Neural Models for Integrating Prosody in Spoken Language Understanding

Trang Tran

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

Doctor of Philosophy

University of Washington

2020

Reading Committee:

Mari Ostendorf, Chair

Hannaneh Hajishirzi

Richard Wright

Program Authorized to Offer Degree:
Electrical Engineering

University of Washington

Abstract

Neural Models for Integrating Prosody
in Spoken Language Understanding

Trang Tran

Chair of the Supervisory Committee:
Professor Mari Ostendorf
Electrical and Computer Engineering

Prosody comprises aspects of speech that communicate information beyond written words related to syntax, sentiment, intent, discourse, and comprehension. Decades of research have confirmed the importance of prosody in human speech perception and production, yet spoken language technology has made limited use of prosodic information. This limitation is due to several reasons. Words (written or transcribed) are often treated as discrete units while speech signals are continuous, which makes it challenging to combine these two modalities appropriately in spoken language systems. In addition, as variable as text can often be, text has fewer sources of variation than speech. Different meanings of a written or transcribed sentence can be communicated through punctuation, but a sentence can be spoken in many more ways, where prosody is often essential in conveying information not reflected in the word sequence. Moreover, given the highly variable nature of speech, most successful systems require a lot of data that covers these different aspects, which in turn requires powerful computing technology that was not available until recently.

Given these challenges, and taking advantage of the recent advances in both the speech processing and natural language processing communities, this work aims to develop new mechanisms for integrating prosody in spoken language systems, using spontaneous and expressive speech. This thesis focuses on two language understanding tasks: (a) constituency

parsing (identifying the syntactic structure of a sentence), motivated by the fact that prosodic boundaries align with constituent boundaries, and (b) dialog act recognition (identifying the segmentation and intents of utterances in discourse), motivated by the fact that prosodic boundaries signal dialog act boundaries, and intonational cues help disambiguate intents. Both parsing and dialog act recognition are important components of spoken language systems.

This work makes several contributions. From the modeling perspective, we propose a method for integrating prosody effectively in spoken language understanding systems, which is shown empirically to advance the state of the art in parsing and dialog act recognition tasks. Further, our methods can be extended to other spoken language processing tasks. Through many experiments and analyses, our work contributes to a better understanding and design of language systems. Finally, speech understanding has broad impact on many areas, as it facilitates accessibility and allows for more natural human-computer interactions in education, health care, elder care, and AI-assisted domains in general.

TABLE OF CONTENTS

Page

List of Figures . . . iii

List of Tables . . . v

Glossary . . . viii

Chapter 1: Introduction . . . 1
- 1.1 Prosody in Spoken Language Understanding . . . 2
- 1.2 Thesis Focus and Contributions . . . 3
- 1.3 Thesis Overview . . . 4

Chapter 2: Background . . . 7
- 2.1 Prosody Overview . . . 7
- 2.2 Prosody in Spoken Language Processing . . . 10
- 2.3 Prosody in Spoken Language Understanding: Our Focus . . . 13
- 2.4 Neural Language Representations . . . 19

Chapter 3: Computational Models for Integrating Prosody in Spoken Language Understanding Tasks . . . 22
- 3.1 Neural Networks for Language Processing . . . 22
- 3.2 Modeling Prosody . . . 29

Chapter 4: Constituency Parsing and Prosody . . . 34
- 4.1 Models . . . 34
- 4.2 Research Questions and Datasets . . . 36
- 4.3 Results and Discussion . . . 38
- 4.4 Summary of Findings . . . 46

Chapter 5: Dialog Act Recognition and Prosody 47
5.1 Models 51
5.2 Research Questions and Datasets 52
5.3 Results and Discussion 53
5.4 Summary of Findings 61

Chapter 6: Effects of Imperfect Transcripts 63
6.1 Automatic Speech Recognizer 64
6.2 Constituency Parsing Experiments 65
6.3 Dialog Act Recognition Experiments 72
6.4 Summary of Findings 77

Chapter 7: Conclusion 80
7.1 Summary of Contributions 80
7.2 Future Directions 82

Appendix A: Appendix 107
A.1 Implementation 107
A.2 Data Splits 108
A.3 Pause Duration Statistics 109

LIST OF FIGURES

Figure Number Page

2.1 Example parse tree of a sentence in the Wall Street Journal dataset. Tokens are cased and punctuations are present, which are often good clues to syntax. 15

2.2 Example parse tree of a spoken utterance in the Switchboard dataset. Tokens are lower-cased (as expected in spoken transcripts), no punctuations are present, and disfluent phenomena (EDITED, INTJ nodes) are common. . . . 16

3.1 RNN-based architecture. Left: RNN encoder-decoder model overview; x_i is the sequence of input vectors (features), $i = 1, \dots T_{in}$, and y_t is the sequence of output vectors, $t = 1, \dots T_{out}$; T_{in} and T_{out} do not need to be equal. Right: the RNN encoder have the same form, which consists of RNN cells. For the encoder, $i_{(\cdot)} = x_{(\cdot)}$ and $s_{(\cdot)} = h_{(\cdot)}$; for the decoder, $i_{(\cdot)} = [d, c, m]_{(\cdot)}$ and $s_{(\cdot)} = d_{(\cdot)}$. Optionally, the encoder can be bi-directional, inducing two sets of RNN cells. In LSTMs, m is an additional input to the unit, which is not present in GRUs. 23

3.2 General setup for parsing (left) and DA recognition (right) in the RNN-based models. In both tasks, the input sequence is the sequence of word-level feature vectors. In parsing, the outputs are parse symbols obtained by linearizing parse trees; in DA recognition, the outputs are joint DA tags obtained by labeling each token in a turn with a symbol E_x (x = the utterance's DA) if the token is at the end of the utterance; the token is labeled as I otherwise. . 26

3.3 Transformer-based model with the multihead self-attention encoder, composed of multihead attention (on the input sequence *itself*), layer normalization, and feedforward blocks. 27

3.4 General setup for parsing (left) and DA recognition (right) in the transformer-based models. In both tasks, the input sequence is the sequence of word-level feature vectors. In parsing, the outputs are scores for each tuple of (a, b, l) span representations, from which a parse tree can be reconstructed. In DA recognition, the outputs are joint DA tags obtained by labeling each token in a turn with a symbol E_x (x = the utterance's DA) if the token is at the end of the utterance, and I otherwise. 28

