .95, a simple heuristic solution imposed a minimum probability of miss equal to .05. This provided improved results, but did not completely address the issue. An alternative solution was to train two classifiers, one for slots with confidence greater than .95, and one for those less than .95 (before clipping). As shown in the experiments, combining these approaches gave the best results.

4.1.3 Experimental Paradigm

Experiments are conducted for conversational telephone speech recognition task, as described in Section 3.1.2. The recognizer used for our experiments is the SRI 20 times real time Decipher system with small modifications from the system used for the 2004 DARPA evaluations. This is a state-of-the-art system, which combines three systems (MFCC cross-word, MFCC non-cross-word, and PLP cross-word) with cross adaptation and a final ROVER step. The data sets used for training and testing the missed word probability predictor are the from the NIST RT evaluations. Training is performed on the development test set from 2004 (as in Chapter 3), and testing on the evaluation set from 2003.

The results are evaluated in three ways, associated with the different tasks that might benefit from this approach. First, to evaluate the impact of posterior correction over the whole CN, we present a plot of the word accuracy versus predicted confidence that reflects the percent of words

that are correct over multiple confidence intervals, as in Figure 4.1. To evaluate the impact on the confidence estimate of the 1-best word hypothesis, we use the standard normalized cross entropy (NCE) measure [112]:

$$NCE = (H_{max} - H_{conf})/H_{max}$$

where

$$\begin{aligned} H_{max} &= -p_c \log_2 p_c - (1 - p_c) \log_2 (1 - p_c) \\ H_{conf} &= -1/n \left[\sum_{w_i \text{ corr}} \log_2 p_i + \sum_{w_i \text{ err}} \log_2 (1 - p_i) \right] \end{aligned}$$

and where $p_c = n_c/n$ is the average probability that a hypothesized word is correct, p_i is the predicted confidence that w_i is correct, and the sum is over all n hypothesized words in the test set. (The NCE score has also been referred to as normalized mutual information.) An NCE equal to 1 means every word is correct with confidence 1, while an NCE of zero or less indicates that the word confidences are worse than just using the average accuracy as the confidence for every word. Finally, to evaluate the prediction of the probability of a missing word (i.e. the correct word is not in that slot in the confusion network), we use a DET curve.

4.1.4 Experiments

The primary aim of our experiments is to improve the confusion network confidences as a whole, so that confidences for multiple ASR hypotheses are improved. A series of experiments evaluating different feature sets and model configurations for that purpose are described below. In addition, we examine the impact of the resulting system on 1-best confidence estimation and missing word detection.

Confusion Network Posterior Correction

Our first series of experiments explored the different feature sets (baseline, extended and full) with a single regression SVM. Recall, the baseline feature set is that used in standard NN-based confidence prediction; the extended set adds five related features from a larger window, and the full set also adds lexical identity. As shown in Figure 4.2, the extended features improve results, although the full feature results do not.

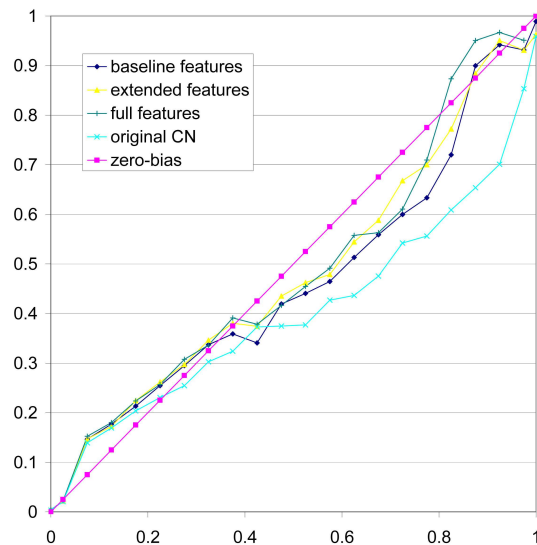


Figure 4.2: Various feature sets using one SVM

The majority of the CN slots had a top word with a posterior (from the recognizer) greater than .95, so to investigate further modeling improvements we experimented with using two SVMs to separate out the large portion of the data which had very high “original” posteriors: one SVM for slots with a top word having a posterior greater than .95, and the other SVM for those with a top word having confidence less than .95. The performance for mid-range confidences was improved, and the best results were obtained with all lexical identity features, but the results for the very highest confidence words were still severely biased. The poor performance was in part attributable to sparseness, because very few words remained with high confidence after applying the two SVMs. This problem is addressed by introducing a heuristic cap of .95 on the posteriors, chosen because the words with confidence of 1 output by the recognizer are only 95% accurate. This threshold would likely need to be tuned for other tasks.

Figure 4.3 shows a comparison of the bias of the original confidence output by the recognizer, the best single SVM (with extended features), and the full feature dual SVM with thresholding.¹ With the best case system, almost all of the bias for words with confidence above .4 has been eliminated. The bias for lower confidence words is more difficult to eliminate, in part because there are not

¹The threshold suppresses all confidence values of greater than .95, hence the lack of those points in the plots.

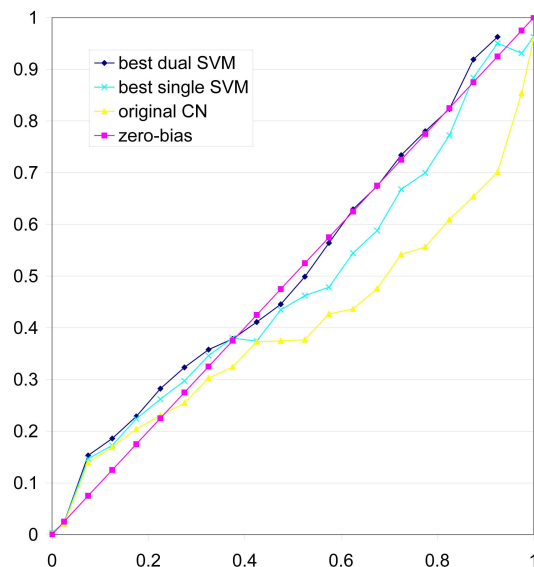


Figure 4.3: Comparison of bias plots for original posteriors, compensation using the best single SVM, and compensation using the best dual SVM with thresholding.

many and so the learning problem is more difficult.

1-Best Word Confidence Prediction

As described earlier, the highest ranking word in a slot is the best recognizer output, and its posterior probability can be used as a confidence estimate. While the goal of this work is to improve the posteriors in the network as whole, it is of interest to assess the impact on the 1-best hypothesis because of its particular importance. The original unwarped CN confidences give a NCE of .16 (with confidences clipped at .05 and .95 to avoid negative NCE values). Using the predicted missed word probability to normalize the network with the baseline feature set increases NCE to .19. Using the same features in a NN trained to predict 1-best word confidence explicitly results in an NCE of .22. The NCE of the 1-best word confidence using our best network compensation system (two SVMs, the full feature set, and thresholding) is .26. An SVM trained to predict confidence had an NCE of .20 (using our full feature set).

Detecting Missed Words

Another way to measure our confidence prediction is to evaluate how well we correctly predict our modeling target, whether or not a confusion network slot includes the correct word. The probability of missed word output by our system can be used to detect slots where the recognizer has not hypothesized the correct word, which would indicate regions where the lattice (or lexicon) should be expanded and the hypothesis rescored. The DET curve in Figure 4.4 shows curves for our system with the baseline features, and with the full feature set. The full feature set is almost always better than the baseline features, but still does not have great success. When false alarms are limited to 10%, 60% of the slots with missing words are not detected.

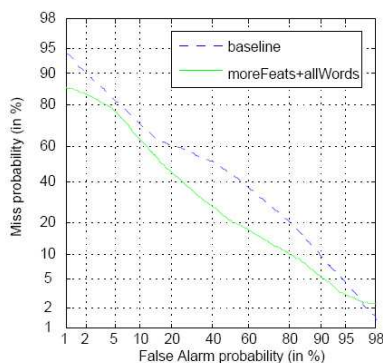


Figure 4.4: DET curve for detecting slots with missing words.

4.1.5 Summary

In summary, this section addresses the problem of accurate word posterior probability estimation at the network level, using prediction of the probability of missed words to adjust for biases introduced by using confusion networks. Using SVMs and simple features of the local context in the confusion network, very good performance is achieved, removing most of the bias in the network estimates. Using the resulting 1-best word posterior as a confidence estimate is an improvement over the original network, though not quite as good as predicting the corrected confidence for that word directly using the same features (using a NN). However, the SVM can make use of additional

features that lead to an overall improvement in 1-best word confidence. A by-product of the method is the availability of a probability of missed words, which might be used for OOV or more general error detection. Unfortunately, the performance of that detector on its own is still quite poor. One direction for future work is to improve this component.

Improved word confidence estimates can potentially contribute to improved ASR system combination, because component systems would have less biased confidence estimates for their hypothesis. Unfortunately, pilot experiments found that the improved word confidence estimates did not lead to better system combination. The next section turns towards leveraging this section’s work in confidence estimation to develop techniques for improving the approach to system combination, rather than the inputs to system combination as in the pilot study.

4.2 ASR System Combination

Rather than addressing uncertainty in ASR by directly evaluating multiple hypotheses in downstream processing, another potential avenue is to leverage confidence estimates from multiple systems to improve the accuracy of the 1-best hypothesis. Most state-of-the-art speech recognition systems combine multiple different component systems, and there is consensus that improvements from combination are usually best when systems are sufficiently different, but there is ambiguity about which system combination method performs the best. In addition, the success of commonly used combination techniques varies depending on the number of systems that are combined [52].

We present a new system combination technique, called *i*ROVER, for improved or intelligent ROVER. Our approach obtains significant improvements over previous approaches, and is consistently better across varying numbers of component systems (where relative improvements over ROVER are particularly large for combination when only using two systems). A classifier learns a selection strategy (i.e. a decision function) on features from the system lattices, and selects the final word hypothesis by learning cues to choose the system that is most likely to be correct at each word location. System combination is often most beneficial when the component systems are diverse, so we focus on the TC-STAR 2006 English speech recognition evaluation set where lattices from multiple sites are available.

Section 2.4.3 described previous work in system combination techniques. Section 4.2.1 de-

scribes our approach, and Section 4.2.2 provides experiments and results. Finally, we summarize the approach and findings in Section 4.2.3.

4.2.1 Approach

As described in the sections that follow, the *i*ROVER approach involves: i) emulating ROVER in lattice preprocessing and system alignment ; ii) extracting features from the different system hypotheses (including some newly proposed features); and iii) using these features in a classifier to select the best system at each slot in the alignment.

Lattice Preparation

The RWTH TC-STAR project partners kindly provided their development and evaluation lattice sets for the English task of the TC-STAR 2006 Evaluation Campaign. Our experiments use lattice sets from four different sites. Naturally, these lattice sets differ in their vocabulary, segmentation, and density. A compatible vocabulary is essential for good combination performance, as it is critical that words are represented by the same surface form across all component systems. The main problems are related to contractions, e.g. “you’ve” and “you have”, and the alternatives in writing foreign names, e.g. “Schröder” and “Schroder”. In ASR this problem is well-known and is addressed in scoring by using mappings that allow alternative forms of the same word. Such a mapping is provided within the TC-STAR Evaluation Campaign and is used it to normalize the lattices. In the case of multiple alternative forms only the most frequent one is used. Allowing multiple parallel alternatives would have distorted the posterior probabilities derived from the lattice. Furthermore, only one-to-one or one-to-many mappings are allowed. In the latter case, the time of the lattice arc is distributed according to the character lengths of the target words.

In order to create comparable posterior probabilities over the lattice sets, they are pruned to similar average density (where density is the number of arcs in the lattice divided by the number of reference words).. The least dense lattice set defined the target density: around 25 for the development and around 30 for the evaluation set.

Finally, the segmentation is unified by concatenating the lattices recording-wise. The concatenation was complicated by segmentations with overlapping regions, but the final concatenated lattices

scored equally when compared to the original lattice sets. The unified segmentation is required for lattice-based system combination methods like frame-based combination.

System Alignments

ROVER style alignment is used in this work as the basis for our system combination approach. At first glance the search space used by ROVER is very limited because only the first-best hypothesis from each component system is used. However, the oracle error rate is often very low, normally less than half of the best system’s error rate and pilot experiments indicated that lattice-based combination techniques seem not to benefit from broader search spaces, especially for three or more systems.

The result of the ROVER alignment can be interpreted as a confusion network with an equal number of arcs in each slot. The number of arcs per slot equals the number of component systems and thus makes the training and application of a classifier straightforward, because there is a constant number of choices for each confusion network slot.

A standard dynamic programming-based matching algorithm is used for the production of the alignments that minimizes the global cost between two hypothesis. The local cost function is based on the time overlap of two words and is identical to the one used by the ROVER tool [38]. Experiments with alternative local cost functions based on string matches could not outperform the simple, time overlap-based, distance function.

Hypothesis Features

We generate a cohort of features for each slot in the alignment, which are then used as input to train the classifier. The features incorporate knowledge about the scores from the original systems, as well as comparisons among each of the systems. The following paragraphs enumerate the six classes of feature types used in our experiments (with their names rendered in *italics*).

The primary, and most important feature class covers the *basic* set of features which indicate string matches among the top hypotheses from each system. In addition, we include the systems’ frame-based word confidence. These features are all the information available to the standard ROVER with confidences voting.

An additional class of features provides extended *confidence* information about each system’s hypothesis. This feature class includes the CN word confidence, CN slot entropy, and the number of alternatives in the CN slot. The raw language model and acoustic scores are also available. In addition, it includes a frame-based confidence that is computed from only the acoustic model and a frame-based confidence that is computed from only the language model score. Frame-based confidences are calculated from the lattices according to [172]; the CN-algorithm is an extension of [177].

The next class of features describes *durational* aspects of the top hypothesis for each system, including: character length, frame duration, frames per character, and whether the word is the empty or null word. A feature that normalizes the frames per character by the average over a window of ten words is also generated; this features can describe words that have unusually short or long durations compared to their neighbors. We use characters here as a proxy for phones, because phone information is not available from all component systems.

We also identify the system dependent *top error* words for the development set, as well as the words that occur to the left and right of the system errors. We encode this information by indicating if a system word is on the list of top ten errors or the top one hundred list, and likewise if the left or right system context word is found in their corresponding lists.

In order to provide *comparisons* across systems, we compute the character distance (the cost of aligning the words at the character level) between the system words and provide that as a feature. In addition, we include the confidence of a system word as computed by the frame-wise posteriors of each of the other systems. This allows each of the other systems to ‘score’ the hypothesis of a system in question. These cross-system confidences could also act as an indicator for when one system’s hypothesis is an OOV-word for another system, because the cross-system confidence would be zero for OOVs (another system cannot output a word that is an OOV). We also compute the standard, confidence-based ROVER hypothesis at each slot, and indicate whether or not a system agrees with ROVER’s decision.

The last set of features is computed relative to the combined *min-fWER* decoding. A confidence for each system word is calculated from the frame-wise posteriors generated by combining across all component systems. The final feature indicates whether each system word agrees with the combined systems’ min-fWER hypothesis.

Classifier

After producing a set of features to characterize the systems, we train a classifier with these features that will decide which system will propose the final hypothesis at each slot in the multiple alignment. The target classes include one for each system and a null class (which is selected when none of the system outputs are chosen, i.e. a system insertion).

The training data begins with the multiple alignment of the hypothesis systems, which is then aligned to the reference words. The learning target for each slot is the set of systems which match the reference word, or the null class if no systems match the reference word. Only slots where there is disagreement between the systems' 1-best hypotheses are included in training and testing.

The classifier for our work is Boostexter [135] using real Adaboost.MH with logistic loss (which outperformed exponential loss in preliminary experiments). Boostexter trains a series of weak classifiers (tree stumps), while also updating the weights of each training sample such that examples that are harder to classify receive more weight. The weak classifiers are then combined with the weights learned in training to predict the most likely class in testing. The main dimensions for model tuning are feature selection and number of iterations, which are selected on the development set as described in the next section.

4.2.2 Experiments

We first perform experiments using cross-validation on the development set to determine the impact of different feature classes, and to select the optimal number of iterations for Boostexter training. We then apply the models to the evaluation set.

Experimental setup

We present results on the EPPS 2006 English corpus. The corpus contains parliamentary speeches from the European Parliament and was collected within the TC-STAR project. All audio files are monaural with 16-bit resolution at a sampling rate of 16kHz. Corpus statistics for the English portion of the corpus are given in table 4.1.

The 2006 TC-STAR Evaluation campaign took place in February 2006. Besides RWTH Aachen [92], the following sites participated in the English task: LIMSI [73], IBM [125], UKA [159], and

Table 4.1: Corpus statistics for the EPPS English task of the 2006 TC-STAR Evaluation Campaign.

Corpus	Recording Period	Speech [h]	# Run. Words
Train06	May'05-Jan'06	87.5	1,600,000
Dev06	Jun'05	3.2	28,000
Eval06	Sept'05	3.2	30,000

IRST [21]. Afterward, all project partners kindly provided RWTH their best performing lattice set, systems and lattice sets are described in [52].

Table 4.2 summarizes the best results achieved on the single lattice sets. The latter columns show the results of CN and min-fWER based posterior decoding [95, 175].

Table 4.2: *WER[%] results for single systems.*

System	Viterbi		min-fWER		CN	
	dev	eval	dev	eval	dev	eval
1	10.5	9.0	10.3	8.6	10.4	8.6
2	11.4	9.0	11.4	9.5	11.6	9.1
3	12.8	10.4	12.5	10.4	12.6	10.2
4	13.9	11.9	13.9	11.8	13.9	11.8

Feature analysis on development data

We evaluate the various feature classes from Section 4.2.1 on the development set with a cross validation testing strategy. The results in Tables 4.3 and 4.4 are generated with ten-fold cross validation, which maintains a clean separation of training and testing data. The total number of training samples (alignment slots where there is system disagreement) is about 3,700 for the 2 system case, 5,500 for the 3 system case, and 6,800 for the 4 system case.

Table 4.3: WER results for development data with different feature classes.

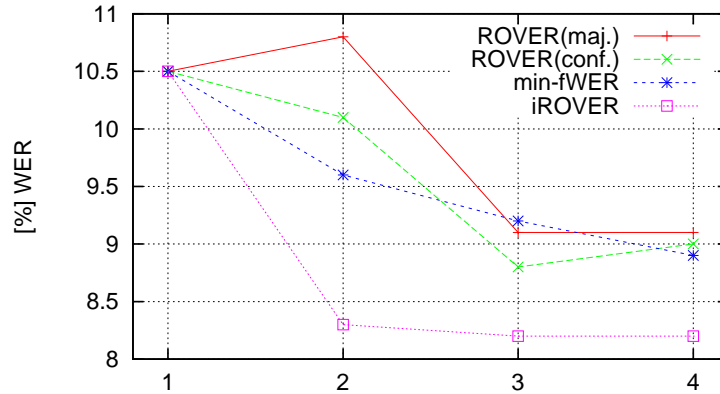
Features	2 System	3 System	4 System
ROVER	10.2%	8.8%	9.0%
<i>basic</i>	9.4%	8.6%	8.5%
+ <i>confidences</i>	9.3%	8.7%	8.4%
+ <i>durational</i>	9.2%	8.6%	8.4%
+ <i>top error</i>	9.0%	8.5%	8.4%
+ <i>comparisons</i>	8.9%	8.6%	8.4%
+ <i>min-fWER</i>	8.5%	8.5%	8.4%
+ <i>top+cmp+fWER</i>	8.3%	8.3%	8.2%
<i>all features</i>	8.3%	8.2%	8.2%

The WER results for different feature conditions on the development set are presented in Table 4.3. The typical ROVER with word confidences is provided in the first row for comparison, and the remainder of the rows contain the results for various configurations of features that are made available to the classifier.

The *basic* features are just those that encode the same information as ROVER, but the classifier is still able to learn better decisions than ROVER with only these features. Each of the following rows provides the results for adding a single feature class to the *basic* features, so that the impact of each type can be assessed.

The last two rows contain combinations of multiple feature classes. First, the best three classes are added, and then all features. Using just the best three classes achieves almost the best results, but a small improvement is gained when all features are added. The number of iterations in training is also optimized on the development set by selecting the number with the lowest average classification error across the ten splits of the training data.

Table 4.4: WER[%] results for development data with manual segmentation, and using cross-validation for *i*ROVER. All methods use the same alignment but different voting functions: majority-, confidence-, classifier-based-, and oracle-voting.

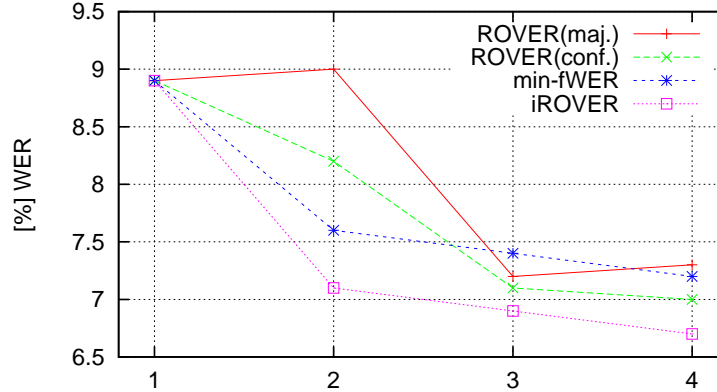


	2 System	3 System	4 System
ROVER (maj.)	10.8%	9.1%	9.1%
ROVER (conf.)	10.1%	8.8%	9.0%
min-fWER	9.6%	9.2 %	8.9 %
<i>i</i> ROVER	8.3%	8.2%	8.2%
oracle	6.5%	5.4%	4.7%
Rel. Improv.	18.6%	6.8%	8.8%

Results on evaluation data

After analyzing the features and selecting the optimal number of training iterations on the development data, we train a final model on the full development set and then apply it to the evaluation set. In all cases our classifier achieves a lower WER than ROVER (statistically significant by NIST matched pairs test). Table 4.4 and Table 4.5 present a comparison of ROVER with majority voting, confidence voting, frame-based combination, and our improved ROVER (*i*ROVER). The final two rows provide the oracle combination result and the relative improvement between *i*ROVER and standard ROVER.