3.5 CNN module for learning acoustic-prosodic features, in particular f0 and energy features. For each word, we convolve a fixed window of M frames ($M = 100$) based on the time alignment of the words with m filters of widths h_i. Here the illustrated CNN filter parameters are $m = 3$ and $h = [3, 4, 5]$. . . 32

4.1 Parser models overview. Left: the RNN-seq model; Right: the Self-attn model; Center: common CNN module for learning acoustic-prosodic features. Both models take word-level features as inputs: $x_1, \cdots, x_{T_1}$, where $x_i = [e_i \; \phi_i \; s_i]$ is composed of word embeddings e_i, pause- and duration-based features ϕ_i, and CNN-based features s_i. 35

4.2 Data preprocessing. Trees are linearized; POS tags (pre-terminals) are normalized as "XX" and merged with input words at the postprocessing step for scoring purposes. 35

4.3 Predicted tree by a parser using only text (left) made a VP attachment error and missed the disfluency (EDITED) node, whereas the parser with prosody (right) avoided, likely thanks to the presence of a pause. 44

5.1 Joint dialog act recognition models used. C_{u-N} denotes the context vector, i.e. encoded history from previous turns. In the RNN-seq models, C_{u-N} is obtained from the mean-pooled hidden states of another RNN that was run on previous N turns. For the transformer-based models, C_{u-N} can be obtained by mean- and max-pooling of word features in the previous N turns, then concatenated with word features of the current turn. 51

5.2 Confusion matrices for the for the most common DA classes, comparing the model trained only on transcript (left) and the one trained with prosody (right). Results are on the dev set, model with no context, labels from DER scoring. 59

A.1 Histogram of inter-word pause durations in our training set. As expected, most of the pauses are less than 1 second. In some very rare cases, pauses of 5+ seconds occur within a sentence. 109

LIST OF TABLES

Table Number Page

2.1 Overview of statistics of the Switchboard corpus in two tasks of interest: constituency parsing (Treebank3) and dialog act recognition (SWDA) 15

2.2 Example of the most frequent dialog acts in the SWDA corpus. 18

4.1 Summary of datasets used in parsing experiments. 37

4.2 Parsing results (F1 scores) on the SWBD dev set, using only text information, comparing different types of embeddings; all parsers were trained on the SWBD train set. Differences between BERT vs. ELMo, and those between BERT/ELMo vs. others are statistically significant with p-val < 0.01. 40

4.3 Parsing results (F1 scores) on the SWBD dev set, using only text information, comparing different types of training data. The differences between SWBD and SWBD+WSJ are not significant. 41

4.4 Parsing results (F1 scores) on the SWBD test set (3823 disfluent + 2078 fluent sentences): using only transcript information vs. adding acoustic-prosodic features. Comparing transcript+prosody and transcript-only models, statistical significance is denoted as: (*) p-val < 0.02; (†) p-val < 0.05. 42

4.5 Test set F1 scores for different sentence lengths. Prosody shows the most benefit in long sentences. 43

4.6 Percentage of error reduction counts from transcript to transcript+prosody models (first 2 columns) and from ELMo to BERT models (last 2 columns). 43

4.7 Parsing results (F1 scores) for mismatched tuning-testing conditions: conversational (C) vs. read (R) vs. read conversational transcripts (RC). Comparing the improvement of text+prosody over text models, statistical significance is denoted as: (*) p-val < 0.02. 45

5.1 An example of a (partial) dialog in SWDA original form. The "+" tag is used when there is speech overlap between speaker sides. 48

5.2 Example of the same partial dialog in Table 5.1, with continuations merged into the same turn. 49

5.3 Example partial dialog in Tables 5.1 and 5.2 after preprocessing. 50

5.4 Example for computing metrics on transcripts. Here DSER = $2/3 = 0.67$ and DER = $3/3 = 1$. For SLER, the edit distance is 1 (error in red), there are 3 reference segments, so SLER = $1/3 = 0.33$. 54

5.5 Computation of micro and macro F1 on the same example in Table 5.4. Per-instance F1 is computed as $F1 = 2 * \text{Match}/(\text{Reference} + \text{Predicted})$. 55

5.6 DA recognition results (error rates and macro F1) on SWDA development set. "Baseline" denotes the best system by Zhao and Kawahara (2019), reimplemented as the original paper used a different data split. "BERT" denotes using BERT embeddings as features (no further fine-tuning); "BERT + top layer" denotes fine-tuning the last layer of the BERT model with the DA recognition task; "BERT + transformer" denotes using BERT as features (no fine-tuning) with another transformer encoder before the decoder; "BERT + transformer + top layer" similarly denotes additionally fine-tuning the last layer of BERT. 56

5.7 DA recognition results (error rates and macro F1) on the development set, comparing with and without using prosody features. For the model with prosody, the feature set used here is the same as those in parsing (also described in detail in Chapter 3): pitch (f0), energy (E), pause embeddings (r_e), raw pause (r), word duration (δ). 57

5.8 DA recognition ablation results (error rates and macro F1) on the model trained with prosody and no context on SWDA dev set. f0 denotes pitch, E denotes energy, r_e denotes pause embeddings, r denotes raw pause features, and δ denotes word duration features. 58

5.9 DA recognition results (error rates and macro F1) on test set. Prosody models are those with the best feature set (raw pause, energy, and pitch). 59

5.10 Example where prosody helped avoid a segmentation error. "sd" is the "statement (non-opinion)" dialog act. 60

5.11 Example where prosody helped avoid a segmentation error. "%" is the "incomplete/abandon" dialog act, and "qy" is the "yes/no question" dialog act. 61

5.12 Example where prosody helped avoid a segmentation error. "^2" is the "collaborative completion" dialog act, and "aa" is the "accept/acknowledge" dialog act. 61