Table 4.5: WER[%] results for evaluation data. All methods use the same alignment but different voting functions: majority-, confidence-, classifier-based-, and oracle-voting.



	2 System	3 System	4 System
ROVER(maj.)	9.0%	7.2%	7.3%
ROVER(conf.)	8.2%	7.1%	7.0%
min-fWER	7.6 %	7.4 %	7.2 %
iROVER	7.1%	6.9%	6.7%
oracle	5.2%	4.1%	3.6%
Rel. Improv.	14.5%	2.8%	4.3%

4.2.3 Summary

In summary, we develop *iROVER*, a method for system combination that outperforms other system combination techniques consistently across varying numbers of component systems. The relative improvement compared to ROVER is especially large for the case of combining two systems (14.5% on the evaluation set). The relative improvements are larger than any we know of to date, and the four system case achieves the best published result on the TC-STAR 2006 English evaluation set. The classifier requires relatively little training data and utilizes features easily available from system lattices. In addition, we have introduced a new method for incorporating lattice based features in selecting first-best hypotheses. Future work will investigate additional classifiers, classifier combination, and expanded training data. We are also interested in applying a language model to decode

an alignment network that has been scored with our classifier. Finally, it would also be interesting to train the classifier using a different objective than WER, which may be better focused towards target downstream tasks.

4.3 Conclusions

In this chapter we develop a new approach for reducing confidence estimation bias in confusion networks. While previous work focused mostly on the first-best words, our improved confidence estimates remove most of the bias present in raw confusion networks. Although bias in confidence estimation was reduced, pilot experiments found little improvements impact on system combination, so we turn to approaches that leverage ideas from confidence prediction methodology for improving system combination (and therefore reducing final WER).

*i*ROVER is a novel ASR system combination approach that reduces errors in speech recognition output from multiple systems. The approach incorporates features from system lattices and from cross-system comparisons that are motivated by our work in confidence prediction. The resulting recognition hypothesis achieves the lowest reported error rates for the TC-STAR evaluation set. In addition, the method can be easily adapted to optimize ASR for alternative objective functions that could optimize ASR for other downstream applications (rather than just WER).

Chapter 5

IMPROVING SENTENCE TRANSCRIPTION FOR MT

This chapter investigates the influence of sentence structure for machine translation (MT) of automatically recognized speech. First, we compare multiple approaches for automatic sentence boundary detection and then evaluate their impact on translation quality. We measure the impact using a state-of-the-art phrase-based statistical MT system on broadcast news translation tasks in Mandarin and Arabic. We find that carefully optimizing the segmentation parameters directly for translation quality improves the translation quality (measured by BLEU and TER) in comparison to independent optimization of segmentation quality for the predicted source language sentence boundaries.

We then turn to examining alternative ASR optimization criteria that preference improved sentence structure rather than just minimum word errors. We analyze translation errors, and find that errors in source sentence parse structure are more correlated with translation errors than are word or character error rates. Therefore, we proceed in optimizing ASR for minimum parse error rates, rather than word error rate, and compare the resulting translation quality. We find that optimizing ASR for parse accuracy can lead to recognition output that is better suited for machine translation.

The analysis of sentence segmentation in translation in this chapter is joint work with Evgeny Matusov and Hermann Ney at RWTH Aachen, Mathew Magimai-Doss and Dilek Hakkani-Tur at ICSI, and Mari Ostendorf at University of Washington, it was published in Interspeech 2007 [99]. The section on parsing-based ASR objective functions for translation is joint work with Mei-Yuh Hwang and Mari Ostendorf at University of Washington and Mary Harper at University of Maryland and was published at ICASSP 2008 [48].

5.1 Sentence Segmentation for Machine Translation

State-of-the-art ASR systems typically divide words into *utterances* based on speech/non-speech detection algorithms. These utterances may be very long, containing several sentences, or very

short sentence fragments (1-2 words). MT systems are often not able to translate (with an acceptable quality) utterances that are too long or too short. The MT search involves reordering that has exponential complexity with regard to the length of the input sequence, so very long sentences are computationally intractable. Also, very short utterances containing only 1-2 words cannot be translated well because of the truncated contextual information.

In this section we investigate the influence of automatic sentence segmentation on MT quality. We compare and combine two existing SU boundary detection algorithms [187, 102] and measure their impact on MT by evaluating the translations which utilize these boundaries. The translations are produced by a state-of-the-art phrase-based statistical MT system. A new feature for sentence segmentation that makes use of the phrase translation model from the MT system is also added and provides better MT results despite lower F-scores for sentence prediction.

This section is organized as follows: in Section 5.1.1 we review our approaches to sentence segmentation and introduce a new phrase feature; our baseline MT system is then presented in Section 5.1.2; Section 5.1.3 describes the boundary prediction experiments performed for the Chinese-to-English and Arabic-to-English large vocabulary translation tasks; and we provide a summary in Section 5.1.4.

5.1.1 Sentence Segmentation Approaches

ICSI+ Approach

In this work, we use the ICSI+ multilingual sentence segmentation tools [187] for sentence boundary detection as described in Section 2.1.3, which predicts sentence boundary locations based on lexical and prosodic cues with a hidden event language model and boosting classifier. For Arabic, in addition to the boosting classifier, a support vector machine (SVM) classifier is also used. Similar to [33], the posterior estimated from the combination of the posterior estimates of the two individual classifiers is then interpolated with the HELM posteriors.

RWTH Approach

The RWTH sentence segmentation algorithm [102] was developed especially for the needs of machine translation, predicting sentence boundaries with a log-linear combination of language model

and prosodic features. The details are described in 2.1.3. The primary difference from the ICSI+ approach is that the minimum and maximum length of a segment can be restricted (e. g. 4 and 60 words, respectively). In addition, other features can be included in the log-linear model, here we can include the ICSI+ posteriors as an additional feature to improve the RWTH approach.

Phrase Coverage Feature

Another feature that can be included in the RWTH approach is motivated by the phrase-based machine translation algorithm that will be applied to the segmented speech in the next processing step. The idea is to make sure that word sequences for which good phrasal translations exist will not be broken into subsequences by a sentence boundary. To this end, all bilingual phrases are extracted from the training data of the MT system which match any word sequence in the automatically recognized words from the evaluation data. A bi-gram language model on the source language parts of these bilingual phrases is trained (using unmodified Kneser-Ney smoothing), where the phrases are treated as sentences (so words within the phrase, but not across phrases, are used to estimate the bi-gram). The amount of training data (bilingual phrases) was not sufficient for training higher order n-grams.

The phrase coverage feature for each word f_j in the input is then the bigram language model probability $p(f_{j+1}|f_j)$. If this probability is high, then the word sequence $f_j f_{j+1}$ probably has a good phrase translation (and a sentence boundary directly after f_j would be undesirable). If this probability is low, the MT system will likely translate each of the two words by backing off to single-word translations and in this case the phrasal context will be lost anyway. Therefore an (incorrect) boundary between f_j and f_{j+1} should not have a significant negative influence on translation quality. Note that the phrase coverage may help to improve the MT quality, but not necessarily improve the segmentation results. In fact, the precision and/or recall of sentence boundaries may actually drop (see Section 5.1.3), so it is important to optimize the scaling factor of this feature for MT quality.

5.1.2 Baseline MT System

In our experiments, we use a state-of-the-art phrase-based translation system [103]. In this system, a target language translation $e_1^I = e_1 \dots e_i \dots e_I$ for the source language sentence $f_1^J =$

$f_1 \dots f_j \dots f_J$ is found by maximizing the posterior probability $Pr(e_1^I | f_1^J)$ as described in Section 2.3.2. The translation model is a log-linear model that includes language model scores, bi-directional word and phrase translation scores, and phrase orientation modeling.

5.1.3 Experimental Results

Corpus Statistics

The experiments were performed on the GALE Chinese-to-English and Arabic-to-English large vocabulary tasks. We evaluated the segmentation and translation quality on the automatically recognized broadcast news portion of the GALE MT 2006 evaluation data. The ASR output was generated by the SRI 2006 Mandarin and Arabic evaluation systems [153]. The reference transcriptions of the Chinese evaluation data contain about 19K characters and 633 sentence units. The Arabic reference transcriptions contain about 12K words and 661 sentence units. The Mandarin ASR system has a character error rate (CER) of 17.8% for the MT 2006 Evaluation set. The Arabic system has a WER of 33.7% on the MT 2006 Evaluation set.

The MT systems were trained using the bilingual training corpora from LDC. The statistics of the training corpora are shown in Table 5.1. For tuning the boundary prediction parameters of the sentence segmentation system, we used a held out part of TDT4 as a development set for Chinese and the BBN 2006 tune set for Arabic (broadcast news speech data). The baseline RWTH MT systems were initially optimized on the NIST 2004 text evaluation data and further adjusted to the speech input using the GALE 2006 tune sets for Arabic and Chinese.

Evaluation Criteria

The quality of SU prediction was measured in terms of precision (P), recall (R), and F-measure in comparison with manual reference boundaries defined on correct transcriptions. For the evaluation, the predicted boundaries were inserted into the reference text based on the edit distance alignment.

The MT quality was determined using the well-established objective error measures BLEU [122] and TER [147] (as described in Section 2.3.2). We used the tool of [101] to determine the alignment with the multiple reference translations based on the word error rate and, using this alignment, to re-segment the translation output to match the number of reference segments. For the evaluation

Table 5.1: Corpus statistics for the bilingual training data of the Chinese-to-English and Arabic-to-English MT systems (GALE large data track).

		Source	Target
Chinese to English	Sentence Pairs	7M	
	Running Words	199M	213M
	Vocabulary Size	223K	351K
Arabic to English	Sentence Pairs	4M	
	Running Words	126M	125M
	Vocabulary Size	421K	337K

data, the error measures were calculated using three manually created reference translations.¹ For the speech development data, only single reference translations were available.

The MT evaluation was case-insensitive, with punctuation marks. The punctuation marks were predicted by the MT system which had been trained by removing punctuation marks from the source phrases, but left them in the corresponding target phrases as described in Section 2.3.2.

Sentence Segmentation LM

All experiments use n -gram language models with modified Kneser-Ney smoothing as implemented in the SRILM toolkit [151]. The HELMs for sentence boundary prediction were trained with the same data sources as for training the ASR language models, including all available newswire text and broadcast news speech transcripts.

Translation LM

The MT system used a 4-gram target language model for the Arabic-to-English translation and a 6-gram language model for Chinese-to-English translation in the search. They were trained on the English part of the bilingual training corpus and additional monolingual English data from the

¹These are the original manual reference translations produced for the GALE evaluation by NVTC, on the basis of which the “gold standard” translation was created.

Gigaword corpus. The total amount of language model training data was about 600M running words.

MT Results for SU Boundary Prediction

Table 5.2 summarizes the segmentation and translation results for the ICSI+ and RWTH algorithms. In the ICSI+ approach, the boundaries are inserted if the sentence end posterior probability exceeds a certain threshold. Here, we tried the thresholds 0.2, 0.5, and 0.8, which led to average segment lengths of 16, 24, and 45 words, respectively. The best threshold determined on a development set is 0.2. This means that shorter segments are better for translation, i. e. recall is more important than precision. For this algorithm, the setting with the highest F-score also results in the best translation quality.

Table 5.2: Segmentation and translation results [%] for different sentence segmentation settings on the Chinese-to-English task. The best results for each score are highlighted in boldface.

algorithm	P	R	F-score	BLEU	TER
ICSI+ 0.8	93.1	38.6	54.6	19.2	68.5
ICSI+ 0.5	81.8	64.8	72.3	20.2	67.5
ICSI+ 0.2	69.6	83.2	75.8	20.7	67.3
RWTH	72.2	74.3	73.2	20.7	67.4
+ phrase LM	57.2	82.2	67.5	21.2	67.0
RWTH+ICSI	75.0	77.5	76.2	20.8	67.1
boundary after every 30 words				18.1	69.7
reference sentence units				20.7	66.9

The RWTH approach with the scaling factors optimized on a development set has a lower F-score than ICSI+, but performs similarly in terms of BLEU and TER. One advantage of this algorithm is that extreme sentence lengths cannot occur in its output. Here, the minimum and maximum SU length was set to 4 and 60 words, respectively. In contrast, even when using a low posterior probability threshold of 0.2 which favors short SUs, the ICSI+ system produced 5 sentences that

were 100 or more words long. Forty sentences contained only one word. Most probably, the translations of these segments were not adequate. Segmentation quality improves when ICSI posteriors are used as a feature in the RWTH approach (RWTH+ICSI), but translation quality is equivalent to the individual approaches.

The best translation quality (BLEU score of 21.2) is achieved by adding the phrase coverage feature. It is notable that the F-score for this setup is lower, but the recall is high. The phrase coverage feature results in additional SU boundaries that may not correspond to manually defined boundaries, but have less impact on the translation because phrasal context at these extra boundaries was not captured during MT training. The ICSI+ system with a 0.2 threshold has even slightly higher recall, but the increased boundaries are less suitable for translation when compared with the phraseLM boundaries, as shown by the poorer translation performance.

For comparison, we also report the translation results for two baseline setups. In the first setup, a boundary is inserted after every 30 words in a document. This is clearly not a good idea, since the BLEU score is low. In the second setup, the manual reference boundaries are inserted into the ASR output based on the alignment with the correct transcriptions. We see that the automatic SU boundary prediction results in translations of the same or even somewhat better quality than when reference sentence boundaries are used.

Another possible baseline result would have been to translate the whole document without segmentation. However, this is not possible due to the time complexity caused by reordering and the large memory needed to store the language model.² These constraints emphasize the importance of sentence segmentation for MT.

In Table 5.3, we report the results for the same experiments on the Arabic-to-English task. Here, the F-measures for the SU boundaries are lower than for Chinese. The main difference relative to Chinese-to-English translations is that it is advantageous to produce longer segments (an ICSI threshold of .8 led to an average length of 33 words). We attribute this to the fact that reordering is mostly local when translating from Arabic to English. If two sentences are translated as one, their words are usually not swapped. In general, the Arabic-to-English MT is less sensitive to SU boundaries than the Chinese-to-English MT. All automatic segmentation approaches are as good in

²We use sentence-specific LMs, loading only the n -grams which appear in a single sentence.

Table 5.3: Segmentation and translation results [%] for different sentence segmentation settings on the Arabic-to-English task. The best results for each score are highlighted in boldface.

algorithm	P	R	F-score	BLEU	TER
ICSI+ 0.8	76.9	43.3	55.4	21.8	62.2
ICSI+ 0.2	40.1	84.9	54.4	21.6	62.8
RWTH	52.6	54.4	53.5	22.0	62.3
+ phrase LM	49.7	60.3	54.5	22.1	61.9
RWTH+ICSI	61.3	68.8	64.8	21.9	62.4
boundary after every 30 words				20.6	63.7
reference sentence units				21.5	62.4

terms of MT quality as when the reference SU boundaries are inserted into the ASR output.

5.1.4 Summary

In this section we test the importance of segment boundaries in automatically recognized speech for MT quality. We combine two approaches for SU boundary prediction in order to produce sentences that are best suited for a state-of-the art phase-based statistical MT system. We used a novel feature, phrase coverage, in order to couple the segmentation with the predictive power of the phrase translation model. Our experiments find that the best translation results are achieved when boundary detection algorithms are directly optimized for translation quality, providing equal or better translation quality when compared with reference sentence boundaries. For conditions with the best results, recall is higher than precision, but other factors such as interaction with MT phrase tables seem to play a role.

5.2 Parsing-based ASR Objectives for Machine Translation

Taking sentence segmentation as fixed, this section proposes an alternative to word error rate (WER) for optimizing speech recognition with the goal of improving MT performance. While speech translation performance usually rises as recognition errors decrease, there is no reason to assume that

WER is the best ASR optimization target for input to translation. Ideally, ASR parameters would be optimized to directly maximize translation quality, but that approach is currently computationally infeasible given the complexity of both ASR and MT systems. Instead, we investigate alternative optimization criteria for speech recognition, which may produce recognition output that is better suited for input to machine translation. We hypothesize that a parsing-based objective, SParseval, may be a better for applications such as translation. SParseval preferences ASR sentences with parse structures most similar to that of the reference sentence, rather than just minimizing word or character errors. We analyze the correlation of parsing error rates on source language recognition output with final translation quality, and then present an approach for optimizing ASR to minimize parse errors.

We describe our motivation in Section 5.2.1 and describe SParseval in Section 5.2.2. As will be shown in Section 5.2.3, even when the MT system does not explicitly use syntax, the SParseval score is more correlated with MT quality than character error rate (CER), which is a more commonly used measure than WER for Mandarin. This motivated us to exploit parsing-based ASR objectives in discriminative score combination for speech recognition in Section 5.2.4 and investigate the impact on MT in Section 5.2.5. We find some improvements in translation when using true sentence segmentation, and summarize our findings in Section 5.2.6.

5.2.1 Motivation

Recent work on machine translation (MT) of speech has provided mixed results on the impact of speech recognition errors. One study shows that recognition errors, source, and domain-mismatch are important variables in predicting translation errors [69]. But other researchers report that improvements in ASR error do not consistently lead to gains in translation performance (especially with low WERs less than 10%). Certainly, large ASR improvements are likely to benefit translation, but could it be the case that smaller gains (e.g., 10-20% reduction in error) simply won't matter until the performance of MT improves? Or could it be that word error rate (WER) is simply not the right objective?

The main problem with WER as an objective for speech recognition when the end goal is language processing (whether machine translation, information extraction, or other types of process-

ing), is that WER considers all errors as equal. Intuitively, it seems clear that all words are not equally important – with filled pauses as an extreme example. Scoring methods sometimes account for such problems with “allowable errors,” but automatic performance optimization is typically based on simpler word error metrics that do not make these distinctions. In addition, it is probably not the case that all error types are equal. For example, a deletion is probably much worse than an insertion or substitution, since all information is lost when a word is deleted but at least some phonetic cues are present with other error types. Ideally, one would use a weighted word error rate, where the weighting function was appropriate for the target application.

A problem with learning a weighting function is that weights dependent only on error types are probably not sufficiently rich, and vocabulary-dependent weights would yield too many free parameters. The alternative would be to use word-based weights that are motivated by the task but not automatically learned. For example, one might use an information-based measure, such as the inverse-document frequency weights often used in information retrieval and topic clustering. The approach explored here is a parsing-based scoring function, specifically *S*Parseval.

5.2.2 *S*Parseval

The standard text-based parse scoring method [17] and the typical scoring tool [137] are insufficient for scoring speech parses because their behavior is undefined when the ASR hypothesis does not exactly match the words of the reference parse tree. *S*Parseval [130] was developed for scoring parses on speech recognition output, so that parses on ASR can be scored against standard reference parse trees. The tool produces recall, precision, and F-score for bracket and head dependency scoring. In our experiments we use dependency scoring, which maps each word in a sentence (for both the hypothesized words+parse and the reference) to a triple that includes the word, the head word that it is dependent on, and the type of dependency. Recall and precision are then measured by the number of matching triples between the reference and hypothesized parse. A headword with multiple dependent children will be present in more triples, and an error on that headword would then contribute multiple errors (giving greater importance to headwords with multiple dependencies). Dependency scoring uses a head percolation table, and dependency scores can be computed either with or without the word-level alignment constraints needed to calculate bracket scores.

For this paper we compute head dependency scores based on a head table that developed by University of Maryland for the LDC Chinese TreeBank 6.0. When there is a mismatch between the sentence segmentations of the reference and ASR transcripts, as will be the case in the correlation studies reported in Section 5.2.3, we use alignment-based head-dependency scores calculated over the spoken document. When reference and ASR hypotheses have matching segmentation, as in the experiments reported in Section 5.2.5, we utilize head-dependency scores without alignment constraints. Although alignment adds an extra match constraint that can slightly reduce dependency scores, the reduction is negligible when scoring the parses on short segments.

Our experiments use automatically generated reference parses, because our translation test set has not been annotated by humans for parses, but the automatic parse on the reference text is relatively high quality (and certainly closer to truth than the parses that are generated over the recognized words with automatic sentence segmentation).

5.2.3 *Correlation Study*

We first assessed the usefulness of SParseval as an ASR objective function in speech translation applications by computing the correlation of ASR scoring criteria on Mandarin speech (CER and SParseval) with error measures on the English translations. Our main interest was in improving HTER, a human-based error measure that computes the translation error rate (TER) between the MT hypothesis and a human-edited version that reflects the same meaning as a reference translation with a minimal number of edits [147]. However, since HTER is costly to obtain, we also compare to TER, which can be automatically computed (on the single available reference).

The study focused on the broadcast news (BN) portion of the Mandarin GALE [1] 2007 evaluation test set, which consists of 66 documents for which HTER results were available. The documents are typically a news story or portion of a story that include about 15 sentences (for a total of 1700 sentences). We also looked at a subset of 30 documents for which the average CER is less than 3% in order to evaluate differences for very low ASR error rate segments.

To obtain the SParseval results, both the Chinese reference transcript and the ASR hypothesis were parsed automatically, as described in Section 5.2.5, and head dependency scores were computed using the SParseval tool. Since the parser used human-annotated sentence segmentations for

Table 5.4: Correlation between two ASR scores (CER and SParseval) and two MT scores (HTER and TER) for Broadcast News. (TER score also provided for comparison with ASR scores.)

		MT Score	
Test Set	Score	HTER	TER
Eval07-BN	CER	0.32	0.46
	SParseval	0.44	0.61
	TER	0.52	–
CER < 3% subset	CER	0.19	0.26
	SParseval	0.38	0.47
	TER	0.32	–

the reference and automatic segmentation for the ASR hypotheses, scoring was based on document-level alignment to handle sentence segmentation mismatch.

The results are reported in Table 5.4, which shows that SParseval is substantially more correlated with both HTER and TER than CER. Even for the low error rate subset, where one would expect less benefit from ASR improvements and hence a lower correlation, there is a big difference in the correlations. Of course, in no cases are the correlations very large, due to the fact many errors are a result of MT modeling (separate from errors that can be attributed to ASR). For comparison, we also provide the correlation between the automatic MT measure TER and the human MT measure HTER. TER computes the distance of the automatic English translation to the English reference, so it should also be able to capture errors from the MT system (whereas CER and SParseval can only measure errors from the ASR system). Across all BN documents TER is more highly correlated with HTER, but for the low CER documents SParseval seems to predict HTER as well as TER.

We investigated whether automatic sentence segmentation hurts the usefulness of the SParseval objective, since it is known that parse scores degrade considerably with segmentation errors. Comparing results using automatic vs. oracle segmentation, we found that the correlation was in fact higher for the automatic case. We hypothesize that SParseval is implicitly incorporating segmentation error into the score, which is useful for predicting MT performance since it also is sensitive to

Table 5.5: Correlation between two ASR scores (CER and SParseval) and two MT scores (HTER and TER) for Broadcast Conversations. (TER score also provided for comparison with ASR scores.)

		MT Score	
Test Set	Score	HTER	TER
Eval07-BC	CER	0.37	0.63
	SParseval	0.26	0.62
	TER	0.30	–

segmentation error.

Finally, we also experimented with broadcast conversations (BC), but correlation with HTER was lower for SParseval compared to CER. The results are presented in Table 5.5. Interestingly, the correlation for TER is in the same range as the ASR scores (and even lower than for CER). ASR (and MT) error are much higher for broadcast conversations because the domain is more difficult, so we do not evaluate low error rate documents separately (because no document has low ASR errors).

5.2.4 ASR Objectives for N-Best Rescoring

The ASR objective function s_{ij} impacts the system at the N-best rescoring stage, where scores from different knowledge sources (e.i. log acoustic model probability, log language model probability, word count) are linearly combined to form a final score for reranking:

$$H^* = \operatorname{argmax}_i \sum_j s_{ij} w_j \quad (5.1)$$

where i indicates the hypothesis index, and j the component score. The weights w_j are trained to optimize some objective function on the top ranking hypothesis based on the weighted combination. We use the weight optimization function in the SRILM toolkit [151], specifically `nbest-optimize`, which uses a simplex-based “Amoeba” search on the objective function [124]. The optimal parameters returned by the search are then used in ASR decoding to select a final hypothesis H . In a 2-system combination framework, weights are optimized separately for N-best lists of the two systems, and the two N-best lists are then combined at the character-level

via confusion network combination [95] with the posterior probability computed by applying the optimized weighting parameters, as described in Section 2.4.3.

The typical objective function used to optimize weights for ASR is minimum CER (or WER) with respect to the correct transcription. We introduce an alternative parse-based optimization criterion, building on the framework of [66], that specifically maximizes SParseval’s dependency F-score by minimizing error $\hat{e} = L \times (1 - F)$. We include L , the number of words in the reference segment, to avoid over-weighting short segments. The dependency F-score for a hypothesis is based on the reference transcription with an automatically generated parse.

In addition, we optionally include the parse confidence as another knowledge source. In combining different knowledge sources, a problem arises when there are cases where there is no score. For example, an utterance with only laughter or noise and no words will have no parse score. To handle such cases, we add another “knowledge source” that is a simple indicator of these conditions so that we can learn a compensating weight, similar to the word insertion penalty.

5.2.5 MT Experiments

Test Data Description

We used the GALE Mandarin 2007 development audio set as our testbed. Our goal is to translate from Mandarin speech to English text. The broadcast news part of this set is denoted as dev07-bn here. Dev07-bn comes from 40 different Chinese shows aired in November 2006, consisting of 54 different documents, for a total of 108 minutes and 19,000 Chinese characters. There are 524 sentences in the English gold translations. Each sentence has only a single gold translation. We evaluate, as in Section 5.1.3, with BLEU and TER.

ASR System

The ASR system adopted in this paper is the one used in our GALE 2007 evaluation [58], except that there is no cross adaptation with the ASR system RWTH Aachen. In brief, two acoustic models were trained on 870 hours of speech data, one based on PLP+pitch features, the other MFCC+pitch+MLP (multi-layer perceptron based phoneme posterior features). Maximum-likelihood based word segmentation on the Chinese training text was used based on 60,000 Chinese lexical words, and n-gram

(up to 4-gram) language models were trained on over 1 billion words of text. Each testing show was automatically segmented into “utterances,” based on long pauses and automatic speaker boundaries prior to recognition. The two AM systems cross adapted each other and produced 1000 best hypotheses each, for each testing utterance, and the two N-best lists were combined via a confusion network at the character-level. The best character sequence was then re-segmented into words with the same word segmenter used during training. The system is an updated version of the system used previously in Section 5.1.

Mandarin Parser

The Mandarin parser is based on a modification of the Berkeley unlexicalized parser [123], which uses a new approach for learning that begins with a PCFG grammar derived from a raw TreeBank and then iteratively refines the grammar. This approach can learn to distinguish alternative uses of words and phrases, thereby producing higher quality parses. The original Berkeley parser achieved a bracketing F-score of 82.4% on the Chinese TreeBank 5.2, and further improvements increase the F-score to 86.5%, while typical F-scores from previous work are between 79% and 81%. Improvements to the baseline parser include addressing unknown words by using all characters of an unknown word to estimate word probability, improving the F-score to 82.8%. Removing rarely invoked unary rules from the trees prior to training gives further improvements to 84.6%. Since the Berkeley parser uses little explicit context of a symbol, parent annotations are added to the TreeBank prior to training, resulting in F-score of 84.93%. Finally, training data from the recent release of CTB6.0 is added, including Broadcast News trees.

For experiments here, the parser is trained on a text-normalized version of CTB6.0 (i.e., all Arabic digits were replaced by verbal tokens in the tree) with punctuation removed to better match the conditions to which the parser would be applied. Recently released Broadcast Conversation TreeBank data was also added to training for parsing of Broadcast Conversations. We parsed both the Chinese reference transcription and ASR hypotheses using the same parser with MAX-RULE-PRODUCT decoding. The confidence scores are the log probabilities for the best parse returned by the parser.

ASR Annotation

The ASR system outputs a sequence of words corresponding to the specified testing segments in the input shows, which is then segmented into sentences with the RWTH+ICSI system described in Section 5.1 (without the phraseLM feature). We also apply inverse text normalization so that spoken numbers are converted to digits. The input to translation for our experiments is then a sequence of Chinese sentences without punctuation and with numbers in digit form.

MT System

For automatic translation, we used the state-of-the-art phrase-based statistical machine translation system built by RWTH Aachen University, which is an updated version with minor changes from the one described in the previous section, Section 5.1.2.

Results

We conduct two series of experiments: the first focuses on broadcast news and evaluates various optimization approaches, and the second looks at both broadcast news and broadcast conversation (using an updated ASR system with reference sentence segmentation).

Our initial experiments use only the broadcast news portion of the corpus, because no broadcast conversation TreeBank was available (so automatic parse quality was much lower for the conversational domain). To better understand the impact of the two objective functions (CER and SParseval), we compare them in both automatic and oracle conditions, where “automatic” involves N-best rescoring with the discriminatively trained weights. We also try to improve results for the SParseval objective by including an additional parse confidence knowledge source (log parse probabilities) with the acoustic and language model scores. This leads to five different experimental conditions:

1. CER: This is the baseline system.
2. SParseval dependency F-score (S-F-score): We use the “error” \hat{e} , described in Section 5.2.4 with the standard knowledge sources.

3. SParseval F-score+Confidence (S-F-score+Conf): We use \hat{e} as above, but adding the parse confidence knowledge source.
4. SParseval Oracle (S-oracle): As an oracle comparison, we select the hypothesis from the N-best lists with the best SParseval F-score.
5. CER-oracle: The comparable oracle for CER is the hypothesis that has the minimum CER among all N-best hypotheses.

In Table 5.6, we report ASR CER, MT TER, and MT BLEU [122] scores for each experiment, together with the number of automatic sentences (SUs). While computing MT errors, we ignore differences in case and punctuation for simplicity. MT errors are computed at a per-sentence basis, based on the sentence definition in the gold translation. To do that, we automatically segment the machine-translation output (now in English per SU sentence) into the same number of sentences as the gold translation, by minimizing the word-alignment errors [101]. Unfortunately, we did not get improvements in translation by optimizing directly for SParseval in recognition, but we also found little margin for improvement in BLEU and HTER between the baseline condition and the best case oracle condition.

Table 5.6: ASR scores and MT scores on dev07-bn.

ASR Objective	#SU	CER	TER	BLEU
CER	905	3.4%	70.4%	18.9
S-F-score	904	3.5%	70.4%	18.8
S-F-score+Conf	905	3.4%	70.3%	18.9
S-Oracle	904	1.2%	69.7%	19.1
CER-Oracle	903	0.9%	69.5%	19.3

We also conducted experiments with updated ASR and MT systems, which also use true sentence segmentation. The results of Section 5.1 show that automatic segmentation is sufficient for our

translation approach when using WER as an objective for ASR, but parse quality can be significantly degraded when using automatic segmentation [65], so segmentation may impact the usefulness of this objective. Hence, we assess whether the usefulness of the SParseval measure is improved when reference segmentation is applied to the source recognition. Baseline performance somewhat improves for ASR as well when compared with automatic segmentation. The results in Table 5.7 compare CER, TER, and BLEU over the same dev07-bn subset as above, but we restrict our experiments to only compare optimizing for CER, and optimizing for SParseval F-Score. Here, we find small improvements for MT when using SParseval-optimized ASR as compared to CER optimization.

Table 5.7: ASR scores and MT scores on dev07-bn (true sentences) for two ASR objectives: character error rate (CER) and SParseval F-score.

ASR Objective	CER	TER	BLEU
CER	3.0%	70.8%	19.3
S-F-score	3.0%	70.6%	19.4

Finally, we also investigated SParseval optimization for the broadcast conversation translation. A recently released Mandarin broadcast conversation TreeBank allowed us to train an improved broadcast conversation parser. The parser was trained by concatenating the broadcast conversation TreeBank to the existing Mandarin TreeBanks. The results are presented in Table 5.8, and we find larger improvements in both TER and BLEU than those in the broadcast news domain, even though there is a small degradation in CER. Further analysis is required to evaluate how SParseval-optimized ASR differs for broadcast news and broadcast conversation domains.

5.2.6 Summary

In summary, this section has attempted to address the question of whether improvements to ASR could have a greater impact on MT if optimized in terms of a different objective function than WER

Table 5.8: ASR scores and MT scores on dev07-bc (true sentences) for two ASR objectives: character error rate (CER) and SParseval F-score.

ASR Objective	CER	TER	BLEU
CER	15.3%	76.0%	12.4
S-F-score	15.4%	75.5%	12.7

(or CER for Mandarin). While we found that SParseval scoring of ASR outputs is more correlated with human evaluations of translations than CER, we did not get large improvements in broadcast news translation by optimizing directly for SParseval in recognition. However, we also found that there is not much margin for improvement in BLEU and TER between the baseline and best case oracle conditions.

Finally, for broadcast conversations, we find that SParseval-optimized ASR can lead to improvements in both TER and BLEU (with our updated system and true sentence segmentation). Although the CER is slightly worse compared with ASR optimized for CER, automatic translation results benefit from the alternative optimization metric. Further experiments that investigate SParseval optimization for broadcast conversation with automatic sentence segmentation are required to fully evaluate the influence of automatic sentence segmentation.

5.3 Conclusions

This chapter evaluates the impact of sentence segmentation approaches for machine translation. Comparisons with previous segmentation approaches found that including features that account for the phrase-based nature of the translation system produced translation quality that is equal or better than translations performed with reference sentence boundaries.

In addition, an alternative optimization ASR criterion is investigated for selecting recognition hypotheses that are better suited for automatic translation. A parsing-based criteria, SParseval, is found to be better correlated with translation quality than the standard ASR measures of word or character error rate. SParseval-optimized ASR can lead to improvements in translation quality, in particular for broadcast conversations. Further research is required to evaluate the impact of automatic segmentation on parsing and parsing-based ASR optimizations.

Chapter 6

AUTOMATIC DETECTION OF SUB-SENTENCE STRUCTURE

This chapter describes approaches for sub-sentence structure detection on speech recognition output. Natural language processing for text often utilizes sub-sentence cues such as commas to improve modeling, but speech recognition does not typically produce punctuation or any other indication of sentence structure. In this chapter, we investigate two alternatives for annotating structure: an orthographic or surface form representation (commas), and an acoustically-motivated representation (prosodic phrase boundaries and emphasis).

Our approach for automatic detection of commas will be described in Section 6.1, and Section 6.2 will propose an alternative approach that learns unsupervised prosodic structure. Reliable detection of commas or other sub-sentence structure would likely be useful in spoken language processing tasks. Because currently the language processing modules that we are working with primarily leverage text data for training, the commas are best suited to their needs. The impact of providing commas will be discussed in Chapters 7, 8, and 9.

6.1 Automatic Comma Detection

We apply the boundary detection approaches described in Section 2.1.2, treating commas as sub-sentence boundaries. Boundary detection is treated as a classification problem, where each word boundary is treated as an event. The classifier uses a combination of a hidden-event language model (5-gram) to exploit lexical information and sequence dependencies and a Boostexter classifier [135] to exploit lexical cues (word tri-grams) in combination with prosodic information and predict the sequence of commas, C .

$$\hat{C} = \underset{C}{\operatorname{argmax}} p(C|W, F) = p(C, W)p(F|W, C) \quad (6.1)$$

Prosodic features are the same features used in sentence segmentation, because the cues are similar for both types of boundary events. The features include various measures and normalizations of

pause duration, phone duration, fundamental frequency, and energy. Posteriors from the two component models are interpolated with weights optimized on a held-out set. Section 6.1.1 describes our Mandarin comma detection approach, and Section 6.1.2 applies the approach to English.

6.1.1 Mandarin Comma Detection

The SRI-LM toolkit [151] (with Kneser-Ney smoothing) is used to train a hidden-event model for boundary prediction on the Chinese Gigaword corpus, where the training text has been stripped of all punctuation but comma and caesura. (Chinese has two forms of commas: the typical commas, and a caesura that occurs only in list contexts.) Preliminary experiments showed that interpolation with additional data sources (such as TDT4) did not provide a gain, and some recent Broadcast News sources use quick transcription that do not distinguish between commas and caesura. The Boostexter model is trained on a subset of the TDT4 Chinese news data (40 shows) using flexible alignment [166] to obtain word transcripts from the closed-captions.

Test Data and Evaluation Criteria

In these experiments, we use ASR words and reference sentence boundaries on a held-out set of 10 shows from our TDT4 set. The held-out set has about 80,000 words, 5,000 commas, and 3,000 sentences. We also evaluate comma performance on an additional speech test set that includes transcripts from the GALE Mandarin ASR/MT 2006 development test set, where we use the four development shows from the GALE Year 1 BN audio release. The data set includes about 15,000 words, 1,500 commas, and 600 sentences.

For performance analysis, reference punctuation is determined by first aligning the ASR words to reference words and then choosing the best punctuation alignment if multiple word alignments are equivalent. While comma and sentence boundary prediction could be treated jointly as a multi-class problem, in this work we take predicted or reference sentence boundaries as given and then predict commas within each sentence, in order to factor out the interaction between comma prediction and sentence prediction from our primary analysis (which focuses on only comma performance). In addition, predicting commas separately from sentence boundaries facilitates the use of different thresholds on sentence and comma posteriors in the various applications that we explore later.

Comma Detection Results

Figure 6.1 shows the precision/recall curve for comma detection with the HELM alone, Boostexter alone, and the combination. Performance for the text only HELM and Boostexter with prosodic features are both very similar, even though the HELM is trained on a significantly larger data set (Gigaword versus only 40 hours of TDT4). The relatively equivalent performance indicates the strength of the prosodic features, which are able to compensate for the significantly smaller size of available speech training data. The best results are obtained with the combined model and show the text and prosodic cues are complementary features. We have merged the two commas types into one class for this figure to evaluate only comma position performance, ignoring comma type. We found only a slight improvement in the merged comma prediction by modeling the two types of commas separately and mapping them to a single comma afterwards (rather than training on merged commas).

Table 6.1 shows confusion between comma and caesura types on the TDT4 held-out set using a posterior threshold of .5. There are many fewer caesuras, but even with the highly skewed distribution there is very little confusion between the two comma types. It is likely that the model is able to easily distinguish the two types, because the caesura is used only to separate items in a list (so occurs in very different acoustic and lexical contexts). In our experiments, less than 0.2% of commas are labeled as caesuras, while less than 10% of caesuras are labeled as commas. When the two classes are merged, detection performance has an F-score of .65 (precision=.75, recall=.57).

Table 6.1: Confusion table counts for comma and caesura prediction on the Mandarin TDT4 held out set, using a .5 comma threshold and reference SUs.

True	Predicted			total
	comma	caesura	null	
comma	2924	8	2049	4981
caesura	32	104	245	381
null	992	12	74207	75211
total	3948	124	76501	80573

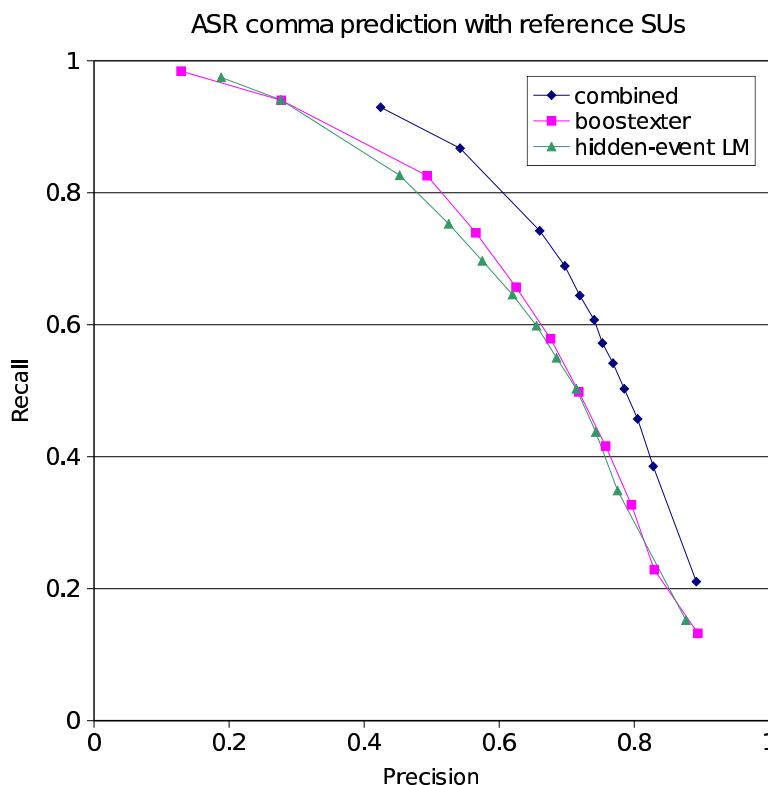


Figure 6.1: Comma prediction for Mandarin TDT4 ASR words with reference sentence boundaries and three different modeling approaches.

Next we turn to the GALE Y1 BN data, where we measure comma performance with automatic sentence boundary prediction. Results for comma prediction on this data are given in Table 6.2 for automatically detected sentence boundaries. Using a threshold of 0.5 for the sentence posterior, sentence boundary detection performance is $P=0.53$, $R=0.79$, and $F=0.63$, where sentence boundary recall is much higher than precision. While precision for the commas with automatic sentence boundaries is similar to using reference sentences, recall is significantly lower. The primary reason is that many “false” sentence boundaries are hypothesized at comma locations, limiting the possible recall. When selecting sentence boundaries with the .5 sentence boundary threshold, 27% of the reference commas locations are unavailable due to being marked as sentence boundaries, while with a .8 sentence boundary threshold, only 12% are unavailable.

While the comma precision is high, recall is relatively low because of false predictions of sentence boundaries at comma locations. Table 6.2 also illustrates trade-offs in comma and sentence boundary detection. Higher SU thresholds reduce SU boundary recall, but allow for better comma prediction (because fewer SUs are predicted at reference comma locations). Overall SU F-Score is not reduced by higher SU thresholds though, because precision increases. Chapter 5 found that Chinese-English translation benefited from low sentence segmentation thresholds, which suggests that comma boundaries (such as those marked as sentences when using a low sentence threshold) may be useful boundaries. But, higher sentence boundary thresholds may be better if commas are also predicted.