6.1 WER (on 1-best) ASR transcripts for each split and task. 65

6.2 Labeled dependency and labeled bracket F1 scores on the development set: "core set" denotes the set of features: parser output score, ASR hypothesis score, sentence length, and number of EDITED nodes. "depth" denotes parse tree depth and "*P" denotes the counts of various constituents in the predicted parse (NP, VP, PP, INTJ) 68
6.3 F1 scores on the development set across different sentence selection settings. 69
6.4 Test set F1 scores: "gain" denotes the relative improvement of the system over the 1-best hypothesis; "gap" denotes the gain achieved relative to the oracle score. 70
6.5 WER on SWBD test set, computed depending on the way a hypothesis is selected: the baseline is ASR 1-best hypothesis; the oracle is WER 1-best selection. 71
6.6 F1 score and WER on the test set, grouped by sentences with and without human transcription errors (based on MS-State corrections). 72
6.7 Example computations of metrics on ASR transcripts. For LER, the label errors are shown in red ("Predicted tags" row); the edit distance here is 4, so LER $= 4/5 = 0.8$. For DAER, the errors also shown in red illustrate edit distance is again 4, but contributed by different tokens, and also result in DAER$= 0.8$. LWER here is $2/3 = 0.67$, NSER $= (4-3)/3 = 0.33$, SER $= \frac{(0+1+0)+(0+1+0+1)}{2*5} = 0.3$. ASER $= \frac{(1+0+0)+(0+1+0+1)}{2*5} = 0.3$. 74
6.8 Macro and micro DA recognition results (error rates) on dev set, comparing DA recognition on human vs. ASR transcripts. LER and SER are overly sensitive to ASR errors. 75
6.9 DA recognition results (error rates) on dev set, comparing DA recognition on human vs. ASR transcripts using the model trained with and without prosody. 76
6.10 Relative differences in macro and micro DA recognition results on dev set, with and without prosody. 77
6.11 DA recognition results on dev set. All metrics are macro averages. 78
6.12 DA recognition results on dev set. All metrics are micro averages. 78

A.1 Data statistics in parsing experiments. 108
A.2 Data statistics in DA recognition experiments. 109

GLOSSARY

AI: Artificial Intelligence

ASR: Automatic Speech Recognition

BERT: Bidirectional Encoder Representations from Transformers

CNN: Convolutional Neural Network

ELMO: Embeddings from Language Models

FFN: Feedforward Network

GRU: Gated Recurrent Unit

HMM: Hidden Markov Model

LR: Logistic Regression

LSTM: Long Short Term Memory

MFCC: Mel-Frequency Cepstral Coefficients

NN: Neural Network

RNN: Recurrent Neural Network

SEQ2SEQ: Sequence to Sequence

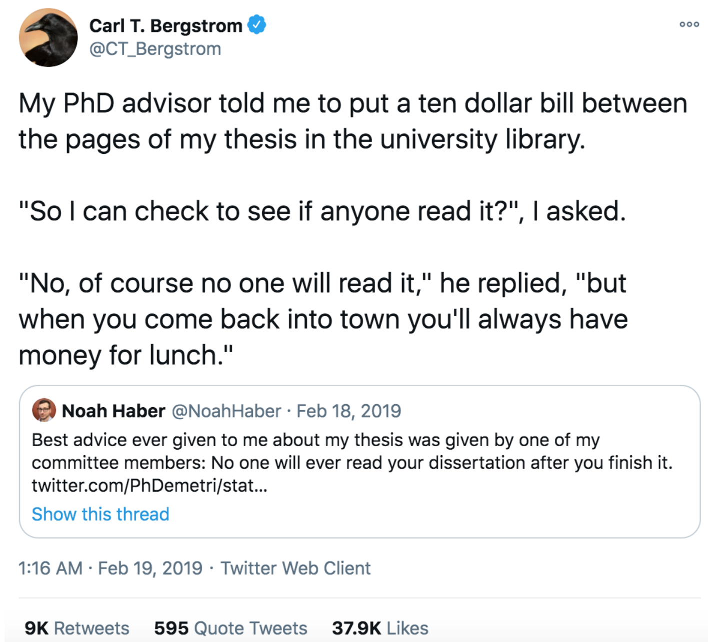
SOTA: State-Of-The-Art

SVC: Support Vector Machine Classifier

SVM: Support Vector Machine

ACKNOWLEDGMENTS

While I was procrastinating on writing this thesis, I came across the following tweet.[1]



Setting aside the fact that everything is now digital (and especially this year, remote!), the number of likes and retweets suggests that Twitterverse agrees with the sentiment — that one's PhD research is largely uninteresting to the majority of *normal* people outside of one's academic bubble. Whether you are reading this because you are actually curious about *prosody in spoken language understanding*, or because I shamelessly advertised my thesis, one important message I hope you would take away is that I did not complete this PhD alone, and never could have.

[1] https://twitter.com/CT_Bergstrom/status/1097787078560034817

First and foremost, I am very fortunate to be advised by Mari Ostendorf. Mari's depth *and* breadth of knowledge, passion for research, uncompromisingly high standards, and commitment to technical rigor and integrity have been nothing short of inspiring; I could only hope to one day reach such level of expertise. Being a great researcher does not necessarily make one a great teacher/mentor, as I have unfortunately realized over the years. Mari, however, is great at both. I have learned a lot just by being a TA for Mari's classes, and improved both my writing and speaking skills thanks to Mari's commitment to educate her students on the importance of scientific communication. I also thank her for being so patient, supportive, and caring — encouraging me to pursue an academic career when I was in doubt, being accommodating to my running (we once scheduled a meeting around my run!), and building a lab that has also become a great resource and my support system.

I could not have asked for a better thesis committee: Richard Wright, Hanna Hajishirzi, Azadeh Yazdan, and Ludo Max. It is always a joy listening to Richard talk about prosody, providing his insights from a linguist's perspective, which has frequently helped pull me out of my self-confined engineering box to see the bigger (and more interesting!) picture of our research problems. I am also thankful for Hanna's insightful questions, comments, and suggestions for research directions, not only as a committee member but also throughout our earlier collaborations when I first started in the lab. I thank Azadeh Yazdan and Ludo Max for providing yet another perspective from their expertise in medical signal processing and speech and hearing sciences. Their questions and comments have been delightfully thought-provoking and they (un)intentionally helped remind me about the value of my own research project, which I sometimes forget as I get lost in the weeds of engineering details.

Pursuing a PhD is tough, and I think it would have been a lot harder without the wonderful TIAL lab colleagues and alumni, whom I am lucky to also call friends: Farah Nadeem, Hao Fang, Hao Cheng, Vicky Zayats, Yi Luan, Kevin Lybarger, Aaron Jaech, Roy Lu, Ellen Wu, Kevin Everson, Sitong Zhou, Sara Ng, Michael Lee, Yuling Gu, Alex Marin, Yuzong Liu, Ji He, Julie Medero, Nicole Nichols. I thank Farah for being my constant cheerleader, and a role model of resilience. How Farah managed to complete her PhD while tirelessly fighting for social justice *and* at the same time raising a son will never cease to amaze me. Coming from a non-CS background, I had trouble even navigating around Linux and had never edited a text file in a terminal. Hao, Hao, and Aaron taught me a lot of very useful software skills and tricks that I now use everyday. More importantly, they helped me without stepping on either end of the spectrum between "RTFM" and "extreme dumbing-down," which is rare, from my experience as a woman in STEM, and therefore much appreciated. I thank all of TIAL for the many technical discussions, talks, as well as generous feedback on my talks and papers; I have learned a lot from every one of you.