Joint prediction of commas and sentence boundaries would likely improve performance, though it is not clear this small improvement would have a big impact on downstream applications. Further, separate modeling is more convenient when the goal is to optimize punctuation for the end application and not for punctuation accuracy directly.

6.1.2 English Comma Detection

Our English comma modeling approach is essentially the same as for Mandarin, except that we do not have caesuras. The Boostexter model is trained on a subset of TDT4 (which excludes ACE data we will test on in Chapter 8) with about 60k words. Reference commas are obtained by aligning commas from reference transcriptions to the words of the flexible alignment. Portions of the corpus with high alignment error (due to poor transcription) are removed from the training data. The number of iterations in training, as well as the optimal threshold, is tuned on a portion of held out development data. The hidden-event language model is a 5-gram with Kneser-Ney smoothing trained on a large collection of English text, including Gigaword, TDT2, TDT4, Hub4, Bizweek, and BBC text.

Comma Detection Results

We report results for a development and test set that are part of TDT4, the same development and test set we will use to evaluate information extraction in Chapter 8. We use reference sentence boundaries in order to focus on comma detection. Results are presented in Table 6.3, where performance

Table 6.2: Results for comma detection on Mandarin ASR transcripts with different thresholds for comma posteriors on GALE Y1 BN data. The far left column contains the threshold used in automatic SU detection, while the far right column contains the threshold used for comma detection under each SU setting.

SU	SU Boundary			Comma Boundary			C
	F	Prec	Rec	F	Prec	Rec	
.2	0.60	0.46	0.86	0.30	0.65	0.20	.2
				0.24	0.67	0.15	.5
				0.19	0.72	0.11	.8
.5	0.63	0.53	0.79	0.38	0.67	0.26	.2
				0.31	0.69	0.20	.5
				0.25	0.73	0.15	.8
.8	0.63	0.66	0.60	0.46	0.65	0.35	.2
				0.39	0.68	0.27	.5
				0.31	0.70	0.20	.8

is consistently lower than for the Mandarin task. The lower F-scores may be explained in part by a smaller amount of available acoustic training examples (caused by reserving a large portion of the TDT4 data for later evaluation with the information extraction task).

6.1.3 Summary

The first published results that we know of for Chinese comma and caesura detection are presented, with performance similar to sentence boundary detection when commas are predicted within reference sentence boundaries. There is significant interaction between comma and sentence boundaries though, so comma performance (particularly recall) degrades when automatic sentence boundaries are used. Joint modeling of commas and sentences may improve this issue by allowing one model to be informed of cues for both types of boundaries, but separate modeling is more convenient for tuning separate thresholds. It is unclear what the optimal trade-off between sentence and sub-sentence

Table 6.3: Results for comma detection on English ASR transcripts with different thresholds for comma posteriors (reference sentence boundaries). Optimal F-score threshold of 0.68 selected on dev set.

	Dev			Test		
Thresh	Prec	Rec	F	Prec	Rec	F
0.2	0.26	0.66	0.37	0.27	0.67	0.39
0.5	0.31	0.55	0.40	0.32	0.55	0.41
0.68	0.35	0.50	0.41	0.35	0.47	0.40
0.8	0.36	0.42	0.39	0.38	0.42	0.40

boundaries is; the best operating point likely depends on the downstream task.

6.2 Unsupervised Learning of Prosodic Structure

This section investigates methods for automatically predicting prosodic events in speech recognition output. While punctuation and sentence boundaries are natural cues in text processing, spoken language processing may benefit more from directly incorporating prosodic structure instead of orthographic conventions because prosody can be richer in that the boundaries occur more frequently and provide cues to more types of syntactic structure than commas.

The amount of hand labeled training data is very limited for most types of prosodic phenomena, especially for languages other than English, so we focus on developing unsupervised techniques for learning to model prosody in speech. The two primary events of interest are prosodic phrase boundaries, which indicate perceptual groupings of words that are roughly related to syntax, and prosodic emphasis, which signals important words in spoken communication. Successful detection of prosodic breaks could be useful for providing reordering constraints in speech translation (such as the approaches of Chapter 9), and emphasized word detection could be used as a guide to focus attention on important words in information extraction and translation tasks.

This section is organized as follows. Section 6.2.1 reviews the corpus and evaluation setup, Section 6.2.2 describes our approach to clustering, Section 6.2.3 describes an approach for unsuper-

vised emphasis detection, and Section 6.2.4 investigates unsupervised break detection. We conclude with a summary in Section 6.2.5.

6.2.1 Corpus and Evaluation

As a first step, we investigate unsupervised clustering of prosodic events on the Boston University Radio Corpus [118], where prosodic break level and emphasis have been labeled by humans using the ToBI prosodic annotation system described in Section 2.2. We use 124 utterances from 32 stories (9000 words) in the “radio” portion of speaker f2b for our clustering experiments, and 24 utterances from 4 stories (2100 words) in the “labnews” portion for testing our classifier. Prosodic labels are also available for a portion of the Switchboard corpus, but we choose the Radio Corpus because we are interested in the news domain (and more previous work has been reported for this corpus).

We use three evaluation methods: accuracy, F-score, and normalized cross entropy (NCE). Accuracy is typically used in related prosody work and F-score facilitates comparisons with comma results. Finally, in order to evaluate the information provided by the classifier posterior (since the goal is to use posteriors, not hard decisions, in subsequent language processing), we compute the NCE. NCE compares the entropy of the confidences (posteriors) generated by the classifier to a baseline entropy determined by the prior distribution of word emphasis.

$$NCE = (H_{max} - H_{conf})/H_{max}$$

where

$$\begin{aligned} H_{max} &= -p_e \log_2 p_e - (1 - p_e) \log_2 (1 - p_e) \\ H_{conf} &= -1/n \left[\sum_{w_i \text{ corr}} \log_2 p_i + \sum_{w_i \text{ err}} \log_2 (1 - p_i) \right] \end{aligned}$$

and where $p_e = n_e/n$ is the average probability that a word is emphasized, p_i is the predicted confidence that w_i is emphasized, and the sum is over all n words in the test set. Typically p_i is also thresholded to avoid severe over or under confidence. An NCE equal to 1 means the classifier is always correct with confidence 1, while an NCE of zero or less indicates that the classifier performs worse than chance.

6.2.2 *Clustering*

Our clustering experiments utilize the CLUTO¹ toolkit, which offers partitional, agglomerative, and graph-partitioning based methods with multiple similarity/distance functions including Euclidean distance, cosine, correlation coefficient, and extended Jaccard. Pilot experiments found that agglomerative (partitional biased agglomerative in CLUTO) and partitional (repeated bisections in CLUTO) clustering performed best, so we utilize those approaches in our experiments. We also restrict our experiments to use cosine distance to reduce experimental dimensions.

In addition, a number of different optimization criteria are available. I1 and I2 attempt to minimize different variants of intra-cluster similarity, E1 attempts to maximize inter-cluster differences, and H1 and H2 attempt to optimize a combination of both, the details can be found in [185]. Previous work has found the H2 criterion to have the best performance in terms of overall accuracy, as well as robustness to different numbers of clusters and clustering approach [185], so we always use the H2 criterion in our experiments.

We use the same prosodic features as in comma detection, except that for emphasis clustering we remove those features that describe pitch and energy difference between the current and next word because those features are less informative to whether the current word is emphasized. Another difference is that pitch for unvoiced regions is estimated with spline interpolation as in [78], so that there is always a pitch value in order to avoid missing features. Standard clustering algorithms do not deal well with missing features, whereas Boostexter for comma prediction could account for missing features.

6.2.3 *Unsupervised Prosodic Emphasis Detection*

Our emphasis clustering experiments compare labels from two automatic clusters to collapsed human labels of accent and no-accent (with label assigned by picking the emphasis cluster as the auto-cluster with maximum overlap with emphasized words in the hand-labeled data). Results for partitional and agglomerative clustering approaches are presented in Table 6.4, where partitional clustering slightly outperforms agglomerative clustering. The accuracy is similar to results reported in [81], but cannot be directly compared because we use different test sets.

¹<http://glaros.dtc.umn.edu/gkhome/views/cluto>

Table 6.4: Comparison of Unsupervised Emphasis Clustering Approaches

clustering method	Precision	Recall	F-Score	Accuracy
partitional	.74	.63	.68	.67
agglomerative	.69	.64	.66	.64

After obtaining automatic labels from clustering, we train a Boostexter classifier to predict accent versus no-accent and compare the results to a classifier trained on true hand-labels for accent using the same features as clustering. This approach uses a different classifier after clustering in order to compare to supervised training with our standard configuration, which allows extensions such as including part-of-speech (POS) tags as features. As Table 6.5 shows, the performance with Boostexter is similar to that achieved with automatic clustering, though there is a slight loss in precision relative to direct clustering. In addition, the performance with unsupervised training is significantly lower than supervised training, though still much better than chance (58%). Table 6.5 also compares classifiers with and without hand part-of-speech tags. For training emphasis classification from hand-labels, POS tags improve all measures. When training from auto-clusters labels though, POS tags do not significantly impact the classification. Analysis of the weak learners selected by Boostexter showed that POS was rarely used in classification.

Table 6.5: Emphasis Classification with Boostexter

training labels	Precision	Recall	F-Score	Accuracy	NCE
hand-labels	.810	.906	.855	.848	.504
hand-labels (with POS)	.826	.922	.871	.865	.542
auto-clusters	.697	.636	.665	.682	.101
auto-clusters (with POS)	.695	.638	.665	.681	.101

NCEs for the classifier trained on hand-labels and the classifier trained on auto-clusters are also presented in Table 6.5. The auto-cluster NCE has been thresholded to only have confidences between .3 and .7 in order to avoid negative NCEs (while the hand-labels case varies from .01 to 0.99). While the auto-cluster trained classifier is degraded compared with the version trained on hand-labels, it still is performing better than chance.

6.2.4 *Unsupervised Prosodic Break Detection*

For supervised training and evaluation, we reduce the ToBI prosodic break indices to two classes: non-break (0 to 3) and break (4). (In this corpus, which is mostly read news and rarely disfluent, no breaks are labeled as disfluent with the p diacritic.) While multiple mappings are possible, we chose to represent two classes, for a more direct comparison to comma prediction. Future work should also evaluate other potential mappings. We perform partitional clustering to obtain two clusters and, as in the emphasis clustering, we compare our clusters to true human labels (assigning the break cluster as the auto-cluster with maximum overlap to the break labels in the hand-labeled data). In addition, after the initial clustering we reassign cluster labels based on the intra-cluster similarity measure so that the cluster sizes match the class priors (so that the remaining instances in the break class are those with the highest intra-cluster similarity measure). Table 6.6 shows the confusion between classes, indicating that with a 2 class model, break level 3 (intermediate intonational phrase) is better mapped to the “no break” class than the “break” class (full intonational phrase). In Table 6.7, precision, recall, and F-Score are presented for breaks, as well as overall word accuracy.

Table 6.6: Break Clustering Confusion Table

	no break	break
break level 1	4592	458
break level 2	1026	155
break level 3	736	61
break level 4	918	1145

Table 6.7: Unsupervised 2 Class Prosodic Break Clustering

clustering method	Precision	Recall	F-Score	Accuracy
partitional	.63	.56	.59	.83

Table 6.8 shows that performance with Boostexter is similar to that achieved with automatic clustering, though there is again a slight loss in precision relative to just clustering. Again, we choose a decision threshold on the confidence such that the proportion of break instances matches the prior (.2). The auto-cluster NCE has been thresholded to only have confidences between .3 and .7 in order to avoid negative NCEs (while the hand-labels case varies from .01 to 0.99).

As in Section 6.2.3, the performance with unsupervised training is significantly lower than supervised training, though still better than chance (20% of words are breaks). Table 6.5 also compares classifiers with and without hand part-of-speech tags. For training break classification from hand-labels, POS tags have little effect, although they slightly improve NCE. When training from auto-clusters labels, as was the case for emphasis POS tags do not significantly impact the classification (again, analysis of the weak learners selected by Boostexter showed that POS was also rarely used in break classification when training with auto-clusters).

Table 6.8: Break Classification with Boostexter

training labels	Precision	Recall	F-Score	Accuracy	NCE
hand-labels	.893	.717	.796	.927	.557
hand-labels (with POS)	.891	.720	.796	.927	.581
auto-clusters	.538	.549	.544	.816	.129
auto-clusters (with POS)	.537	.549	.543	.816	.129

Finally, an alternative evaluation approach ignored prosodic phrase boundaries that coincide with sentence boundaries in order to better compare with comma prediction. F-score degrades about

10% relative (to .71) for the hand-labels training case, and about 20% relative for the auto-clusters training case (to .45). The performance for the auto-cluster trained classifier is similar (even slightly better) to that of the English comma prediction in Section 6.1.2, although the comma prediction is speaker independent (while our prosodic phrase prediction is speaker dependent). This shows promise for future work utilizing prosodic breaks.

6.2.5 *Summary*

We find that unsupervised clustering of prosodic events can discover emphasis and prosodic phrases in unlabeled data, but with an F-score that is 20% higher than what is achieved with supervised training. Future work should investigate unsupervised clustering for prosodic events in languages and domains that do not have human labels (also for multiple speakers) and then measure the impact of providing the prosodic event predictions to downstream applications. While the classifier approach does provide confidence for prosodic events, performance is significantly lower compared to training on hand labeled data. Including lexical information, such as part-of-speech tags, indicated some gains for emphasis prediction, but more investigation is required in order to improve modeling when training from auto-cluster labels.

6.3 *Conclusion*

Section 6.1 presents results for English comma and Mandarin comma/caesura detection and evaluates interactions with sentence boundary detection. Comma prediction performance varies depending on the sentence segmentation operating point, since many false sentence boundaries occur at comma locations. Section 6.2 describes approaches for learning unsupervised prosodic structure. Analysis of these preliminary results indicates that although unsupervised learning techniques show promise for discovering prosodic structure, supervised learning is currently much more reliable (in particular for prosodic phrases), and therefore likely to be more informative to downstream applications. Labeled data with commas is currently much more prevalent than prosodically annotated data (and much larger amounts of training data can lead to higher accuracy in prediction), so we focus on predicting commas. The following chapters (7, 8, and 9) incorporate automatically predicted commas into information extraction and machine translation tasks to evaluate the impact on overall

system performance.

Chapter 7

COMMAS FOR TAGGING

This chapter evaluates the impact of automatically predicted commas on part-of-speech (POS) and name tagging for speech recognition transcripts of Mandarin broadcast news. One of the key differences between tagging speech and text sources (other than the potential for transcription errors) is that typical ASR systems do not output punctuation, which are used in most text processing systems. In [94], researchers from BBN showed that missing commas can have a dramatic impact on information extraction performance, with performance losses typically bigger than those for moving from reference to automatic sentence segmentation (for a range of word error rates on English news). In this chapter, we confirm these results for Mandarin and further look at how much performance can be recovered using automatically predicted commas. We examine name tagging, as in the BBN study, but also look at part-of-speech tagging which benefits name tagging (and other NLP tasks) as a pre-processing step. We find a significant gain in both POS and name tagging accuracy due to using automatically predicted commas compared to sentence boundary prediction alone.

In the next sections, we describe the speech tagging framework, the corpora and evaluation paradigm used in the studies, experimental results on punctuation prediction and its impact on the tagging tasks, and finally conclude with a summary of the key findings. This is joint work with Zhongqiang Huang and Mary Harper at Maryland, Heng Ji and Ralph Grishman at NYU, Dilek Hakkani-Tur at ICSI, Wen Wang at SRI, and Mari Ostendorf at University of Washington. The results were published at the IEEE/ACL Workshop on Spoken Language Technology in 2006 [47].

7.1 Corpora and Evaluation

Different corpora were used for training the various component systems, as described below in Section 7.2. In all cases, text normalization was needed to remove phrases with bad or corrupted codes, and convert numbers, dates and currencies into their verbalized forms in Chinese (to be consistent with the form produced by the ASR system). Among these, number normalizations were

performed using a set of context-independent and context-dependent heuristic rules. Then automatic word segmentation was run, using all punctuation marks as delimiters. For training most systems, we kept sentence boundary punctuation marks, comma and caesura marks, and removed all other punctuation marks.

The speech test set in this work includes transcripts from the GALE Mandarin ASR/MT development test set, where we use the four dev shows from the GALE Year 1 BN audio release (the same set as we used for evaluating commas in Section 6.1.1). The data set includes about 15K words (about 26K characters). To avoid over-tuning on this set, all text data from months covered by these shows are excluded from training.

The target data for this work is automatically transcribed speech, specifically Mandarin broadcast news, but there is no such speech data with hand-annotated part-of-speech tags and name labels. For that reason, most of the development work involved text corpora, where annotated data is available and precision/recall can easily be measured. For experiments with speech, we have adopted a change comparison method to assess the impact of comma prediction on both POS and name tagging accuracy for speech recognition output. Specifically, human annotators examine only those tokens for which the automatic POS (or name) predictions differ on the speech recognition output and assess whether the change corrects or introduces an error, with access to the reference transcription.

7.2 *Speech Tagging Framework*

The overall system architecture used here involves running automatic speech recognition, punctuation prediction, and then part-of-speech tagging or name tagging.

7.2.1 Speech Recognition System

The speech recognition system used in this work is a state-of-the-art system, based on the SRI Decipher recognizer [153] and trained/tuned specifically for the Mandarin broadcast news task. Training texts for the system from a variety of sources were automatically word-segmented according to the maximum n-gram probability as in [57], using all punctuation marks as delimiters during segmentation. The top 60K words were used as the decoding vocabulary, which includes several thousand frequent Chinese person names. The recognizer combines cepstra, pitch, and neural network phone

posteriors as features, and uses MPE training, cross-system adaptation, and a 5-gram mixture language model with components from 9 separate text sources. On the broadcast news development set used in these experiments, character error rate is 5.6%.

7.2.2 Sentence and Comma Prediction

We use the approach of Section 6.1 and the sentence segmentation tools of [187] for comma and sentence boundary detection, respectively. Language models are trained using the same data as for ASR, and prosodic models train mostly on TDT4. Sentence boundaries are selected with a confidence threshold for optimal sentence detection F-Score, and then commas are predicted within the selected sentences. On the test set, sentence boundary detection performance at threshold 0.5 is $F=0.63$ ($P=0.53$ and $R=0.79$) and comma detection performance at threshold 0.5 is $F=0.31$ ($P=0.69$ and $R=0.20$). Downstream tasks are evaluated with commas at multiple thresholds, in order to optimize commas for best performance on the final objective.

7.2.3 Part-of-speech Tagger

University of Maryland provided a Viterbi part-of-speech tagger that builds on the tagger developed in [163] which uses trigram transition probability $P(T_i|T_{i-1}, T_{i-2})$ and trigram emission probability $P(W_i|T_i, T_{i-1})$, where T_i and W_i represent the i -th tag and word. When a word was not observed during training (unknown word), it estimates the emission probability as a weighted sum of $P(S_i^k|T_i, T_{i-1})$, where S_i^k is the k -th suffix in word W_i . When applied to LDC Chinese Treebank 5.2, the tagger obtained a tagging accuracy of 93.6% (69.2% on unknown words); however, the accuracy of the tagger was improved to 94.5% (76.8% on unknown words) by enriching the context model in two ways: the emission probability is replaced by $P(W_i|T_i, T_{i-1})^{\frac{1}{2}} \times P(W_{i-2}|T_{i-2}, T_{i-1})^{\frac{1}{2}}$ for both known and unknown words, and $P(W_i|T_i, T_{i-1})$ is replaced by the geometric mean of $P(C_i^k|T_i, T_{i-1})$ for all the characters C_i^k in any unknown word W_i (and similarly for $P(W_{i-2}|T_{i-2}, T_{i-1})$). The POS tagger is trained on the Mandarin TreeBank.

7.2.4 *Name Tagger*

The name tagger provided by NYU is based on an HMM that generally follows the Nymble model [16]. It identifies names of three classes: people, organizations, and locations. Nymble used an HMM with a single state for each name class, plus one state for non-name tokens. To take advantage of the structure of Chinese names, the NYU approach uses a model with a larger number of states, 14 in total. The expanded HMM can handle name prefixes and suffixes, and has separate states for transliterated foreign names. The HMM is supplemented with a set of post-processing rules to correct some omissions and systematic errors. Some of these rules are dependent on the part-of-speech tags assigned to the tokens. The name tagger is trained on the ACE 2005 training corpora.

7.3 *Part-of-Speech Tagging*

In an attempt to optimize the tagger for the condition of tagging speech transcripts with automatically generated commas and caesuras, we performed a series of experiments on textual data to determine the impact of punctuation on tagging accuracy, and to assess the best conditions to train our tagger when using automatically generated commas. For these experiments, before scoring the tag sequence, we remove all punctuation along with their tags to more fairly compare the tag accuracy on words resulting from the absence or presence of punctuation of various qualities. These studies, selectively reported in Table 7.1, used the LDC Chinese Treebank 5.2 with 10-fold cross-validation. We report results with no punctuation, reference punctuation, predicted punctuation, and combined punctuation (where the tagger is trained on the concatenation of all the three previous conditions). We also report on four different punctuation types: none, all (includes other reference punctuation, such as quotes), merged comma, and caesura/comma.