I have also been fortunate to collaborate with many people outside of UW. My first internship was at TTIC, where Karen Livescu, Shubham Toshniwal, Kevin Gimpel, Mohit Bansal, and Hao Tang generously hosted me and still remain my great resources. This is also where my first prosody project started, and I am thankful it has become part of this thesis. I thank Yanbin Lu and Jiexian Li for mentoring me during my time as an intern at Amazon, and for remaining as my good friends. I also thank Yang Liu and Jiahong Yuan for hosting my internship at Liulishuo/LAIX, their mentorship, and generous collaboration beyond my time at Liulishuo/LAIX. I am thankful for my recent collaboration with Morgan Tinkler, Gary Yeung, and Abeer Alwan; I have learned a lot about children's speech from them, something I never thought I'd get a chance to explore.

I thank the funding agencies and grants that supported my research: National Science Foundation Grants IIS-1617176 and #1734380, and the Clairmont L. Egtvedt Endowed Engineering Fellowship that got me through this last quarter during the pandemic.

I probably wouldn't have considered a PhD were it not for the inspiration, mentorship, and support from my undergraduate advisors and professors: Robert Nickel, Maurice Aburdene, Richard Kozick, Samuel Craig, Sally Koutsoliotas, and Judy Harris. I first learned

about speech processing from Robert Nickel, and the Bucknell community will always be my first home away from home.

I am thankful for friends in the UW community: Stephanie Swanson, Brenda Larson, Jennifer Huberman, and Elise deGoede Dorough for helping me navigate departmental logistics and policies; Momona Yamagami and Nivii Kalavakonda for being my non-TIAL-but-within-EE friends; Eldridge Alcantara, Ken Yasuhara, and Brek Meuris for the many discussions about pedagogy, which not only gave me a decent head start on developing my teaching philosophies, but also reminded me that I really enjoy teaching; it gave me a sense of purpose I desperately needed when research was lacking progress.

I am thankful that I picked up running as a hobby. It is rather strange to thank a hobby, but running has been truly essential to my mental health. As an unexpected but very much welcomed bonus, I have met some amazing people from the running clubs I joined: RCR, CNW, and ARTC. I thank Chandra Nandi, Zach Tatlock, Ellis Michael, Sami Davies, and Max Willsey (coincidentally, but not surprisingly, Ragnar Van 2!) for being so welcoming when I first joined the group runs. Were it not for Chandra and Zach's incredible openness and friendliness, I would not have felt as comfortable sticking to these runs and would have missed out on becoming friends with this delightful bunch of nerds. I am glad that John Leo and Bing Gao encouraged me to go to CNW's club workouts, for I would have been too intimidated to join by myself. Also, thanks to John for all the in-depth training discussions and book recommendations, a few of which have become my favorites. I am thankful for my workout partners Jim McGill, Jen Rozler, and Laura Goodfellow, our club coaches Ashley Nichols and Spencer Walsh, for consistently showing up and generously dedicating their time. Special thanks to coach Pepper, who is the best at recovery pets. I am grateful for so many ARTC friends I've made: Nina, Nikki, Gina, Charlotte, Allison, Timur, Joe, Nathan, Megan, Ali, Russ, Manny, Peter,... and many others I don't even know the full name of, for not only understanding and sympathizing with my ~~obsession~~ enthusiasm with citystrides, streaking,

etc. but also encouraging and celebrating it. Thank you for celebrating my every PR with me, no matter how small, especially since most non-runners would never understand.

I thank my housemates, old and new: Dan Widener, Kelvin Luu, Kacper Lachowski, Sarah Swansen, Amanda Li, and Stephen MacDonald for being such a mature and considerate group of people, making my living situation not only stress-free but also fun. Thank you for taking care of many house logistics and cutting me a lot of slack on chores when I was busy with this thesis. I am grateful for my awesome landlord, Emily Mattson, who has been incredibly accommodating to me during these uncertain times, and for generously bringing us random treats just because she enjoys cooking!

I am very much not a fan of corny sayings, but this extra cheesy one rings true: "Good friends are like stars. You don't always see them, but you know they're always there." I am lucky to have a lot of such stars: Ha Nguyen, Nadir Sharif, Rishav Chakravarti, Paula Myers, Muyambi Muyambi, Yifan Ge, Tisorn Songsermsawas, Geoff Woo, Michelle Ho, Tony Li, Tejal Raichura, Malyse Uwase, Malika Subramanian, Kaitlin Kerr, Syed Nasr, Aleem Naqvi, Ly Le, Trang Nguyen, Luy Dam, Nguyet Nguyen, Phi Nguyen, Duong Nguyen, Ana Gabriela Aguilera Silva, Jasna Rodulfa Blemberg.

Last but absolutely not least, I thank my Mom, my Dad, my sister Chi, and my extended family "Nha Ba." They would probably agree wholeheartedly with the tweet by Carl Bergstrom above, and wonder if my research is even interesting enough to miss a decade of lunar new years, birthdays, and countless other family celebrations. They have constantly supported me nonetheless.

DEDICATION

To my parents, for supporting me unconditionally.

Chapter 1

INTRODUCTION

"Alexa! Hey Siri! Okay Google!" are now common utterances in the daily speech of many people. The increasing normalization of speech communication with smart devices in everyday life benefited from many advances in spoken language understanding (SLU) research. This increase in voice-based communication, consequently, also motivates further demand for quality improvement in language technology. For example, users now expect to be able to converse naturally with their voice assistants or chatbots, instead of simplifying their speaking patterns to accommodate these devices. Deeper levels of natural or spoken language understanding, beyond recognizing a sequence of words, are therefore becoming more important in artificial intelligence (AI) systems.