Our preliminary results found that training and testing on matched conditions is always better than mismatched. For example, if we train the tagger using all of the Treebank punctuation and then apply it to tag word sequences with automatically generated commas, there is a serious increase in tagging error (e.g., 7.10% using comma and caesura predictions). We also found that using a comma prediction threshold of 0.5 (out of 0.2, 0.5, and 0.8) gave the best accuracy. So all results in Table 7.1 train and test with matched conditions, using a comma threshold of .5.

There is a negligible improvement from keeping the distinction between comma and caesura,

rather than merging the two to a single comma type. The best overall tagging accuracy for comma predicted data was obtained when training on the concatenation of all three punctuation sources (combined), with a error reduction of about 3% relative.

We also evaluated the impact of automatic comma prediction on POS tagging accuracy for the ASR output on the speech test set. We compared tagging results using our Viterbi tagger under two conditions. For the first case without punctuation, the ASR output was tagged by a tagger trained on the Treebank with all punctuation removed. For the second case, in which ASR output was augmented with predicted commas and caesuras with a 0.5 threshold, the best training scenario for automatic commas from Table 7.1 was used. Three annotators were asked to compare the POS tag changes in the two tagging outputs, without knowing in advance which system they came from. To support the comparison, we adapted an emacs tagging tool used by LDC to highlight the differences and mark up whether the change was from incorrect to correct, correct to incorrect, or incorrect to some other incorrect tag. All tag changes related to word segmentation and/or ASR errors were discarded. If the POS of a word could not be agreed upon among the annotators, then the majority vote was used for scoring (or the tag change of the word was discarded when all annotators disagreed). Of the 247 differences between the no punctuation and the automatic comma prediction tagging outputs, 29 were discarded, 120 were positive changes, 69 were negative changes, and 29 were wrong in both cases. Hence, the predicted commas significantly improved POS tagging accuracy ($p \leq 0.00027$ using the sign test).

7.4 Name Tagging

The name tagger was trained on 585 documents from the training data for the 2005 ACE (Automatic Content Extraction) evaluation, containing 198k words. Three separate name tagger models were trained: one with all sentence-internal punctuation in the training texts removed, one with the commas and caesuras added automatically, and one with the reference commas and caesuras retained.

The effect of comma prediction on named entity tagging was then evaluated using two test corpora, a text corpus and an ASR transcript. The text corpus consisted of 50 documents from the ACE 2005 training set (42 manually-prepared broadcast news transcripts and 8 newswire articles),

Table 7.1: POS tagging performance on various training/test conditions using the Viterbi tagger.

Punctuation Source	Train Punctuation Type	Test Punctuation Type	Error (%)
None	None	None	7.01
Treebank	All	All	6.51
	Merged comma	Merged comma	6.60
	Caesura/comma	Caesura/comma	6.56
Predicted	Merged comma	Merged comma	6.87
	Caesura/comma	Caesura/comma	6.86
Combined	Merged comma	Merged comma	6.84
	Caesura/comma	Caesura/comma	6.83

containing a total of 2671 names. The results are shown in Table 7.2. The first row represents a system trained and tested without commas; the last row a system trained and tested with the commas and caesuras from the original corpora. True commas produced a 1% gain in NE F-measure. The intervening row shows the results using comma prediction (with the system distinguishing commas and caesuras); this yields half the gain (0.5% in F-measure) of the true-comma case. The comma predictions changed the tagging of 31 tokens in the text test corpus; 20 incorrect tags were corrected, 6 correct tags were changed to incorrect ones, and 5 incorrect tags were changed to other incorrect tags. (The performance when not distinguishing commas and caesuras was slightly but not significantly worse – 0.1% lower in F-measure.)

The ASR test corpus, as for the POS tests, was the speech test set drawn from GALE Y1 Mandarin ASR+MT common dev set, and included 881 sentences with approximately 1700 names. The comma predictions changed the tagging of 59 tokens in the test corpus; 44 incorrect tags were corrected, 9 correct tags were changed to incorrect ones, and 6 incorrect tags were changed to other incorrect tags. The evaluation of changes was done ‘blind’ by two native speakers and then adjudicated; the independent assessments agreed 94% of the time.

In examining the changes, we observed a number of cases where the comma predictor was able

Table 7.2: Named entity tagging performance on news text under different punctuation conditions.

	Recall	Precision	F-measure
No commas	85.1	84.7	84.9
Comma prediction	85.6	85.1	85.4
True commas	85.7	86.1	85.9

to predict a comma before a name, and this enabled the name tagger to identify a name that it had previously missed, or to correct a name boundary error. For example, in the sentence (translating the actual Chinese example):

More than 200 pictures including the masterpieces by [Zhang Daqian]_{PER1}, [Zhao Zhi Qian]_{PER2}, [Xu Beihong]_{PER3} and [Qi Baishi]_{PER4} etc. were on sale in [Shanghai]_{LOC}.

The second name, “Zhao Zhi Qian” is missed when commas are not present because the “Zhi” in “Zhi qian” can also be interpreted as the common word meaning “’s” or “of”. The comma predictor (correctly) predicts commas before and after this name, and the name tagger then recognizes it.

Preliminary experiments on ACE entities and relations with the text data show a larger impact than for name tagging. The ACE entity score (as described in Section 2.3.1) improves from 52.9 in the no-comma case to 55.3 with predicted commas, compared to 56.4 for reference commas. Chunking results on text without commas are poor, so many nominal mentions are missing (in many cases two NPs spanning a comma are mistakenly connected). As a result of missed name detections co-reference results further degrade due to error compounding. Additional tuning of commas for information extraction may provide further gains, and will be investigated in Chapter 8.

7.5 Conclusions

In summary, in experiments with Mandarin broadcast news, this chapter finds that automatically predicted commas, despite a relatively low F-measure, can lead to a significant improvement in both POS and name tagging, relative to the case of using only automatically predicted sentence boundaries in ASR transcripts. The results on name tagging confirm prior work by BBN on English

news, that lack of commas hurts performance, and extends the work by demonstrating that about half of the loss can be recovered through automatic comma prediction. Automatic prediction with distinction between comma and caesura is possible, since there are few confusions between the two types, but it did not lead to significant gains in POS or name tagging. Our experiments found that tagging performance was influenced by how commas were including in training (using reference commas, or a threshold on automatic commas), and future work could better address this issue by a tighter coupling (or joint modeling) of punctuation and POS or names.

Chapter 8

COMMAS FOR INFORMATION EXTRACTION

This chapter extends the studies of Chapter 7 to evaluate the effect of automatic sentence boundary detection and comma prediction on entity and relation extraction in speech, which goes beyond finding names to also identifying links between them. A previous study found that this task degraded by 13.5% for entity scores and 25% for relation scores when reference sentence internal punctuation was removed [94]. In this chapter, we investigate how to optimally predict automatic punctuation for information extraction. Beyond just adding automatic punctuation, we show that punctuating ASR transcripts according to maximum F -measure of period and comma annotation results in sub-optimal information extraction. Similar to the study of machine translation in Chapter 5, we find that different punctuation decision thresholds can be chosen in order to improve the entity value score and the relation value score.

After presenting the experimental setup in Section 8.1, we present results in Section 8.2. Section 8.3 discusses the behavior of the system with a deeper analysis of the results. Section 8.4 summarizes the contributions and discusses future work.

This work is a collaboration with Benoit Favre and Dilek Hakkani-Tür at ICSI, Ralph Grishman and Heng Ji at NYU, and Mari Ostendorf and University of Washington. It was published at ICASSP 2008 [37].

8.1 Experimental Setup

We evaluate the impact of automatic punctuation detection on information extraction by varying the confidence threshold for comma and sentence boundary prediction.

8.1.1 Test Data

All the presented experiments are conducted on the portion of TDT4 English broadcast news that overlaps with the ACE'04 information extraction reference data. The speech was transcribed by

SRI's Broadcast News speech recognizer [165] with an estimated word error rate of 18%. The reference is formed by subtitles that do not exactly match the actual spoken words. In total, we use 131 stories from 101 shows that represent approximately 38k words and 4 hours of speech. Mean sentence length in the reference data is close to 15 words.

This set of stories is split into a test set for evaluation and a development set (dev) for parameter tuning. Each of the two sets represent half of the data. Punctuation prediction and information extraction systems are trained on separate data from similar corpora as detailed in the next section.

ACE reference data is only available in the form of character spans from the reference text and had to be mapped to the ASR output using word-to-word alignment. This results in imprecise offsets where the ASR output contains insertions and deletions. Imprecisions are accounted for by relaxing IE evaluation: after IE is performed, offsets in the ASR output are mapped (based on a token alignment) into offsets in the reference text and the result is then scored.

8.1.2 *Punctuation Prediction*

We focus on two types of punctuation: periods and commas. (By periods, we mean sentence boundaries, because so few instances of other sentence-final punctuation occur in this corpus.) Experiments showed that joint prediction of commas and periods was similar to independent prediction, so we chose to model them separately in order to more easily investigate different thresholds over the two types of events. Performance of both period and comma detection are presented in terms of F -measure.

Sentence boundary detection

ICSI provided sentence boundary detection as described in Section 2.1.2. The system is trained on 500k words from a portion of the TDT4 corpus. The decision threshold is determined on the development set and tested against the held-out test set. The training set is disjoint from the ACE annotated files used for IE evaluation.

Comma detection

Our comma modeling approach combines a hidden event language model with word level posteriors from Boostexter, as described in Section 6.1.2. The Boostexter model is trained on a subset of TDT4 that is separate from the ACE data, with about 60k words. The optimal F -measure threshold is determined with reference sentence boundaries on the development set, and evaluated on the test set.

8.1.3 Information Extraction

The IE components used for these experiments were developed by NYU for the ACE evaluations, as described in Section 2.3.1. Names, entities, and relations are detected. The name model was trained on 800K words, the nominal classifier on 600K words, and the relation model on about 90K words of ACE training data. More details about the NYU IE system can be found in [42]. The available speech data corresponded to documents used for the 2004 ACE evaluation, so we have adhered to the 2004 ACE specifications and scoring rules throughout. All IE results are given in terms of the entity value and relation value scores, as produced by the official ACE 2004 scorer (where higher scores are better). The scores include weighted penalties for missing items, spurious items, and feature errors in corresponding items; details are given in the ACE 2004 Evaluation Plan.¹ Scoring is based on offsets in the reference text.

8.2 Results

Three kinds of experimental setups are evaluated in this section: a baseline system, a system where punctuation is optimized for its own performance, and a system where punctuation prediction is tuned to optimize IE. The baselines compare the effect of reference punctuation (upper bound) and fixed-length punctuation (lower bound) on IE annotation. The reference punctuation is restricted to periods and commas (and periods only) while the fixed-length punctuation corresponds to chopping the word stream every 15 words (the mean sentence length). A no-boundary baseline would generate document-length sentences that can provoke system failure. In addition, the effect of ASR is compared to the reference words. Note that the reference words are flexible alignments from

¹<http://www.itl.nist.gov/iaui/894.01/tests/ace/ace04/doc/ace04-evalplan-v7.pdf>

the closed captions stripped from case information. They do not reflect the performances of the IE system on text data since the ACE reference has been designed on the closed-captions instead of the spoken words. Results are presented in Table 8.1, showing that both erroneous words and poor punctuation affect IE and that the effect is cumulative. Moving from reference to ASR words results in degradations of 10% or more. Removing commas has limited effect in entity extraction, but results in about 10% relative degradation in relation extraction for reference words, and nearly 20% relative degradation for ASR words. With reference words, removing reference sentence boundaries produces further degradation, but that is not the case for ASR words.

Table 8.1: Baseline IE performance for entities and relations on the test set. Various conditions are presented which include: reference words (Ref), machine generated words (ASR), reference punctuation (with and without commas) and fixed length punctuation.

Words	Punctuation	Entity	Relation
Ref	Reference	55.7	22.1
Ref	Ref. w/o commas	55.0	18.4
Ref	Pause Segments	54.9	18.8
Ref	Fixed length	50.2	16.9
ASR	Reference	46.9	20.0
ASR	Ref. w/o commas	47.9	16.4
ASR	Pause Segments	47.0	15.6
ASR	Fixed length	45.7	17.4

In order to verify the hypothesis that traditional maximization of punctuation classification performance is sub-optimal for the task of IE, the system is run for two conditions on ASR words. First, IE is performed on punctuation decisions resulting from maximizing F-measure for period and comma annotation on the development set. Then, the decision thresholds are chosen according to IE performance on the same set. The results are reported in Table 8.2 for the development and test sets and show that optimizing punctuation for IE is valuable because the entity score on the test set can be improved (compared to optimizing for punctuation F -measure) by 4% relative (significance level: $p < 0.06$), and the relation score can be improved by 4% relative ($p < 0.01$). However, the

Table 8.2: IE performance when punctuation is self-optimized (Punc.) or optimized in order to improve entities (Ent.) and relations (Rel.). Period and comma decision thresholds (thr_p , thr_c) are chosen in order to maximize performance on the development set and used blindly on the test set. Punctuation F -measure is reported in F_p and F_c for periods and commas, respectively. Comma F -measure is reported using reference sentence boundaries.

	Opt.	thr_p	thr_c	F_p	F_c	Ent.	Rel.
Dev.	Punc.	0.27	0.68	68.5	41.2	43.6	10.9
	Ent.	0.09	0.50	59.8	39.6	46.0	12.8
	Rel.	0.21	0.28	67.7	37.8	43.8	14.1
Test	Punc.	0.27	0.68	65.1	40.2	46.1	17.6
	Ent.	0.09	0.50	58.0	40.7	48.2	16.9
	Rel.	0.21	0.28	64.1	39.8	46.1	18.4

best setup for one task is not optimal for the other one, because the best thresholds differ depending on the task. In practice, a compromise point that is not best for either task (but is still better than optimizing independently for F -measure) could be selected.

Figure 8.1 shows IE performance according to comma and period posterior probabilities (each contour line represents a 0.2 drop from the highest score). An interesting result is that development and test conditions lead to quite similar parameter spaces, which is favorable for the robustness of IE-oriented optimization. The plots also make it easy to find a good compromise between optimal entity and relation scores. In general, IE performance is more sensitive to comma thresholds than to sentence boundary thresholds, especially for relation extraction.

8.3 Discussion

Similar to the effect on translation, IE may be influenced by very short or very long segments. Intuitively, shorter segments are more likely to break entities, especially if they involve long phrases. However, it is not obvious why fewer sentence boundaries would decrease the IE score. Therefore, we studied the output of the system and observed that one major effect of missed punctuation was noun-phrase (NP) merging. As illustrated by examples in Figure 8.2, if a sentence boundary is enclosed between NPs, it is likely that removing the period will confuse the NP chunker and merge

the NPs. In this case, the former NP can act as adjunct to the head of the latter NP. Similarly, removing commas can lead to undetected appositions and erroneous parsing because in written text the role of the comma is often to disambiguate the syntactic parse. In our opinion, these errors are partly due to the assumed presence of punctuation when developing syntactic analysis rules. We observed that in ACE reference data, 28% of entity mentions (18% of heads) are adjacent to a punctuation mark; one third of these interact with a comma and two thirds interact with a period (a few interact with both).

A more subtle but significant effect relates to the fact that a sentence-initial token is more likely² to be a name than a sentence-internal token. Consequentially, the HMM name tagger favors identifying tokens as names in sentence-initial position. In marginal cases, the tagger may correctly identify a name in sentence-initial position (after a period) but miss it elsewhere. Having fewer periods, therefore, may lead to missed names and a lower entity value score.

We also look at how punctuation over-generation affects entity mention splitting by computing the number of reference mentions split at different thresholds. Splitting a NP may result in two entity candidates from which at least one will affect performance (since heads are used for scoring). The curve in Figure 8.3 shows that choosing a lower threshold in order to reduce NP merging may result in more splits and not lead to IE improvement. In general, commas tend to lead to more entity splits, particularly for high comma confidence thresholds.

8.4 Conclusion

The work presented in this chapter focuses on automatically punctuating speech in order to improve information extraction. Predicting punctuation for the ASR transcript is beneficial in order to take better advantage of the large quantity of annotated textual data available to train IE (and lack of annotated speech data). We have shown that setting the punctuation decision thresholds to maximize punctuation performance is sub-optimal for IE. Moreover, improvements are obtained at different thresholds when annotating entities or relations. This suggests that punctuation should be generated differently, depending on the overall objective. An analysis of the results showed that punctuation errors can result in merged noun phrases or split entities. The former phenomenon can be traced to

²For our training corpus, roughly twice as likely.

syntactic parsing that usually requires accurate punctuation. As future work, we suggest improving the integration of speech related parameters in IE by, for example, optimizing ASR for parsing performance or adapting the parser to ill-punctuated content. In addition, rather than providing punctuation with fixed thresholds, joint punctuation/IE models may better predict entities by directly incorporating punctuation confidences as features in the IE model.

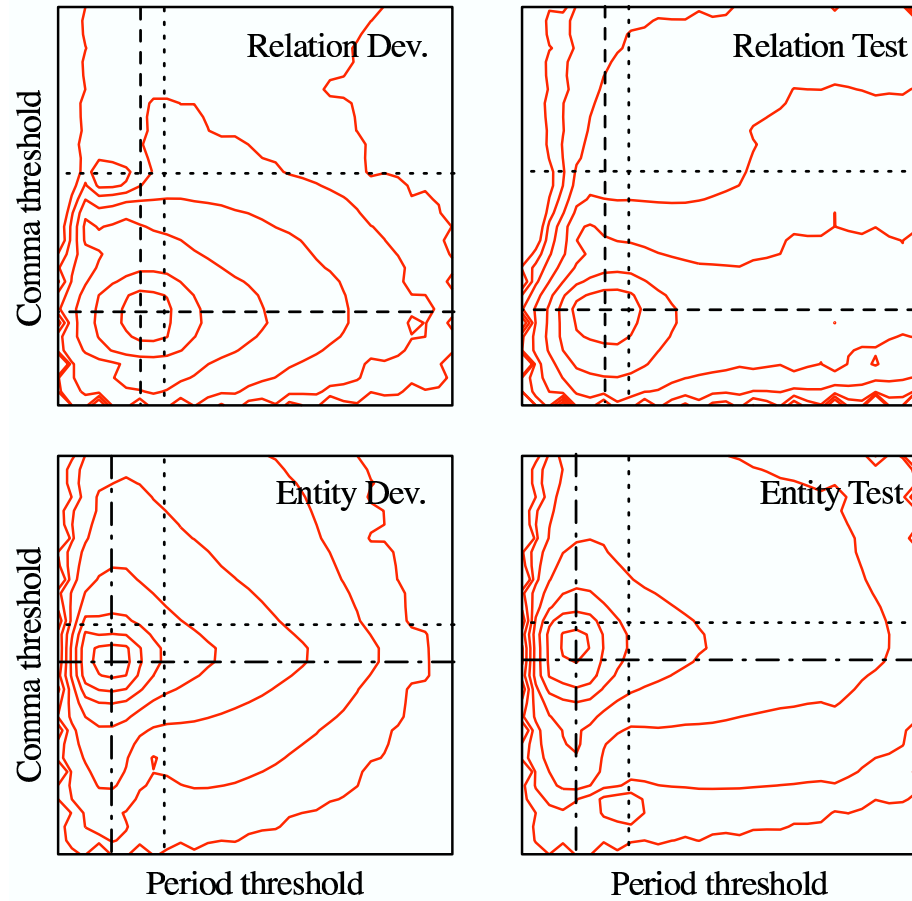


Figure 8.1: IE performance on entities and relations when period and comma thresholds are varied from 0 to 1 (from left to right and bottom to top). Contours are displayed every 0.2 point drop from the highest score (artifacts are created by undersampling). The punctuation-optimal thresholds are indicated by dotted lines, the entity-optimal thresholds by dash-dot lines, and the relation-optimal thresholds by dashed lines.

-
- (1) ... aides [NP his children]. [NP senators] ...
 ... aides [NP his children senators] ...
-
- (2) ... the president of [NP mexico vincente fox]
 ... the president of mexico, [NP vincente fox]
-

Figure 8.2: Examples where noun phrase assignment is ambiguous due to a missed sentence boundary (1) or comma (2). Even if semantically unlikely, the assignment is usually syntactically correct. Similarly, inserting a punctuation mark in the middle of a noun phrase will result in a split.

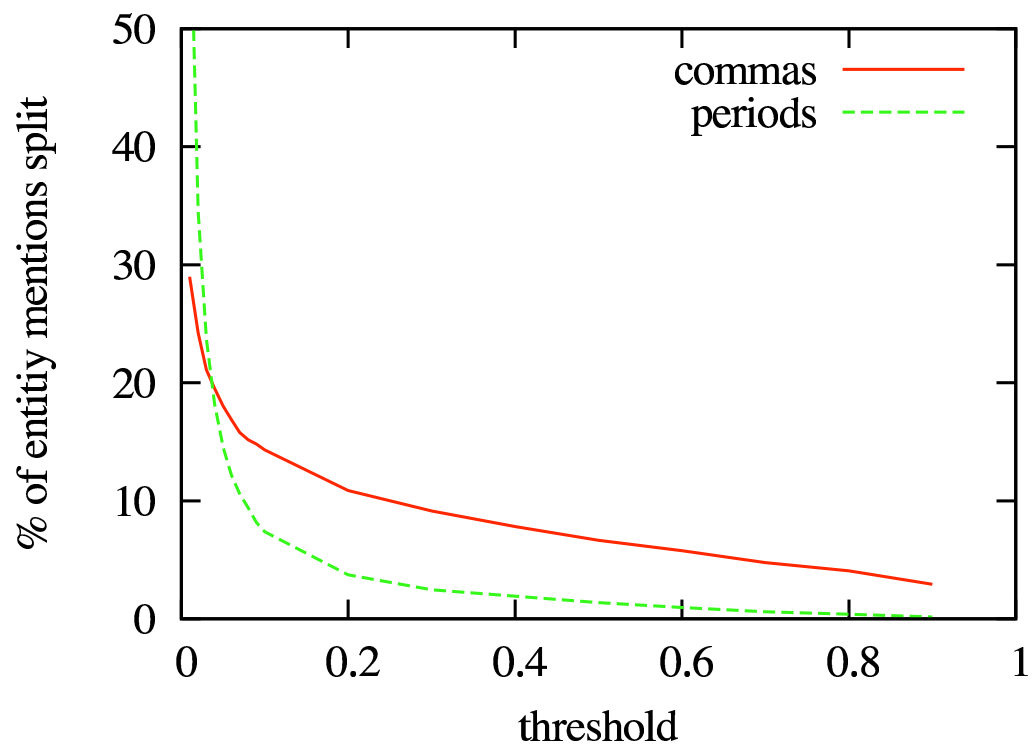


Figure 8.3: Percentage of reference entity mention extents split by inserting commas or periods at their respective decision thresholds.