Better SLU, however, does not only benefit AI/language systems. There is much to study in human-human communication that has potential for applications in education and health care. For example, analysis of oral reading or narration can provide signals of literacy (Medero and Ostendorf, 2013), comprehension (Lochrin et al., 2015), and language acquisition (Kory, 2014). In the clinical domain, both lexical and acoustic signals can help detect mild cognitive impairment (MCI) (Roark et al., 2011), primary progressive aphasia (PPA) and its subtypes (Fraser et al., 2013), as well as Alzheimer's Disease (AD) (Orimaye et al., 2015; Fraser et al., 2016) and related dementias (Yancheva et al., 2015). All these tasks are facilitated by automatic analysis of speech, and can benefit from effective integration of speech signals that provide valuable information beyond language models.

Despite the growth of voice-based interactions with smart devices, current language technologies have not been able to use speech information fully and effectively. Spoken language processing tasks, such as speech translation and spoken information retrieval, have been largely studied from a text-only perspective. Most resources (datasets) and methods (neural network architectures) for SLU research have been developed from written or transcribed text. State-of-the-art (SOTA) systems only use speech transcripts as inputs, while the acoustic signal carries additional information beyond words: *prosody*.

1.1 Prosody in Spoken Language Understanding

Prosody comprises aspects of speech that communicate information beyond written words related to syntax, sentiment, intent, discourse, and comprehension. On the lowest level, prosody disambiguates many homographs (e.g. REcord vs. reCORD, PERmit vs. perMIT), especially in situations where they cannot be distinguished from context. On a higher level, prosody helps resolve syntactic ambiguities ("Mary knows many languages, you know." vs. "Mary knows many languages (that) you know."). Via stress, intonation, and timing patterns, prosody helps convey speaker's intent (statement vs. question: "You want coffee." vs. "You want coffee?") and content emphasis ("I want TEA," implying "not coffee, or other beverage options"). On yet another level, prosody can signal speaker's sentiment ("The book was interesting." vs. "The book was INTERESTING!"), attitude and level of engagement ("Yeah, sure." vs. "YEAH! SURE!'), and comprehension or proficiency (fluent vs. disfluent speech). *Prosody* therefore helps disambiguate meaning, dialects, intent, sentiment, etc. — aspects of communication not always reflected by even the most faithful transcripts.

Linguistics research has long confirmed the importance of prosody in speech perception and production, but language processing systems still face challenges in integrating prosody effectively. Computational modeling of prosody has been difficult for multiple reasons: (1) by definition, important aspects of prosody are not explicitly communicated by transcribed words, so it is harder to learn from such data; (2) prosody has mostly been studied in controlled and read speech, while most applications involve spontaneous speech; and (3) integrating continuous prosodic signals with discrete words is not straightforward.

Given the challenges in modeling prosody in spoken language systems, the goal of this work is to develop new mechanisms for integrating prosody using spontaneous and expressive speech, and taking advantage of recent advances in neural approaches for combining continuous and discrete information. Specifically, our approach uses convolutional neural networks (CNNs) to automatically learn prosodic features aligned with the (transcribed or recognized) word sequence, yielding word-synchronous prosodic vectors used jointly with contextualized embeddings.

1.2 Thesis Focus and Contributions

The contributions of this thesis are as follows. We present a computational model of prosody that automatically learns acoustic representations useful for language understanding tasks. Our approach uses a convolutional neural network (CNN) to capture energy and pitch contours over words and their context, which are jointly learned with downstream tasks. Leveraging recent advances in contextualized word representations learned from written text, we show that our use of prosody can still benefit SLU tasks over strong word-only baselines, improving the state-of-the-art results.

To assess the proposed approach in modeling prosody, this work focuses on two language understanding tasks: constituency parsing and joint dialog act (DA) segmentation and classification (henceforth referred to as dialog act recognition). On the sentence level, we study how using prosody can benefit **constituency parsing** — the task of identifying the syntactic structure of a sentence. This study is motivated by the fact that prosodic boundaries align with constituent boundaries. On the sentence and discourse level, we develop methods of using prosody in **dialog act recognition** — the task of identifying segments within turns and their corresponding communicative function, i.e. speech/dialog act. This study is motivated by the fact that prosodic boundaries help signal segment boundaries and intonational cues help disambiguate intents.

We show analyses of cases where prosody most benefits parsing and DA recognition, contributing to a better understanding of how speech information can benefit NLP systems. In particular, we show that for constituency parsing, prosody benefits longer and more disfluent sentences, helping disambiguate and avoid attachment errors. In DA recognition, we show that prosody provides most benefit in segmentation, as well as helps reduce the most common types of DA confusions (statement vs. opinions).

We show empirically that spontaneous speech and read speech differ in both the lexical style and prosodic style, where a parser trained on spontaneous speech suffers less performance degradation when evaluated on read speech, unlike vice versa. This result suggests that spontaneous speech in general is more useful for training AI systems, which we hypothesize is in part thanks to its diverse prosody.

We assess the effects of imperfect transcripts on parsing and DA recognition, by studying the performance of our models on automatic speech recognition (ASR) data. Using a simple re-ranking system, we show that prosody still helps parsing, yielding improvements over 1-best parses relative to the oracle N-best gain. In all settings, parsing using prosodic features outperforms parsing with only transcript information. Similarly, in joint DA recognition, we show that prosody still helps improve performance, especially in segmentation, where the gain is significantly larger compared to transcript-only baselines.

Both parsing and dialog act recognition are important components of spoken language systems, and provide better understanding of prosody in human-human communication. The methods in our work can be generalized to other SLU tasks, and have the potential to contribute to more natural human-computer interactions in education, health care, elder care, and numerous other AI-assisted domains.

1.3 Thesis Overview

This dissertation is structured as follows.

In Chapter 2, we provide background on research in prosody, language processing research that uses prosody, as well as current widely successful NLP methods that we build on. Section 2.1 gives an overview of definitions and common conventions for prosody annotations and research in speech perception and production that motivates the use of prosody in language systems. A review of common spoken language understanding studies using prosody is also provided in Section 2.2, including a brief overview of prosody in both speech synthesis and speech understanding. Our tasks of interest, constituency parsing and dialog act recognition, are introduced in Section 2.3, including the standard spontaneous speech dataset, Switchboard, and related parsing and DA recognition research. Additionally, in Section 2.4 we give a brief overview of recent successful approaches to word representations in NLP.