Chapter 9

IMPROVING MT REORDERING WITH COMMAS

This chapter investigates the impact of sub-sentence punctuation on machine translation of automatically recognized speech. As discussed in previous chapters, standard speech recognition does not typically provide punctuation such as commas. Previous work in machine translation has found that automatic punctuation prediction on the source language is not the best method for improving punctuation performance for the target language. Usually, the best approach is to instead strip internal sentence punctuation from the source side, build phrase tables, and let the phrase translation system learn where to insert target language punctuation [102]. So, rather than focusing on source language punctuation prediction for the sake of improving target punctuation, we instead investigate approaches that leverage sentence structure implied by source language punctuation to improve overall translation quality.

Fluency is one aspect of automatic translations that still lags far behind human translations, and a prominent issue is poor reordering. While a human translator has little difficulty deciding how a translated sentence should be structured, and the correct word and phrase order, machine translation still has difficulty in correctly generating complex reorderings. Chinese to English translation is particularly challenging, with regard to reordering, because there are relatively more long range reorderings (when compared to other romance languages, or Arabic).

In Chapter 5, we found that Chinese to English translation improved when segmentation was tuned to generate shorter sentences. The shorter sentences helped to limit incorrect reordering hypotheses, leading to better translation quality, but at the cost of cutting lexical context where it might have otherwise been helpful in determining the best translation. Rather than tune sentence segmentation to produce shorter sentences, an alternative approach is to improve the reordering model. So, we instead develop modeling approaches that penalize unlikely reorderings which cross sub-sentence boundaries, while allowing longer sentences where the target language model can still utilize the lexical context around those boundaries.

To evaluate potential sub-sentence boundary candidates, we analyze a portion of the Chinese TreeBank that has been annotated with hand word alignments (where Chinese and English words have been linked in the reference parallel translation). We investigate the human word alignments to determine if commas act as plausible sub-sentence reordering boundaries. The results are presented in Table 9.1, where we found that very little translation reordering occurs across commas. For sentences that contain commas, less than 10% have a word reordering that crosses a source sentence comma. Therefore, when a sentence is divided into sections at comma boundaries, reordering is usually contained within those sections.

Table 9.1: Commas as reordering boundaries in a hand aligned section of the Chinese Treebank

total sentences	1865
sentences with commas	1093
sentences with alignments crossing commas	107

So, we turn to investigating the use of automatically predicted commas as soft boundary constraints for translation reordering, where we use the term “soft boundary” in part because there are exceptions to the “rule” that words do not reorder across commas and in part because there are comma prediction errors. We develop new approaches so that these “soft” boundaries can be used to penalize reordering in the MT search, while the phrasal context across these boundaries is still considered by the MT system.

The following sections describe two attempts at improving reordering in machine translation. The first (joint work with Evgeny Matusov [99]) inserts a new soft boundary penalty into translation decoding so that any reordering configuration that crosses a soft boundary is penalized (Section 9.1). The second (joint work with Evgeny Matusov, Richard Zens and Arne Mauser) extends a maximum entropy model that predicts phrase orientation based on source and target lexical features (Section 9.2). A feature that indicates when reorderings cross comma boundaries in the source sentence is added, which improves phrase orientation prediction.

9.1 *Soft Boundaries*

This section describes an approach for introducing a new score in translation decoding that penalizes reordering hypotheses that cross predicted soft-boundaries. First we describe our corpora and evaluation, followed by the sentence and soft-boundary prediction systems. We then describe the soft-boundary penalty and finally present results incorporating the approach into translation decoding.

9.1.1 *Corpora and Evaluation*

The baseline translation system for this section is the same as the system used in Section 5.1. We also evaluate on the same test sets, the broadcast news portion of the GALE MT 2006 evaluation data. MT quality is again assessed with BLEU [122] and TER [147] (as described in Section 2.3.2), in the same manner as for Section 5.1.

9.1.2 *Sentence and Soft Boundary Prediction*

In this work, commas are predicted using the approach described previously in Section 6.1, employing the same lexical and prosodic features. While comma and sentence boundary prediction could be treated jointly as a multi-class problem, here we take predicted sentence boundaries as given and then predict commas (without distinguishing between comma and caesura) within the sentence, as in the IE study of Chapter 8. We predict the probability of a comma after each word, and that confidence is passed on to the MT system if the confidence is above a minimum threshold (.2 for our experiments). We investigate three sentence boundary scenarios from Chapter 5: reference sentences, RWTH+ICSI (without the phraseLM feature), and ICSI+ with a 0.5 threshold. The interaction of commas and sentence boundaries leads to different precision/recall operating points in soft-boundary prediction, so we wish to evaluate our approach under varying conditions.

9.1.3 *Using Soft Boundaries in MT*

One of the features in the log-linear translation model in Section 5.1.2 is the reordering model. The reordering model of the baseline system is a distance-based model. It assigns costs based on the distance from the end position of a phrase to the start position of the next phrase; “jumps” over a

long distance are penalized. This very simple reordering model is widely used, for instance in [70]. For Chinese-to-English translation, this simple model is combined with a maximum entropy model predicting the probability of a phrase orientation class [181].

In this section, we extend the reordering model by an additional penalty, the *soft boundary penalty*. Reordering across a soft boundary is assumed to be highly unlikely (as found in the analysis of the Chinese Treebank) and is penalized. The soft boundaries described in Section 9.1.2 implicitly divide a source sentence into several parts. We develop two types of soft boundary penalties. The first is useful in the oracle case of hand labels, or if hard decisions are used. The second incorporates predicted soft boundary confidences.

In the first type, each word f_j at position j in a sentence is labeled with an integer label $c(j)$ which encodes the (soft boundary separated) section of the sentence that the word is from. We penalize the movement of a phrase from the position j to a position j' by a weight α if the two positions have different section labels:

$$w(j, j') = \alpha \cdot |c(j') - c(j)| \quad (9.1)$$

If we simply take the approach of Chapter 5 and translate each sentence part as if it were a separate sentence, we lose context across these boundaries. The soft boundary approach mitigates this issue by imposing a penalty for unlikely reorderings while still maintaining context across sub-sentence boundaries. Note that the penalty in Eq. 9.1 naturally increases in case the hypothesized phrase movement is across two, three, etc. boundaries, making reordering from the beginning to the end of a long sentence very unlikely. Given a text or a speech transcript with sub-sentence punctuation, we can consider commas to be soft boundaries and define the labels $c(j)$ accordingly.

In case of automatically predicted soft boundaries, we can use the posterior probability of a boundary to make the penalty dependent on the boundary confidence (our second type of penalty). For our experiments we use all commas with a confidence greater than 0.2 as soft boundaries. Boundaries with confidence near one will have similar effect to boundaries in our first penalty type, but for boundaries with lower confidence the penalty will be smaller since phrase reordering across this boundary may still be somewhat probable. Incorporating soft boundary confidence scores is straightforward: the labels $c(j)$ in Eq. 9.1 are replaced by real values $r(j)$. These are computed

recursively as follows:

$$r(j) = \begin{cases} 0, & \text{if } j = 0 \\ r(j-1), & \text{if } c(j) = c(j-1) \\ r(j-1) - \log p_{nb}(j) & \text{if } c(j) \neq c(j-1) \end{cases} \quad (9.2)$$

Here, $p_{nb}(j)$ is the posterior probability that the soft boundary *does not* appear between the words f_{j-1} and f_j . If the new position j' and the old position j of the first word in a phrase are in the same sentence part, no penalty will be added, since $r(j) - r(j') = 0$. We could also have implemented the penalty to just sum over p_b (probability that a soft boundary appears), but chose the framework of Eq. 9.1 because it is more similar to the other log probability scores in the MT model. An example is shown in Figure 9.1, where $c(j)$ is an integer boundary count, p_b is the probability of a comma boundary, and $r(j)$ sums over $-\log p_{nb}(j)$.

j	1	2	3	4	5	6	7	8	9	10	11
	Despite	the	criticism	they	continued	working	and	people	finally	liked	it
$c(j)$	0	0	0	1	1	1	2	2	2	2	2
p_b	0	0	0	.7	0	0	.8	0	0	0	0
$\log(p_{nb})$	0	0	0	-.52	0	0	-.70	0	0	0	0
$r(j)$	0	0	0	.52	.52	.52	1.22	1.22	1.22	1.22	1.22

Figure 9.1: Example of $c(j)$ and $r(j)$ reordering penalties. p_b is the probability (confidence) of a comma boundary, and p_{nb} is the probability of no comma boundary ($1 - p_b$)

9.1.4 MT Results for Soft Boundary Prediction

Table 9.2 presents the comma prediction and translation results for three settings. In the first setting, we used the integer penalties $c(j)$ as in Eq. 9.1. The penalties were computed relative to the

Table 9.2: Comma and translation results [%] for the different SU and soft boundary settings on the Chinese-to-English task.

SU algorithm	P	R	F-score	Baseline		Soft Boundary	
				BLEU	TER	BLEU	TER
reference SUs	100	100	100	20.7	66.9	20.8	66.9
RWTH+ICSI	73.8	35.6	48.0	20.8	67.1	20.7	67.1
ICSI+ 0.5	77.0	40.1	52.7	20.2	67.5	20.3	67.4

reference (oracle) commas and SU boundaries that we inserted into the ASR output. The second and third settings use automatically predicted commas and their posterior probabilities as in Eq. 9.2, which are inserted given the predicted SU boundaries of the RWTH+ICSI or the ICSI+ 0.5 threshold system. In the third setting, we used the somewhat longer SUs of the ICSI+ system at a threshold of 0.5, which resulted in using more automatically predicted commas (with higher comma recall). Comma recall increases for less frequent sentence boundaries because inserted SUs can often occur at reference comma locations.

In all cases, the BLEU and TER improvements were not significant with respect to the translation results in Table 5.2 without using the soft boundaries. We attribute this in part to the good quality of the baseline maximum entropy reordering model that already restricts unnecessary long-range phrase reordering. Nevertheless, in an anecdotal analysis of a subset of translated sentence that differed from the baseline translated sentences, word order and cause-effect relations were subjectively more correct when the soft boundary penalty was used. Examples are shown in Table 9.3.

9.2 Maxent models for reordering

Providing an additional soft boundary penalty in translation reordering did not provide measurable improvements as indicated by BLEU and TER MT evaluation criteria. One possible reason is that there are some syntactic contexts where reordering does occur across sub-sentence boundaries, in which case it would be useful to make the boundary a feature rather than a constraint. Therefore, we investigate an alternative approach that instead integrates boundary information directly as features

Table 9.3: Example of improved MT quality by using automatically predicted commas as soft boundaries (Chinese-to-English task).

baseline	according to statistics , in 2005 , the china national tourism administration ...
+commas	according to the china national tourism administration statistics in 2005 ...
reference	the statistics from the national tourism administration shows that in 2005 ...
baseline	the protesters , chanting slogans green belt ...
+commas	protesters circumspect green belt , shouted slogans ...
reference	the protesters , wearing green turbans , shouted slogans ...
baseline	after rapid reaction , the government mud-rock flows ...
+commas	mud-rock flows , the government has reacted ...
reference	after the mudslide broke out , the government responded ...

in a maximum entropy framework for modeling reordering.

9.2.1 Baseline phrase orientation model

While many translation systems incorporate simple costs for phrase movement that are linear in the distance of the move [117, 70], recent approaches have developed more detailed models for predicting phrase orientation, which predict where the next phrase belongs in relation to the current phrase [71, 164]. Here, our baseline model is the maximum entropy modeling approach of [181], as specified in Equation 9.3, where h_m is again a feature function, and λ_m a weight to be learned.

$$p(c_{j,j'} | f_1^J, e_1^I, i, j) = \frac{\exp \left(\sum_{m=1}^M \lambda_m h_m (f_1^J, e_1^I, i, j, c_{j,j'}) \right)}{\sum_{c'} \exp \left(\sum_{m=1}^M \lambda_m h_m (f_1^J, e_1^I, i, j, c') \right)} \quad (9.3)$$

The approach models the position of the next target phrase, where $c_{j,j'}$ represents the phrase orientation class for moving from source position j to source position j' , given features derived from both the source (f_1^J) and target (e_1^I) sentences (where word automatic word alignments for training). The modeling target is either two or four classes. The two class case predicts whether the next

target phrase is to the left ($j' < j$) or to the right ($j' > j$) of the current target phrase. The four class case predicts whether the next target phrase is: more than one position to the left, one position to the left, one position to the right, or more than one position to the right. The features for the baseline model are a collection of binary features on the source and target words (and potentially on the part of speech tag or class for those words). Using more features generally improves phrase orientation prediction, but our experiments find that improved phrase orientation prediction does not necessarily lead to improved translation performance. In practice, using only the last source word of the current phrase and the first source word of the next phrase is sufficient for modeling phrase orientation (although we also add target words for one experimental condition).

9.2.2 *Boundary features in phrase orientation models*

We extend the baseline reordering model of Section 9.2.1 to include features derived from boundaries in the source sentence. In addition to the baseline features which include source and target words, we include three additional binary features that encode if the first source word of the next phrase is from the previous, same, or next boundary section as the last word of the current phrase. The source sentence is divided into sections based upon either reference punctuation or hard decisions from automatically predicted commas (similar to the $c(j)$ labels in Section 9.1.3). The maximum entropy model is then retrained with the new boundary features included.

9.2.3 *Training and testing corpora*

The phrase orientation models are trained and tested on multiple Chinese-English parallel corpora. Initial experiments train with all the parallel corpora provided by LDC (about 3 million parallel sentences), and this set is then extended with Chinese-English data from the United Nations (UN), for a total of 7 million sentences. We hold out the first 100,000 training sentences for evaluating the phrase orientation prediction.

The Chinese-English word alignments used for training the phrase orientation model are the result of symmetrizing alignments from IBM Model 4 (with GIZA++) in both directions¹. Each word alignment is assigned an orientation class for maximum entropy model training, where one-to-many

¹The alignments were kindly provided by Richard Zens at RWTH, Aachen.

and many-to-many alignments have been simplified so that only one alignment link is considered. Results for phrase orientation are for reference words and punctuation, because (unlike the case in information extraction) commas have not been used as features for reordering in prior work in translation. The approach could also be used in text translation, so we evaluate with reference words and punctuation in order to assess potential gains while avoiding the effect of comma prediction error.

9.2.4 Phrase orientation prediction results

In this section we present classification results for the maxent phrase orientation models. The errors (in percent of alignment links incorrectly labeled) are compared between models with and without boundary features, for two-class and four-class phrase orientation settings.

Table 9.4 presents results for including the comma boundary feature for the LDC subset, as well as the full training set (including the UN data), in this case using reference commas. The comma boundary feature provides 30% relative improvement in the two-class phrase orientation modeling, compared to the baseline model. We also compare results using only source word features versus both source and target features in training the two-class model. Including target word features provides a small, but consistent gain in all experiments.

Table 9.4: Two-class phrase orientation prediction errors on Chinese-to-English newswire translations, where “LDC only” is the parallel corpora from LDC, and “with UN” includes the UN parallel corpora.

features	LDC only		with UN	
	baseline	with boundary	baseline	with boundary
source	13.89%	10.94%	13.65%	10.82%
source+target	12.60%	9.85%	12.24%	9.62%

Table 9.5 presents results for the four-class phrase orientation model. Including the comma boundary feature provides more than 10% relative improvement in this case, again with reference commas. This and later experiments omit the target features for simplicity.

Table 9.6 tests phrase orientation prediction using automatically detected commas (with a confi-

Table 9.5: Four-class phrase orientation prediction errors on Chinese-to-English newswire translations, where “LDC only” is the parallel corpora from LDC, and “with UN” includes the UN parallel corpora.

features	LDC only		with UN	
	baseline	with boundary	baseline	with boundary
source	39.72%	34.98%	39.42%	34.94%

dence threshold of 0.2). For experiments with reference punctuation above, punctuation is included as words (as in previous work, and for use in text translation), so punctuation is also included in the baseline lexical features. The baseline system for auto-commas does not include punctuation as words, so the error rate is higher (because punctuation is no longer available as a lexical feature, which degrades performance). In this setting, the improvements over the baseline model with no boundaries are smaller, but still present. In addition, when automatic commas are used in testing, we find that it is better to train on automatically generated commas rather than the true punctuation in the training data. Further experiments are required to select an optimal comma confidence threshold.

Table 9.6: Phrase orientation prediction error on Chinese-to-English newswire translations, testing with automatic commas (LDC corpora only).

orientation classes	training condition		
	baseline	auto commas	ref commas
two	15.55%	14.37%	14.48%
four	41.63%	40.24%	41.15%

Finally, we incorporate the improved phrase orientation in MT decoding, and compare against the baseline phrase orientation model using reference commas on two text test sets. We compare the 2 class orientation model with source and target words (baseline) to the 2 class orientation model with boundaries and source and target words (with boundaries). The orientation models trained on

the full training set (with UN) are used. The results are presented in Table 9.7. The system including boundary features is not significantly different for both test sets, but further experiments are required to re-optimize MT parameters for the new orientation model. Pilot experiments in speech translation of ASR output with automatic commas have also found no gains so far.

Table 9.7: MT BLEU score [%] on Chinese-to-English broadcast news reference words, with and without boundary features in phase orientation prediction.

test set	without target word features		with target word features	
	baseline	with boundaries	baseline	with boundaries
dev07	17.32	17.20	17.39	17.25
nist06	16.82	16.39	16.76	16.89

Comparing the translation outputs from the baseline system and the system with boundaries features (where both also include target word features), some qualitative improvements are evident, as in Table 9.8, where the baseline system misplaced the word “study” while the system with boundary features correctly kept “studied” in the right clause.

9.2.5 Discussion

While significant gains in phrase orientation prediction are possible when including boundary features in the reordering model, so far no improvements in BLEU score have been achieved when using the better phrase reordering models during MT decoding. Improved reordering from parse based features is a promising direction for future research.

9.3 Conclusion

In summary, this chapter finds that while preliminary improvements in reordering modeling for machine translation show promise, the approaches do not yet achieve measurable performance gains. Including an additional decoding penalty for reordering hypotheses that cross sub-sentence boundaries gave qualitative improvements in translations but does not yet result in better BLEU or TER. An alternative approach, which incorporates comma features in a maximum entropy reordering

Table 9.8: Example of improved MT quality by using commas from text as boundary features in phrase orientation modeling (Chinese-to-English task).

baseline	... he also economic theory, study of a very, very preliminary ...
+boundaries	... he also studied the economic theory, a very, very preliminary ...
reference	... he has also studied economic theory, researching a very forward ...

model, achieves more than 20% relative improvement in phrase orientation prediction. Incorporating the better phrase orientation model in translation decoding does not lead to higher quality translations as measured by BLEU scores when compared to a phrase orientation model without comma features.

Chapter 10

CONCLUSION

This chapter summarizes the main conclusions of the dissertation, comments on its impact, and proposes directions for future research.

10.1 Main Conclusions

The primary motivation of the dissertation is to improve spoken language processing with:

- Improved annotation of sentence and sub-sentence structure for speech
- Tighter coupling between speech recognition, segmentation, and downstream applications

The work is organized to address the two major types of segmentation: sentence and sub-sentence structure. The key findings in these areas are presented below.

10.1.1 Sentence segmentation

Automatic sentence segmentation for ASR degrades with increasing WER, and while a small portion of the loss may be recovered with approaches that consider N-Best lists (or joint decoding of words and sentence boundaries in ASR lattices), our experiments show that improvements in the top ASR hypothesis are more important to sentence segmentation tasks than considering multiple ASR hypotheses.

Our experiments showed that while the bias in ASR word confidence estimates can be significantly reduced by predicting the probability of missing words, there is no further benefit to system combination. However, the same type of classifier approach as used for improving confidence prediction can be used in system combination. Our *i*ROVER approach trains a classifier to select the word that is most likely to be correct based on many features from the component system lattices. The resulting classifier more reliably improves performance compared to other previously proposed

system combination methods. Reductions in WER lead directly to improvements in most downstream tasks.

When providing automatic sentence segmentation to speech translation, performance can be improved when the automatic sentence segmentation is optimized for translation performance, rather than only optimizing sentence segmentation on its own. Translation from Mandarin to English generally benefits from higher recall sentence boundaries which are shorter and serve to limit reordering errors in translation. When sentence segmentation considers the modeling constraints of the translation system, final translation quality improves, even compared with reference sentence boundaries. Mandarin-English translation BLEU scores improve by .5 or more from a baseline system of 20.2, when the sentence segmentation system is optimized for translation quality, rather than minimum sentence error. Similar, but smaller, improvements are found for Arabic-English translation as well. Imposing length constraints to ensure that only reasonable sentences are presented to the system is one improvement, but additional gains are possible by further augmenting the approach to penalize against inserting sentence boundaries at inter-word locations that frequently occur inside phrases. Finally, after selecting a segmentation, the choice of ASR words can be optimized at the sentence level towards well formed sentences (which have parse structure similar to the reference sentence). This provides hypotheses better suited for translation (even when the MT system does not explicitly use parses), and shows promise for improvements in translation performance. Additionally, the approach may have broader impact for other applications that will benefit from better parsing accuracy, such as information extraction.

10.1.2 Sub-sentence structure

Reliable automatic comma detection for English is described, and results for Mandarin comma and caesura detection are reported. In addition, unsupervised methods for detecting sub-sentence prosodic phrase boundaries and emphasis are developed and tested.

Automatically predicted commas for Mandarin speech recognition are shown to improve both part-of-speech and named entity tagging when compared to a system that does not have commas available. Automatic commas are able to recover about half of the performance loss attributed to removing human generated commas.

Further analysis of comma and sentence boundary optimization for English information extraction reinforces the Mandarin findings, while also investigating the links between sentence and sub-sentence events. In addition, the optimal operation point is found to be dependent on the specific task for which maximum performance is desired.

Translation quality benefits from optimizing segmentation, but for Mandarin the optimal sentences tend to be shorter, and therefore reduce the context available to the MT system. In order to recover context while still retaining the benefit of reordering restrictions, commas can be treated as soft reordering boundaries. A penalty is imposed for reorderings that cross source commas, and while no significant gains in BLEU or TER are found, qualitative assessments of the resulting translations are better than the baseline system. Alternatively, phrase reordering prediction can obtain 10-30% relative improvements in predicting reordering, but no significant gain in translation performance as measured by BLEU.

10.2 General Findings and Impact

With the goal of evaluating across a broad range of possible applications, we find that sentence and sub-sentence structure can make important contributions to many spoken language processing tasks. Part of speech tagging, named entity tagging, relation extraction, and machine translation all benefit from incorporating explicit sentence and sub-sentence segmentation, rather than relying only on pause-based segmentation or fixed length word sequences.

Segmentation, both sentence and sub-sentence, should be tuned towards the desired downstream application. Optimizing the precision/recall operating point in segmentation detection leads to better performance on spoken language processing when compared with only maximizing performance for segmentation tasks independently. In addition, incorporating information from downstream tasks can aid segmentation in avoiding harmful errors. For example, in machine translation, we develop a feature that penalizes segmentation boundaries that would break likely phrases.

Errors from ASR have a larger impact than on downstream processing than segmentation errors. We find that optimizing ASR for parsing performance is a promising approach with potential to improve ASR in ways that are not captured by measuring WER but that can benefit spoken language processing. Sentence structure can still have significant impact (particularly because segmentation

can impact both ASR and parsing), and a pipeline that considers uncertainty and errors in all components should outperform approaches that just optimize each component independently. While it is impractical to optimize all model parameters jointly (in particular because not all training corpora are used for all model components), it is not difficult to tune a small number of feature weights or thresholds on a development set to improve overall performance.

10.3 Future Research

Future research is likely to benefit most from further integration between ASR, sentence structure annotation, and downstream processing. Improved modeling of prosody in speech could also provide a new source of cues for further improvements.

10.3.1 Improving ASR for downstream processing

Approaches that optimize ASR output for downstream tasks are a promising research direction. While some gains are already evident when optimizing ASR with more weight on sentence structure (parsing), experiments so far have mostly adjusted parameters that already exist in the ASR system. Expanding the optimization to incorporate additional information sources, such as the parse features utilized in [104] could likely lead to moving ASR towards the goal of a well structured sentence that is more suitable for automatic translation and other natural language processing. In addition, parse features most relevant to specific tasks (such as noun phrases for IE, or major clauses for MT) could be emphasized.

The system combination approach of Chapter 3 can also be adapted to directly optimize for other metrics beyond WER. The classifier that chooses system words with a target of minimum WER could instead be trained to select system words that directly minimize SParseval, or the error of a downstream system (such as minimizing TER for a translation system). Additional features that inform ASR about MT models could also bring improvements, such as bringing MT phrase probabilities into the language modeling aspect of the ASR system.

10.3.2 Integrating segmentation with spoken language processing

Sentence segmentation of ASR output for downstream applications likewise remains a direction for further exploration. While most of our experiments find that shorter sentences are better suited to many spoken language processing tasks, new methods of incorporating sub-sentence structure into downstream models could lead towards improved performance with longer, more complex, sentences. Including parse information is another promising direction for automatic processing of sentences with complex structure. Because sentence segmentation has a large impact on parsing performance [65, 44], it will likely be an important component of improving parse quality for ASR output.

While it is clear that automatic commas are helpful for information extraction tasks, the optimal precision/recall point differs depending on the task. Additional investigation into the most appropriate level of comma detection is an important direction for determining the optimal approach for frameworks that may require commas at differing recall/precision trade-off points. Complicated systems with multiple contributing components may require different amounts of segmentation at different stages. For name and entity extraction, joint comma/IE modeling would be an interesting direction to pursue for addressing these issues. Including boundaries as features in spoken language processing may be a direction that could avoid problems with multiple thresholds. Incorporating posteriors for commas or other types of segmentation (such as prosodic boundaries) also makes it possible to treat these as hidden events in a search for the most likely tagging sequence or parse. Parse and prosodic structure features could also provide helpful cues when multiple levels of segmentation for different types of downstream processing are required.

10.3.3 Learning prosodic structure

Much of the natural structure in spoken documents is communicated prosodically and not always present in the lexical content (while lexical cues to structure are often prosodically reinforced). Little prosodically annotated data is available, so unsupervised and semi-supervised learning approaches are important research directions for leveraging prosody, particularly for non-English languages that tend to have fewer resources. Evaluating automatically detected prosody may be challenging for domains and languages with limited hand-labeled data, but the implications of the studies in

this dissertation are that a more appropriate direction would be to move towards directly measuring performance on downstream tasks.

Prosodic breaks provide cues to sentence boundaries, but more importantly to sub-sentential phrasing. Automatically detected sub-sentence phrases would likely provide additional information to information extraction models, and could be useful in selecting snippets of speech for automatic question answering [96].

Finally, prosodically emphasized words can indicate important words that might be flagged for additional attention in translation or information retrieval on spoken documents. Prosodic emphasis could also contribute to an alternative ASR optimization approach that might give greater preference to prosodically stressed words, which are likely to carry more information, and therefore be more important to downstream tasks [18].

BIBLIOGRAPHY

- [1] Global Autonomous Language Exploitation (GALE). <http://www.darpa.mil/ipto/programs/gale/>.
- [2] NIST Speech Recognition Scoring Toolkit (SCTK). <http://www.nist.gov/speech/tools>.
- [3] TC-STAR, Technology and Corpora for Speech-to-Speech Translation Components, <http://www.tc-star.com>.
- [4] S. Abney. Chunks and dependencies: Bring processing evidence to bear on syntax. *Computational Linguistics and the Foundations of Linguistic Theory*, pages 145–164, 1995.
- [5] M. Agnas et al. *Spoken Language Translator: First-Year Report*. Joint SRI/SICS technical report, 1994.
- [6] Y. Al-Onaizan and L. Mangu. Arabic ASR and MT integration for GALE. In *Proc. ICASSP*, pages 1285–1288, 2007.
- [7] J. Allen and M. Core. Coding dialogs with the DAMSL annotation scheme. In *working notes of the AAAI Fall 1997 Symposium on Communicative Action in Humans and Machines*, pages 28–35, November 1997.
- [8] S. Ananthakrishnan and S. Narayanan. Combining acoustic, lexical, and syntactic evidence for automatic unsupervised prosody labeling. In *Proc. Interspeech*, pages 829–832, 2006.
- [9] S. Ananthakrishnan and S. Narayanan. Automatic prosodic event detection using acoustic, lexical, and syntactic evidence. *IEEE Transactions on Audio, Speech and Language Processing*, 16(1):216–228, 2007.
- [10] D. Baron et al. Automatic punctuation and disfluency detection in multi-party meetings using prosodic and lexical cues. In *Proc. ICSLP*, pages 949–952, 2002.
- [11] A. Batliner, R. Kompe, A. Kießling, M. Mast, H. Niemann, and E. Nöth. M = syntax + prosody: A syntactic-prosodic labelling scheme for large spontaneous speech databases. *Speech Communication*, 25(4):193–222, 1998.
- [12] D. Beeferman, A. Berger, and J. Lafferty. Cyberpunc: a lightweight punctuation annotation system for speech. In *Proc. ICASSP*, pages 689–692, 1998.

- [13] F. Vanden Berghen and H. Bersini. CONDOR, a new parallel, constrained extension of powell's UOBYQA algorithm: Experimental results and comparison with the DFO algorithm. *Journal of Computational and Applied Mathematics*, 181:157–175, 2005.
- [14] N. Bertoldi, R. Zens, and M. Federico. Speech translation by confusion network decoding. In *Proc. ICASSP*, pages 1297–1300, 2007.
- [15] S. Bhagat, H. Carvey, and E. Shriberg. Automatically generated prosodic cues to lexically ambiguous dialog acts in multi-party meetings. In *ICPhS*, pages 2961–2964, 2003.
- [16] D. Bikel, S. Miller, R. Schwartz, and R. Weischedel. Nymble: a high-performance learning name-finder. In *Proc. Conference on Applied Natural Language Processing*, pages 194–201, 1997.
- [17] E. Black, S. Abney, D. Flickenger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. Marcus, S. Roukos, B. Santorini, and T. Strzalkowski. A procedure for quantitatively comparing syntactic coverage of English grammars. In *Proc. 4th DARPA Speech & Natural Lang. Workshop*, pages 306–311, 1991.
- [18] C. Boulis. *Topic Learning in Text and Conversational Speech*. PhD thesis, University of Washinton, 2005.
- [19] L. Breiman et al. *Classification And Regression Trees*. Wadsworth International Group, Belmont, CA, 1984.
- [20] P. Brown, S. Della Pietra, V. Della Pietra, and R. L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, June 1993.
- [21] F. Brugnara et al. The ITC-IRST transcription systems for the TC-STAR-06 evaluation campaign. In *Proc. TC-STAR Workshop*, pages 117–122, 2006.
- [22] J. G. Carbonell, Y. Geng, and J. Goldstein. Automated query-relevant summarization and diversity-based reranking. In *IJCAI-97 Workshop on AI and Digital Libraries*, pages 9–14, 1997.
- [23] C. Chelba. *Exploiting syntactic structure for natural language modeling*. PhD thesis, Johns Hopkins University, Maryland, 2000.
- [24] C. Chelba and A. Acero. Position specific posterior lattices for indexing speech. In *Proc. of ACL*, pages 443–450, 2005.
- [25] C. J. Chen. Speech recognition with automatic punctuation. In *Proc. Eurospeech*, pages 447–450, 1999.

- [26] K. Chen, M. Hasegawa-Johnson, and A. Cohen. An automatic prosody labeling system using ANN-based syntactic-prosodic model and GMM based acoustic-prosodic model. In *Proc. ICASSP*, pages 509–512, 2004.
- [27] T. K. Chia, H. Li, and H. T. Ng. A statistical language modeling approach to lattice-based spoken document retrieval. In *Proc. EMNLP*, pages 810–818, 2007.
- [28] H. Christensen, Y. Gotoh, and S. Renals. Punctuation annotation using statistical prosody models. In *Proc. ISCA Workshop on Prosody in Speech Recognition and Understanding*, pages 35–40, 2001.
- [29] J. Chu-Carroll. A statistical model for discourse act recognition in dialogue interactions. In *Applying Machine Learning to Discourse Processing. Papers from the 1998 AAAI Spring Symposium*, pages 12–17, 1998.
- [30] M. Collins and T. Koo. Discriminative reranking for natural language parsing. *Computational Linguistics*, 31(1):25–70, 2005.
- [31] M. Collins, M. Saraclar, and B. Roark. Discriminative syntactic language modeling for speech recognition. In *Proc. of ACL*, pages 507–514, 2005.
- [32] M. Diab, K. Hacioglu, and D. Jurafsky. Automatic tagging of Arabic text: From raw text to base phrase chunks. In *Proc. HLT-NAACL*, pages 149–152, 2004.
- [33] M. Magimai Doss, D. Hakkani-Tur, O. Cetin, E. Shriberg, J. Fung, and N. Mirghafori. Entropy based classifier combination for sentence segmentation. In *Proc. ICASSP*, pages 189–192, 2007.
- [34] V. Doumpiotis and W. Byrne. Pinched lattice minimum Bayes risk discriminative training for large vocabulary continuous speech recognition. In *Proc. ICSLP*, pages 776–779, 2004.
- [35] G. Evermann and P. Woodland. Posterior probability decoding, confidence estimation and system combination. In *NIST Speech Transcription Workshop*, 2000.
- [36] M. Fach. A comparison between syntactic and prosodic phrasing. In *Proc. Eurospeech*, pages 527–530, 1999.
- [37] B. Favre, R. Grishman, D. Hillard, H. Ji, D. Hakkani-Tur, and M. Ostendorf. Punctuating speech for information extraction. In *Proc. ICASSP*, 2008.
- [38] J.G. Fiscus. A post-processing system to yield reduced word error rates: Recognizer Output Voting Error Reduction (ROVER). In *Proc. ASRU*, pages 347–352, 1997.

- [39] F. Gallwitz, H. Niemann, E. Nöth, and V. Warnke. Integrated recognition of words and prosodic phrase boundaries. *Speech Communication*, 36(1–2):81–95, 2000.
- [40] V. Goel and W.J. Byrne. Minimum Bayes-risk automatic speech recognition. *Computer Speech and Language*, 14:115–136, 2000.
- [41] V. Goel, S. Kumar, and W.J. Byrne. Segmental minimum Bayes-risk decoding for automatic speech recognition. *IEEE Transactions on Speech and Audio Processing*, 12:234 – 249, 2004.
- [42] R. Grishman, D. Westbrook, and A. Meyers. NYU’s English ACE 2005 System Description. In *Proc. of ACE 2005 Workshop*, 2005.
- [43] U. Guz, S. Cuendet, D. Hakkani-Tür, and G. Tur. Co-training using prosodic and lexical information for sentence segmentation. In *Proc. Interspeech*, pages 2597–2600, 2007.
- [44] M. Harper, B. Dorr, B. Roark, J. Hale, Z. Shafran, Y. Liu, M. Lease, M. Snover, L. Young, R. Stewart, and A. Krasnyanskaya. Parsing speech and structural event detection. John Hopkins University Summer Workshop Final Report, 2005.
- [45] M. Hasegawa-Johnson, J. Cole, K. Chen, P. Lal, A. Juneja, T. Yoon, S. Borys, and X. Zhuang. Prosodically organized automatic speech recognition. In *Proc. Linguistic Processes in Spontaneous Speech*, 2006.
- [46] D. Hillard, B. Hoffmeister, M. Ostendorf, R. Schülter, and H. Ney. iROVER: Improving system combination with classification. In *Proc. HLT-NAACL*, pages 65–68, 2007.
- [47] D. Hillard, Z. Huang, H. Ji, R. Grishman, D. Hakkani-Tur, M. Harper, and M. Ostendorf. Impact of automatic comma prediction on POS/name tagging of speech. In *Proc. of SLT*, pages 58–61, 2006.
- [48] D. Hillard, M.Y. Hwang, M. Harper, and M. Ostendorf. Parsing-based objective functions for speech recognition in translation applications. In *Proc. ICASSP*, 2008.
- [49] D. Hillard and M. Ostendorf. *Scoring structural MDE: Towards more meaningful error rates*. presented at NIST RT04 Workshop, November 2004.
- [50] D. Hillard and M. Ostendorf. Compensating for word posterior estimation bias in confusion networks. In *Proc. ICASSP*, pages 1153–1156, 2006.
- [51] D. Hillard, M. Ostendorf, A. Stolcke, Y. Liu, and E. Shriberg. Improving automatic sentence boundary detection with confusion networks. In *Proc. HLT-NAACL*, pages 69–72, May 2004.
- [52] B. Hoffmeister, D. Hillard, S. Hahn, R. Schülter, M. Ostendorf, and H. Ney. Cross-site and intra-site ASR system combination: Comparisons on lattice and 1-best methods. In *Proc. ICASSP*, pages 1145–1148, 2007.

- [53] B. Hoffmeister, T. Klein, R. Schlüter, and H. Ney. Frame based system combination and a comparison with weighted ROVER and CNC. In *Proc. ICSLP*, pages 537–540, 2006.
- [54] J. Horlock and S. King. Discriminative methods for improving named entity extraction on speech data. In *Proc. Eurospeech*, pages 2765–2768, 2003.
- [55] J. Huang and G. Zweig. Maximum entropy model for punctuation annotation from speech. In *Proc. ICSLP*, pages 917–920, 2002.
- [56] Z. Huang, M. Harper, and W. Wang. Mandarin part-of-speech tagging and discriminative reranking. In *Proc. of EMNLP-CoNLL*, pages 1093–1102, 2007.
- [57] M. Hwang, X. Lei, W. Wang, and T. Shinozaki. Investigation on Mandarin Broadcast News Speech Recognition. In *Proc. ICSLP*, pages 1233–1236, 2006.
- [58] M.Y. Hwang, G. Peng, W. Wang, A. Faria, A. Heidel, and M. Ostendorf. Building a highly accurate Mandarin speech recognizer. In *Proc. ASRU*, pages 490–495, 2007.
- [59] G. Ji. *Backoff Model Training using Partially Observed Data: Application to Dialog Act Tagging*. University of Washington, Electrical Engineering, Technical Report, 2005.
- [60] H. Jing, N. Kambhatla, and S. Roukos. Extracting social networks and biographical facts from conversational speech transcripts. In *Proc. of ACL*, pages 1040–1047, Prague, Czech Republic, June 2007.
- [61] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proc. European Conference on Machine Learning*, pages 137–142, 1998.
- [62] D. Jurafsky et al. Automatic detection of discourse structure for speech recognition and understanding. In *Proc. IEEE Workshop on Speech Recognition and Understanding*, pages 88–95, 1997.
- [63] J. Kahn. Moving beyond the lexical layer in parsing conversational speech. Master’s thesis, University of Washinton, 2005.
- [64] J. Kahn, M. Lease, E. Charniak, M. Johnson, and M. Ostendorf. Effective use of prosody in parsing conversational speech. In *Proc. HLT-NAACL*, pages 233–240, 2005.
- [65] J. Kahn, M. Ostendorf, and C. Chelba. Parsing conversational speech using enhanced segmentation. In *Proc. HLT-NAACL*, pages 125–128, 2004.
- [66] J. G. Kahn, D. Hillard, M. Ostendorf, and W. McNeill. *Joint optimization of parsing and word recognition with automatic segmentation*. Technical Report, University of Washington, 2007.

- [67] S. Kanthak and H. Ney. FSA: An efficient and flexible C++ toolkit for finite state automata using on-demand computation. In *Proc. of ACL*, pages 510–517, 2004.
- [68] J.-H. Kim and P. Woodland. The use of prosody in a combined system for punctuation generation and speech recognition. In *Proc. Eurospeech*, pages 2757–2760, 2001.
- [69] K. Kirchhoff, O. Rambow, N. Habash, and M. Diab. Semi-automatic error analysis for large-scale statistical machine translation. In *Proc. of the MT Summit XI*, pages 289–296, 2007.
- [70] P. Koehn. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. In *Conf. of the Association for Machine Translation in the Americas*, pages 115–124, Washington DC, 2004.
- [71] P. Koehn, A. Axelrod, A. B. Mayne, M. Osborne C. Callison-Burch, and D. Talbot. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *Proc. IWSLT*, 2005.
- [72] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. MOSES: Open source toolkit for statistical machine translation. In *Proc. of ACL*, page demonstration session, 2007.
- [73] L. Lamel et al. The LIMSI 2006 TC-STAR transcription systems. In *Proc. TC-STAR Workshop*, pages 123–128, 2006.
- [74] A. Lavie, D. Gates, N. Coccaro, and L. Levin. Input segmentation of spontaneous speech in janus: A speech-to-speech translation system. In *ECAI '96: Workshop on Dialogue Processing in Spoken Language Systems*, pages 86–99, London, UK, 1996. Springer-Verlag.
- [75] A. Lavie, L. Levin, Y. Qu, A. Waibel, D. Gates, M. Gavalda, L. Mayfield, and M. Taboada. Dialogue processing in a conversational speech translation system. In *Proc. ICSLP*, pages 554–557, 1996.
- [76] Y. Lee. IBM Arabic-to-English Translation for IWSLT 2006. In *Proc. IWSLT*, pages 45–52, November 2006.
- [77] Y. Lee, Y. Al-Onaizan, K. Papineni, and S. Roukos. IBM spoken language translation system. In *Proc. IWSLT*, pages 13–18, 2006.
- [78] X. Lei, M. Siu, M.Y. Hwang, M. Ostendorf, and T. Lee. Improved tone modeling for mandarin broadcast news speech recognition. In *Proc. ICSLP*, pages 237–242, 2006.
- [79] L. Levin, A. Lavie, M. Woszczyna, D. Gates, M. Gavalda, D. Koll, and A. Waibel. The Janus-III translation system: Speech-to-speech translation in multiple domains. *Machine Translation*, 15(1–2):3–25, 2000.