Chapter 3 describes our general approach for integrating prosody in our studies. Section 3.1 reviews the general encoder-decoder neural network approaches that have benefited multiple NLP tasks recently, including recurrent neural network models and transformer models. These architectures provide strong baselines and set up frameworks that can be used for integrating prosodic information in our tasks. We then present our proposed model for incorporating prosody in Section 3.2. This model is developed to use low-level and frame-based speech features, such as pitch and energy, that are learned jointly with a specific task, therefore providing task-specific speech signal representations that are learned automatically, without the need for expensive human annotations but still motivated by previous research on prosody.

Our studies on constituency parsing are presented in Chapter 4. We introduce the models used in Section 4.1 and review research questions in Section 4.2. Our experiment results and discussion are presented in Section 4.3, where we provide analyses on how prosody benefits parsing, and show the importance of using expressive, spontaneous speech in parser training. Section 4.4 summarizes the findings of this chapter.

For DA recognition, our studies on are presented in Chapter 5. The models we used are introduced in Section 5.1 and research questions in Section 5.2. We present experiment results and discussion in Section 5.3, where we show how prosody helps improve DA segmentation and detection of opinions, among other results. A summary of findings is also presented in Section 5.4.

Chapter 6 presents our study on ASR transcripts. Here we assess the performance of our developed systems on imperfect transcripts and determine how useful prosody can still be in this scenario. Experiments on parsing are provided in Section 6.2 and DA recognition in Section 6.3. For both parsing and DA recognition, we again show that prosody is still beneficial, and in the case of DA recognition, even more so compared to perfect transcripts. A summary of findings from this chapter is in Section 6.4.

Finally, a summary of findings and discussion of future directions are provided in Chapter 7. We review our contributions in Section 7.1 and suggest directions for future research in Section 7.2.

Chapter 2

BACKGROUND

This chapter reviews literature related to prosody in human communication research and prosody in the broader spoken language processing area. Motivated by prosody research in processing human-human dialogs, we focus on two tasks (parsing and dialog act recognition), both of which are based on the Switchboard (SWBD) dataset. We also review recent neural language processing approaches that facilitated our work.

2.1 Prosody Overview

In this section, we give an overview of definitions and conventions of prosody from perception studies and a linguistics research perspective.

2.1.1 Definitions and Conventions

Prosody consists of elements in speech beyond orthographic words, i.e. the part of human communication that emphasizes and groups words, disambiguates meaning, and expresses speakers' attitudes and emotions. While definitions of prosody often encompass a variety of speech phenomena, researchers have largely converged to representing prosody on two levels: symbolic and acoustic. These two levels are also related to two common ways of defining prosody in the linguistics community, by its *function* (the symbolic level) and its *form* (the acoustic level) (Wagner and Watson, 2010). From the *function* perspective, prosody refers to properties of speech that depend on and help convey structure and meaning of an utterance, such as marking phrase boundaries and prominence, communicating speakers' attitude and focus. From the *form* perspective, prosody comprises of segmental (syllable-level) and suprasegmental (word- and phrase-level) aspects of speech, which are reflected in acoustic cues such as pauses, word/syllable lengthening, pitch (f0), and energy. These variations in the acoustic signal, individually and in combination, contribute to the realization of a sentence's structure and meaning.

In representing the symbolic structure of prosody, at least for standard American English, researchers have focused on two aspects of speech: (a) **prominence**, which characterizes locations of relative salience in an utterance, and (b) **phrasing**, which creates groupings of words. Both prominence and phrase boundaries are signaled by a combination of energy, f0, duration lengthening, and pausing; each aspect exhibiting different patterns of energy, timing, and f0 changes. One common framework for describing prosody is ToBI (TOnes and Break Indices), motivated by works of Pierrehumbert (1980) and Price et al. (1991). ToBI has been largely adopted as a prosody transcription system for standard American English (Silverman et al., 1992). Briefly, ToBI represents the intonation contour in an utterance by a series of H(igh) and L(ow) tone markings, and phrase boundaries by break indices (0-4) quantifying the degree of disjuncture between words. After ToBI was introduced, there have been efforts to adapt it to other languages: e.g. Korean (Jun, 2000), Japanese (Venditti, 2000), Chinese (Aijun, 2002), and German (Grice et al., 2005), among others.

While ToBI remains the most common prosodic event annotation framework, many others exist. For instance, INTSINT (INternational Transcription System for INTonation) developed by Hirst (1987) was an attempt at becoming the prosodic equivalent of the IPA (International Phonetics Alphabet). Tilt, proposed by Taylor (1998), and SLAM (Stylization and LAbeling of speech Melody), by Obin et al. (2014), are models designed to facilitate automatic analysis and labeling of intonation, i.e. these models described the intonation patterns in a simpler way to be integrated into spoken language systems. For English, RaP (Rhythm and Pitch), proposed by Dilley (2005), is another annotation system developed to address certain aspects in ToBI that were found to be lacking, e.g. the precise correspondence between phonetic attributes to categories of intonational contrast and speech rhythm labeling (Breen et al., 2006). A key difference between RaP and ToBI is RaP's emphasis on transcribers' *perception* of prosodic events, hence the pitch (f0) contour is considered an aid rather than a requirement as in ToBI, for example.

2.1.2 Prosody in Human Communication

Research on the role of prosody in language production and comprehension dates back to 1970s (Wagner and Watson, 2010; Dahan, 2015). Following the definitions in Section 2.1.1, most research has focused on how the acoustic cues — energy, pitch, and timing (word/syllable duration, pausing) — interact and reflect two symbolic aspects: prominence and phrasing. This relationship is commonly revealed and analyzed in the way prosody contributes to resolving ambiguities and therefore communicating the intended meaning.

For **phrasing**, pre-boundary lengthening has been shown to correlate with the strength of the boundary (Wightman et al., 1992). Specifically, the articulatory difference between segments is greater around a prosodic boundary (Fourgeron and Keating, 1997), and boundary effects extend up to 3 syllables from the boundary, decreasing with the distance from the prosodic boundary (Byrd et al., 2006). Further, these observations are supported by ERP (Event-Related Potentials) studies, which show reliable elicitation of a positive shift in electrical activity at the closure of the phrase, i.e. a CPS (Closure Positive Shift) (Bögels et al., 2011; Peter et al., 2014).

Similarly, **prominence** is also signaled by duration, pitch, and energy cues. Beckman and Edwards (1992) suggested that the changes in duration related to prominence are different from those related to phrase boundaries: increased vs. decreased gestural stiffness (one parameter of their speech articulation model). Ladd and Morton (1997) found increased pitch range to encode emphasis, Xu and Xu (2005) found decreased pitch range to signal post-focal material, and Kochanski et al. (2005) suggested that loudness is a better acoustic correlate of focus than pitch.