- [80] M. Levit, D. Hakkani-Tür, and G. Tur. Integrating several annotation layers for statistical information distillation. In *Proc. ASRU*, pages 671–676, 2007.
- [81] G. Levow. Unsupervised and semi-supervised learning of tone and pitch accent. In *Proc. HLT-NAACL*, pages 224–231, 2006.
- [82] P. Liu, J.-L. Zhou, and F. Soong. Background model based posterior probability for measuring confidence. In *Proc. Eurospeech*, pages 1465–1468, 2005.
- [83] Y. Liu et al. MDE Research at ICSI+SRI+UW, NIST RT-03F Workshop. <http://www.nist.gov/speech/tests/rt/rt2003/fall/presentations/>, 2003.
- [84] Y. Liu et al. The ICSI-SRI-UW metadata extraction system. In *Proc. ICSLP*, pages 577–580, 2004.
- [85] Y. Liu et al. *Resampling Techniques for SU Detection: A Case Study in Machine Learning from Imbalanced Data for Spoken Language Processing*. Technical Report, ICSI, 2004.
- [86] Y. Liu and E. Shriberg. Comparing evaluation metrics for sentence boundary detection. In *Proc. ICASSP*, pages 451–458, 2007.
- [87] Y. Liu, E. Shriberg, and A. Stolcke. Automatic disfluency identification in conversational speech using multiple knowledge sources. In *Proc. Eurospeech*, volume 1, pages 957–960, 2003.
- [88] Y. Liu, E. Shriberg, A. Stolcke, D. Hillard, M. Ostendorf, and M. Harper. Enriching speech recognition with sentence boundaries and disfluencies. *IEEE Trans. Audio, Speech and Language Processing*, 14(5):1526–1540, 2006.
- [89] Y. Liu, E. Shriberg, A. Stolcke, B. Peskin, J. Ang, D. Hillard, M. Ostendorf, M. Tomalin, P. Woodland, and M. Harper. Structural meta-data research in the EARS program. In *Proc. ICASSP*, pages 957–960, 2005.
- [90] Y. Liu, A. Stolcke, E. Shriberg, and M. Harper. Using conditional random fields for sentence boundary detection in speech. In *Proc. of ACL*, pages 451–458, 2005.
- [91] K. Livescu, O. Cetin, M. Hasegawa-Johnson, S. King, C. Bartels, N. Borges, A. Kantor, P. Lal, L. Yung, A. Bezman, S. Dawson-Haggerty, B. Woods, J. Frankel, M. Magimai-Doss, and K. Saenko. Articulatory feature-based methods for acoustic and audio-visual speech recognition: Summary from the 2006 JHU summer workshop. In *Proc. ICASSP*, pages 621–624, 2007.
- [92] J. Löff et al. The 2006 RWTH parliamentary speeches transcription system. In *Proc. ICSLP*, pages 105–108, 2006.

- [93] J. Makhoul, A. Baron, I. Bulyko, et al. The effects of speech recognition and punctuation on information extraction performance. In *Proc. Interspeech*, pages 57–60, 2005.
- [94] J. Makhoul, A. Baron, I. Bulyko, L. Nguyen, L. Ramshaw, D. Stallard, R. Schwartz, and B. Xiang. The effects of speech recognition and punctuation on information extraction performance. In *Proc. Eurospeech*, pages 57–60, 2005.
- [95] L. Mangu, E. Brill, and A. Stolcke. Finding consensus in speech recognition: word error minimization and other applications of confusion networks. *Computer Speech and Language*, 14:373–400, 2000.
- [96] S. Maskey and J. Hirschberg. Comparing lexical, acoustic/prosodic, structural and discourse features for speech summarization. In *Proc. Interspeech*, pages 621–624, 2005.
- [97] M. Mast, R. Kompe, S. Harbeck, A. Kiessling, and V. Warnke. Dialog act classification with the help of prosody. In *Proc. ICSLP*, pages 1732–1735, 1996.
- [98] S. Matsoukas, I. Bulyko, B. Xiang, K. Nguyen, R. Schwartz, and J. Makhoul. Integrating speech recognition and machine translation. In *Proc. ICASSP*, pages 1281–1284, 2007.
- [99] E. Matusov, D. Hillard, M. Magimai-Doss, D. Hakkani-Tur, M. Ostendorf, and H. Ney. Improving speech translation with automatic boundary prediction. In *Proc. Interspeech*, pages 2449–2452, 2007.
- [100] E. Matusov, S. Kanthak, and H. Ney. On the integration of speech recognition and statistical machine translation. In *Proc. Eurospeech*, pages 467–474, 2005.
- [101] E. Matusov, G. Leusch, O. Bender, and H. Ney. Evaluating machine translation output with automatic sentence segmentation. In *Proc. IWSLT*, pages 148–154, 2005.
- [102] E. Matusov, A. Mauser, and H. Ney. Automatic sentence segmentation and punctuation prediction for spoken language translation. In *Proc. IWSLT*, pages 158–165, Kyoto, Japan, November 2006.
- [103] A. Mauser, R. Zens, E. Matusov, S. Hasan, and H. Ney. The RWTH Statistical Machine Translation System for the IWSLT 2006 Evaluation. In *Proc. IWSLT*, pages 103–110, Kyoto, Japan, November 2006.
- [104] W. McNeill, J. Kahn, D. Hillard, and M. Ostendorf. Parse structure and segmentation for improving speech recognition. In *Proc. of SLT*, pages 90–93, 2006.
- [105] D. Miller, R. Schwartz, R. Weischedel, and R. Stone. Named entity extraction from broadcast news. In *Proc. DARPA Broadcast News Workshop*, pages 37–40, 1999.

- [106] N. Morgan et al. The meeting project at ICSI. In *Proc. HLT-NAACL*, pages 246–252, 2001.
- [107] T. Morimoto, M. Suzuki, T. Takezawa, G. Kikui, M. Nagata, and M. Tomokiyo. A spoken language translation system: SI-trans2. In *Proc. of ACL*, pages 1048–1052, 1992.
- [108] M. Nakano, N. Miyazaki, J. Hirasawa, K. Dohsaka, and T. Kawabata. Understanding unsegmented user utterances in real-time spoken dialogue systems. In *Proc. of ACL*, pages 200–207, 1999.
- [109] S. Narayanan, S. Ananthakrishnan, R. Belvin, E. Ettaile, S. Ganjavi, P. Georgiou, C. Hein, S. Kadambe, K. Knight, D. Marcu, H. Neely, N. Srinivasamurthy, D. Traum, , and D. Wang. Transonics: A speech-to-speech system for English-Persian interactions. In *Proc. ASRU*, pages 670–675, 2003.
- [110] A. Nenkova, J. Brenier, A. Kothari, S. Calhoun, L. Whitton, D. Beaver, and D. Jurafski. To memorize or to predict: Prominence labeling in conversational speech. In *Proc. HLT-NAACL*, pages 9–16, 2007.
- [111] H. Ney, S. Niessen, F.J. Och, H. Sawaf, C. Tillmann, and S. Vogel. Algorithms for statistical translation of spoken language. *IEEE Trans. Speech and Audio Processing*, 8(1):24–36, 2000.
- [112] NIST. The 2001 NIST Evaluation Plan for Recognition of Conversational Speech over the Telephone. http://www.nist.gov/speech/tests/ctr/h5_2001/h5-01v1.1.pdf, 2000.
- [113] NIST. RT-03S Workshop Agenda and Presentations. <http://www.nist.gov/speech/tests/rt/rt2003/spring/presentations/>, 2003.
- [114] E. Noeth, A. Batliner, A. Kiessling, R. Kompe, and H. Niemann. Verbmobil: The use of prosody in the linguistic components of a speech understanding system. *IEEE Trans. Speech and Audio Processing*, 8(5):519–532, 2000.
- [115] F. Och. Minimum error rate training in statistical machine translation. In *Proc. of ACL*, pages 160–167, 2003.
- [116] F. Och and H. Ney. Discriminative training and maximum entropy models for statistical machine translation. In *Proc. of ACL*, pages 295–302, July 2002.
- [117] F. J. Och, C. Tillmann, and H. Ney. Improved alignment models for statistical machine translation. In *Proc. Joint SIGDAT Conf. on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 20–28, 1999.
- [118] M. Ostendorf, P. J. Price, and S. Shattuck-Hufnagel. *The Boston University radio news corpus*. Technical Report ECS-95-001, Boston University, 1995.

- [119] M. Ostendorf and K. Ross. A multi-level model for recognition of intonation labels. *Computing Prosody*, pages 291–308, 1997.
- [120] Mari Ostendorf, I. Shafran, Stefanie Shattuck-Hufnagel, L. Carmichael, and W. Byrne. A prosodically labeled database of spontaneous speech. In *Proc. ISCA Tutorial and Research Workshop on Prosody in Speech Recognition and Understanding*, pages 119–121, 2001.
- [121] D. Palmer, M. Ostendorf, and J. Burger. Robust information extraction from automatically generated speech transcriptions. *Speech Communication*, 32(1-2):95–109, 2000.
- [122] K. Papineni, S. Roukos, T. Ward, and W. Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proc. of ACL*, pages 311–318, July 2002.
- [123] S. Petrov and D. Klein. Improved inference for unlexicalized parsing. In *Proc. HLT-NAACL*, pages 404–411, 2007.
- [124] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1998.
- [125] B. Ramabhadran et al. The IBM 2006 speech transcription system for European parliamentary speeches. In *Proc. ICSLP*, pages 1225–1228, 2006.
- [126] N. Reithinger and M. Klesen. Dialogue act classification using language models. In *Proc. Eurospeech*, pages 2235–2238, 1997.
- [127] S. Renals and Y. Gotoh. Integrated transcription and identification of named entities in broadcast speech. In *Proc. Eurospeech*, pages 1039–1042, 1999.
- [128] S. Reynolds and J. Bilmes. Part-of-speech tagging using virtual evidence and negative training. In *Proc. HLT-NAACL*, pages 459–466, 2005.
- [129] K. Ries and A. Waibel. Activity detection for informal access to oral communication. In *Proc. HLT-NAACL*, pages 1–6, 2001.
- [130] B. Roark, M. P. Harper, E. Charniak, B. Dorr, M. Johnson, J. Kahn, Y. Liu, M. Ostendorf, J. Hale, A. Krasnyanskaya, M. Lease, I. Shafran, M. Snover, R. Stewart, and L. Yung. SParseval: Evaluation metrics for parsing speech. In *Proc. LREC*, 2006.
- [131] B. Roark, Y. Liu, M. Harper, et al. Reranking for sentence boundary detection in conversational speech. In *Proc. ICASSP*, pages 57–60, 2006.
- [132] B. Roark, M. Saraclar, and M. Collins. Discriminative n -gram language modeling. *Computer Speech and Language*, 21(2):373–392, April 2007.

- [133] M. Saraclar and R. Sproat. Lattice-based search for spoken utterance retrieval. In *Proc. HLT-NAACL*, pages 129–136, 2004.
- [134] R. Sarikaya, B. Zhou, D. Povey, M. Afify, and Y. Gao. The impact of ASR on speech-to-speech translation performance. In *Proc. ICASSP*, pages 1289–1292, 2007.
- [135] R. E. Schapire and Y. Singer. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168, 2000.
- [136] H. Schwenk and J. Gauvain. Improved ROVER using language model information. In *Proc. ISCA ITRW Workshop on ASR*, pages 47–52, 2000.
- [137] S. Sekine and M. J. Collins. The EVALB software. <http://cs.nyu.edu/cs/projects/proteus/evalb>, 1997.
- [138] W. Shen, R. Zens, N. Bertoldi, and M. Federico. The JHU workshop 2006 IWSLT system. In *Proc. IWSLT*, 2006.
- [139] E. Shriberg et al. Can prosody aid the automatic classification of dialog acts in conversational speech? *Language and Speech*, 41(3–4):439–487, 1998.
- [140] E. Shriberg et al. Prosody-based automatic segmentation of speech into sentences and topics. *Speech Communication*, 32(1-2):127–154, September 2000.
- [141] E. Shriberg et al. Observations on overlap: Findings and implications for automatic processing of multi-party conversation. In *Proc. Eurospeech*, pages 1359–1362, 2001.
- [142] E. Shriberg and A. Stolcke. Prosody modeling for automatic speech understanding: An overview of recent research at SRI. In *Proc. ISCA Tutorial and Research Workshop on Prosody in Speech Recognition and Understanding*, pages 13–16, 2001.
- [143] E. Shriberg, A. Stolcke, and D. Baron. Can prosody aid the automatic processing of multi-party meetings? Evidence from predicting punctuation, disfluencies, and overlapping speech. In *Proc. ISCA Tutorial and Research Workshop on Prosody in Speech Recognition and Understanding*, pages 139–146, 2001.
- [144] S. Siegel and J. Castellan. *Nonparametric Statistics For the Behavioral Sciences*. McGraw-Hill Inc., New York, NY, second edition edition, 1988.
- [145] K. Silverman, M. Beckman, J. Pitrelli, M. Ostendorf, C. Wightman, P. Price, J. Pierrehumbert, and J. Hirsberg. Tobi: A standard for labeling English prosody. In *Proc. ICSLP*, pages 867–870, 1992.

- [146] M. Siu and H. Gish. Evaluation of word confidence for speech recognition systems. *Computer Speech and Language*, 13(4), pages 299–318, 1999.
- [147] M. Snover, B. Dorr, R. Schwartz, L. Micciula, and J. Makhoul. A study of translation edit rate with targeted human evaluation. In *Conf. of the Association for Machine Translation in the Americas*, 2006.
- [148] F. Soong, W.K. Lo, and S. Nakamura. Optimal acoustic and language model weights for minimizing word verification error. In *Proc. ICSLP*, pages 441–444, 2004.
- [149] A. Srivastava and F. Kubala. Sentence boundary detection in Arabic speech. In *Proc. Eurospeech*, pages 949–952, 2003.
- [150] A. Stolcke. Modeling linguistic segment and turn boundaries for n-best rescoring of spontaneous speech. In *Proc. Eurospeech*, volume 5, pages 2779–2782, 1997.
- [151] A. Stolcke. SRILM – An extensible language modeling toolkit. In *Proc. ICSLP*, volume 2, pages 901–904, 2002.
- [152] A. Stolcke, H. Bratt, J. Butzberger, H. Franco, V. Gadde, M. Plauche, C. Richey, E. Shriberg, K. Sonmez, J. Zheng, and F. Weng. The SRI March 2000 Hub-5 conversational speech transcription system. In *NIST Speech Transcription Workshop*, 2000.
- [153] A. Stolcke et al. Recent innovations in speech-to-text transcription at SRI-ICSI-UW. *IEEE Trans. Audio, Speech and Language Processing*, 14(5):1–16, 2006.
- [154] A. Stolcke, Y. Konig, and M. Weintraub. Explicit word error minimization in N-best list rescoring. In *Proc. Eurospeech*, pages 163–166, 1997.
- [155] A. Stolcke and E. Shriberg. Automatic linguistic segmentation of conversational speech. In *Proc. ICSLP*, volume 2, pages 1005–1008, 1996.
- [156] S. Strassel. *Simple Metadata Annotation Specification V5.0*. Linguistic Data Consortium, 2003.
- [157] S. Strassel. *Simple Metadata Annotation Specification V6.2*. Linguistic Data Consortium, 2004.
- [158] S. Stücker, M. Paulik, M. Kolss, C. Fügen, and A. Waibel. Speech translation enhanced ASR for European parliament speeches - on the influence of ASR performance on speech translation. In *Proc. ICASSP*, pages 1293–1296, 2007.
- [159] S. Stücker et al. The ISL TC-STAR spring 2006 ASR evaluation systems. In *Proc. TC-STAR Workshop*, pages 139–144, 2006.

- [160] E. Sumita, S. Yamada, K. Yamamoto, M. Paul, H. Kashioka, K. Ishikawa, and S. Shirai. Solutions to problems inherent in spoken-language translation: The ATR-MATRIX approach. In *Proc. of MT Summit VII*, pages 229–235, 1999.
- [161] X. Sun. Pitch accent prediction using ensemble machine learning. In *Proc. ICSLP*, pages 953–956, 2002.
- [162] Technical Report, Compaq Computer Corporation. *A Boosting Approach for Confidence Scoring*, 2001.
- [163] Scott M. Thede and Mary P. Harper. A second-order hidden Markov model for part-of-speech tagging. In *Proc. of ACL*, pages 175–182, 1999.
- [164] C. Tillmann and T. Zhang. A localized prediction model for statistical machine translation. In *Proc. of ACL*, pages 557–564, 2005.
- [165] A. Venkataraman, R. Gadde, A. Stolcke, D. Vergyri, W. Wang, and J. Zheng. SRI’s 2004 Broadcast News speech to text system. In *Proc. of Fall RT’04 Workshop*, 2004.
- [166] A. Venkataraman, A. Stolcke, W. Wang, D. Vergyri, V. R. R. Gadde, and J. Zheng. An efficient repair procedure for quick transcriptions. In *Proc. ICSLP*, pages 2002–2005, 2004.
- [167] A. Waibel et al. Meeting browser: Tracking and summarizing meetings. In *DARPA Broadcast News Transcription and Understanding Workshop*, pages 281–286, 1998.
- [168] A. Waibel et al. Advances in meeting recognition. In *Proc. HLT-NAACL*, pages 11–13, 2001.
- [169] A. Waibel, A. Jain, A. McNair, H. Saito, A. Hauptmann, and J. Tebelskis. Janus: A speech-to-speech translation system using connectionist and symbolic processing strategies. In *Proc. ICASSP*, pages 793–796, 1991.
- [170] V. Warnke, R. Kompe, H. Niemann, and E. Nöth. Integrated dialog act segmentation and classification using prosodic features and language models. In *Proc. Eurospeech*, volume 1, pages 207–210, 1997.
- [171] M. Weintraub, F. Beaufays, Z. Rivlin, Y. Konig, and A. Stolcke. Neural-network based measures of confidence for word recognition. *Proc. ICASSP*, pages 887–890, 1997.
- [172] F. Wessel, K. Macherey, and R. Schlüter. Using word probabilities as confidence measures. In *Proc. ICASSP*, pages 225–228, 1998.
- [173] F. Wessel, R. Schluter, K. Macherey, and H. Ney. Confidence measures for large vocabulary continuous speech recognition. *IEEE Trans. Speech and Audio Processing*, 9(3):288–298, 2001.

- [174] F. Wessel, R. Schlüter, K. Macherey, and H. Ney. Confidence measures for large vocabulary continuous speech recognition. *IEEE Transactions on Speech and Audio Processing*, 9:288 – 298, 2001.
- [175] F. Wessel, R. Schlüter, and H. Ney. Explicit word error minimization using word hypothesis posterior probabilities. In *Proc. ICASSP*, pages 33–36, 2001.
- [176] C. Wightman and M. Ostendorf. Automatic labeling of prosodic patterns. *IEEE Transactions on Speech and Audio Processing*, 2(4):469–481, 1994.
- [177] J. Xue and Y. Zhao. Improved confusion network algorithm and shortest path search from word lattice. In *Proc. ICASSP*, pages 853 – 856, 2005.
- [178] N. Xue and F. Xia. *The Bracketing Guidelines for the Penn Chinese Treebank*, 2000.
- [179] Y. Al-Onaizan Y. Lee, S. Roukos and K. Papineni. IBM spoken language translation system. In *Proc. IWSLT*, pages 13–18, Barcelona, Spain, June 2006.
- [180] R. Zens and H. Ney. Improvements in phrase-based statistical machine translation. In *Proc. HLT-NAACL*, pages 257–264, Boston, MA, May 2004.
- [181] R. Zens and H. Ney. Discriminative reordering models for statistical machine translation. In *Proc. HLT-NAACL*, pages 55–63, New York City, NY, June 2006.
- [182] L. Zhai, P. Fung, R. Schwartz, M. Carpora, and D. Wu. Using n-best lists for named entity recognition from chinese speech. In *Proc. NAACL*, pages 37–40, 2004.
- [183] R. Zhang and A. Rudnicky. Investigations of issues for using multiple acoustic models to improve continuous speech recognition. In *Proc. ICSLP*, pages 529–533, 2006.
- [184] Z. Zhang, M. Gamon, S. Corston-Oliver, and E. K. Ringger. Intra-sentence punctuation insertion in natural language generation. Technical Report MSR-TR-2002-58, Microsoft Research, 2002.
- [185] Y. Zhao and G. Karypis. Empirical and theoretical comparisons of selected criterion functions for document clustering. *Machine Learning*, 55(3):311–331, 2004.
- [186] Z.-Y. Zhou and H. Meng. A two-level schema for detecting recognition errors. In *Proc. ICSLP*, pages 449–452, 2004.
- [187] M. Zimmerman, D. Hakkani-Tür, and J. Fung. The ICSI+ Multilingual Sentence Segmentation System. In *Proc. ICSLP*, volume 11, pages 14–16, 2006.

VITA

Dustin Lundring Hillard was born in Salt Lake City, Utah. He lived in Pleasanton, California and Issaquah, Washington before coming to the University of Washington. At the University of Washington he earned a Bachelor of Science degree in Electrical Engineering in 2002 and a Master of Science degree in Electrical Engineering in 2004. In 2008 he earned a Doctor of Philosophy in Electrical Engineering.