In relation to sentence structure, syntactic boundaries have been found to be well-aligned with prosodic boundaries (Grosjean et al., 1979). Lehiste (1973) showed that the most reliable acoustic cues for resolving syntactic ambiguities are pre-boundary lengthening and pauses. Fant and Kruckenberg (1996) found strong correlations between pause duration and syntactic boundary level, and Ladd (1988) found that pitch scaling helps disambiguate different coordination structures. Price et al. (1991) also showed that listeners can use prosodic information to resolve syntactic ambiguities, which is further supported by recent work (Watson and Gibson, 2005; Snedeker and Casserly, 2010).

In relation to meaning (besides syntactic disambiguation), prosody signals important aspects of information both on the utterance and discourse levels. For example, prominence signals the relative importance of an entity in discourse (Grosz, 1977), and the location of nuclear stress aids the interpretation of sentences with focus-sensitive operators (e.g. *only, sometimes, all, most,* etc.) (Halliday, 1967a; Wagner et al., 2010). Older linguistic studies suggest that prosody helps distinguish given vs. new information status, with old (given) items being de-accented (Halliday, 1967b; Chafe, 1976). More recent work shows that the acoustic realization still depends on many factors such as the location of the item in the utterance, and whether its surrounding items are accentable due to their own information status (Huang and Hirschberg, 2015). In standard American English, Grosz and Hirschberg (1992) found that phrases with new topics are begun with a wider pitch range and follow longer pauses, while topic-final phrases are characterized by a narrow pitch range and but also precede longer pauses.

To summarize, there is evidence from linguistic studies that prosody plays an important role in speech production and perception. These findings inform us of important acoustic correlates to prosodic structure and therefore provide a guide to our feature selection and model development.

2.2 Prosody in Spoken Language Processing

Motivated by the results of the linguistic studies above, researchers in the engineering community have looked at ways to incorporate prosody into spoken language processing systems, with more effort (and success) in speech synthesis than speech understanding.

In speech synthesis, a generated utterance that is considered to be of high quality is often one that has natural prosody. It is therefore unsurprising that mechanisms for controlling prosody have been well studied in synthesis research. In traditional text-to-speech (TTS) systems, where the input is (user-defined) text, there is often a separate text analysis module, which predicts symbolic prosody elements (phrasing and prominence), informing the audio generation module via timing and pitch parameters. Direct prosody control is then achieved by learning appropriate pitch and timing characteristics, often parametrized by the source-filter speech production model conditioned on the predicted prosodic symbols. For example, Maia et al. (2007) focused on learning the source excitation parameters while others trained Hidden Markov Models (HMMs) to learn filter parameters (Tokuda et al., 2013). Another approach for direct prosody control involves waveform modification, as in concatenative synthesis systems (Obin et al., 2012). In domain-constrained synthesis systems, i.e. concept-to-speech (CTS), prediction of prosody symbols and search of concatenative speech units can be done jointly, by passing an annotated network that represents concepts (Bulyko and Ostendorf, 2002), or by representing speech units with a variety of sentence- and document-level semantic features (Pan, 2002).

Most recently, end-to-end neural approaches for TTS (van den Oord et al., 2016; Wang et al., 2017) demonstrated high-quality synthesized speech in addition to diverse realistic voices. For prosody control, a recent approach proposed learning latent style embeddings (Skerry-Ryan et al., 2018), which capture certain aspects of reference prosody, e.g. voice quality and pitch, without direct modeling of prosody. Subsequently, Wan et al. (2019) proposed a TTS system, CHiVE, that learns to directly predict prosodic features (pitch, energy, duration) using a hierarchical variational auto-encoder. However, these types of end-to-end approaches do not allow for prosody control through markup languages, such as the Speech Synthesis Markup Language (SSML),[1] Speech Integrating Markup Language (SIML) (Pan and McKeown, 1997), and Sable.[2] Although limited in range of control, these markup workarounds allow for some flexibility from the users' perspective.

Spoken language understanding, however, has not seen success from using prosody to

[1]https://www.w3.org/TR/speech-synthesis11

[2]http://www.cstr.ed.ac.uk/projects/festival/

the same extent as spoken language generation. One reason is that the fundamental step in speech understanding involves correctly identifying the word sequence. Thus, automatic speech recognition (ASR) has been the priority of research for a long time, and most applications using prosody either rely on available transcripts or jointly model prosody with the recognition task.

Several lines of work have used prosody in varying degrees of scope and tasks. For word recognition itself, earlier work modeled phoneme and words by conditioning the acoustic and language models on pitch accent and intonational phrase boundaries (Chen et al., 2006), showing reduction in word error rate (WER) by up to 10%. Hasegawa-Johnson et al. (2005) showed that a prosody-dependent speech recognizer, which also learned to predict prosodic events, can lower WER compared to prosody-independent systems. Similarly, Vicsi and Szaszak (2010) trained their speech recognizer jointly with word boundary detection module, and improved word recognition by incorporating prosodic information in N-best lattice rescoring. However, current SOTA ASR systems do not use prosody.

Another line of studies focuses on predicting prosodic events, in particular pitch accent detection/classification and intonational phrase boundary classification. The types of pitch accent and intonational phrase boundaries are most often based on those defined in ToBI. Many researchers (Wightman and Ostendorf, 1994; Levow, 2005; Brenier et al., 2005; Rosenberg and Hirschberg, 2009; Rosenberg, 2010) have proposed systems that learn to predict ToBI labels with traditional machine learning approaches such as decision trees and maximum-entropy classifiers. More recent approaches are neural-based, which typically use convolutional neural networks (CNNs) or recurrent neural networks (RNNs) to model the sequence of words and contexts that can be used as features for prosody prediction (Stehwien and Vu, 2017; Stehwien et al., 2018). However, as these studies rely on supervision signals from prosodic annotation, i.e. ToBI, their scope has been relatively constrained in terms of both data — ToBI annotation is expensive — and diversity in style — mostly done on read news speech, typically using the Boston News Corpus (Ostendorf et al., 1996).

For downstream applications, prosodic features have been shown to benefit a range of tasks, from segmentation-related tasks to meta information and paralinguistics tasks. Segmentation tasks where prosody was shown to improve performance include topic segmentation (Hirschberg and Nakatani, 1998; Tür et al., 2001), sentence boundary detection (Kim and Woodland, 2003; Liu et al., 2004; Kolář et al., 2006), and turn segmentation (Hirschberg et al., 2004). For meta information recognition and paralinguistics tasks, prosody has been helpful in language identification (Martinez et al., 2012; Martinez et al., 2013), emotion recognition (Luengo et al., 2005; Cao et al., 2014), stance classification (Ward et al., 2017, 2018), and deception detection (Levitan et al., 2018; Chen et al., 2020), to name a few.

Most of these studies either rely on available prosodic annotation (or predicted prosodic annotation) on the word level (discrete ToBI prosodic representation), or attempt to model the prosodic patterns for the whole utterance, i.e. without considering the alignment between the acoustic stream and the word sequence. These studies often involve simple averaging of frame-level features, or stacking utterance-level frame statistics of a large set of hand-selected features (Eyben et al., 2010). While there have been some success with such approaches (Stehwien and Vu, 2017; Roesiger et al., 2017), this type of prosody modeling might not capture the word-level acoustic variations, which can provide valuable information in tasks that rely on the word sequence identity.

In summary, computational models of prosody have been more effectively explored in speech synthesis than in speech understanding. Our approach aims to address the limitations in scope of use and modeling. Specifically, we do not require expensive annotations, and develop a framework for integrating word-synchronous acoustic representations without relying heavily on perfect transcripts and timing information.

2.3 Prosody in Spoken Language Understanding: Our Focus

In order to use prosody effectively in language systems, we need to learn from natural conversational speech. This section reviews our basis of focus — the Switchboard corpus of conversational speech with rich annotations, and two tasks that can be studied in detail given this corpus: constituency parsing and dialog act recognition.

2.3.1 The Switchboard Corpus

Switchboard (SWBD), originally collected by Godfrey and Holliman (1993) and later cleaned up by Marcus et al. (1999), is a collection of 2,400 telephone conversations between 543 speakers of American English. The speakers were strangers, and were asked to discuss a predefined topic from a set of prompts. Many follow-up datasets were based on SWBD, each annotating a different aspect of the conversations. About 642 of the conversations were annotated with constituency parse trees as part of the Penn Treebank corpus — Treebank 3 (Marcus et al., 1999), and a bigger set of 1,155 conversations was annotated with dialog acts as part of the SWBD-DAMSL project (Jurafsky et al., 1997), the SWDA corpus. Other layers of annotations have also been released; Calhoun et al. (2010) provides a comprehensive overview.

Because human transcribers are imperfect, the original transcripts contained errors, some of which were corrected in the Treebank3 release, but not all. Mississippi State University researchers ran a clean-up project which hand-corrected 1,126 conversations and produced alignments between the transcripts indicating the type of errors (missed, inserted, or substituted) (Deshmukh et al., 1998). The authors did not re-annotate other aspects of the dataset such as disfluency, parse structure, and dialog acts. However, these MS-State transcriptions provide a more accurate reference; in our experiments involving prosody, they also make a good resource for analyses for comparing performance of our models that might have been affected by transcription errors.

Table 2.1 presents corpus statistics for the two tasks of our interest. Note that the conversations (and transcripts) for the two tasks are not the same, as Jurafsky et al. (1997) annotated an earlier (original) version of SWBD.

2.3.2 Prosody in Constituency Parsing

Constituency parsing is the task of identifying the syntactic structure of a sentence, which is an important component in many language understanding systems. As mentioned in Section

Table 2.1: Overview of statistics of the Switchboard corpus in two tasks of interest: constituency parsing (Treebank3) and dialog act recognition (SWDA)

	Parsing	Dialog Acts
# conversations	642	1,155
# sentence units	108,783	201,191
# turns	-	101,015
# tokens	828,322	1,582,993

2.1.2, the alignment between syntactic boundaries and prosodic boundaries motivates the use of prosodic features in constituency parsers. Two examples of sentences and their parse representations are shown in Figures 2.1 and 2.2, also demonstrating the difference between typical written text and spoken sentences. Specifically, written text is usually cased (clues to noun phrases) and has punctuations (clues to constituent units), while transcripts of spoken utterances lack such structure signals. More importantly, spoken utterances often include phenomena not seen in written sentences, such as disfluencies (the EDITED node) and filled pauses (the INTJ node).

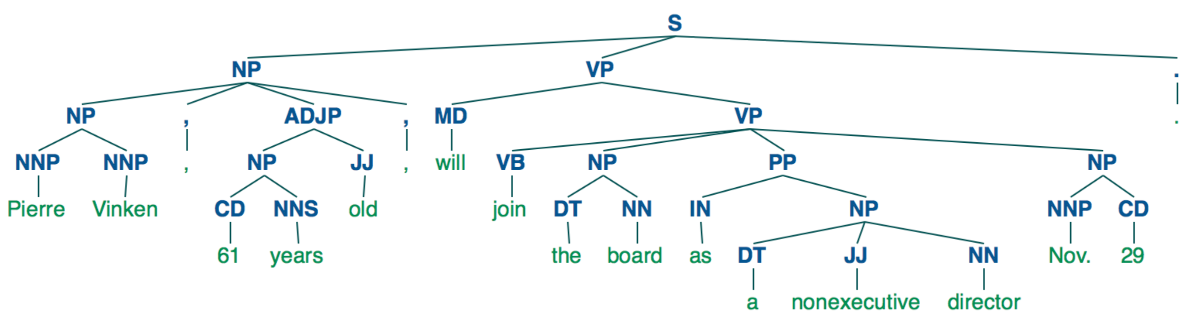


Figure 2.1: Example parse tree of a sentence in the Wall Street Journal dataset. Tokens are cased and punctuations are present, which are often good clues to syntax.

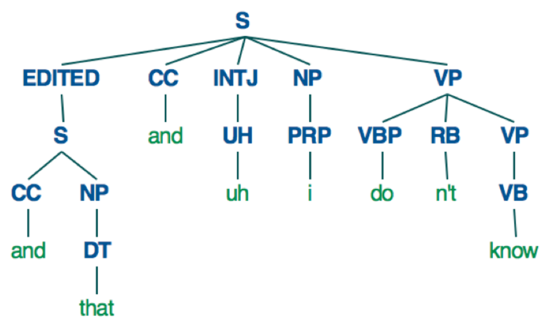


Figure 2.2: Example parse tree of a spoken utterance in the Switchboard dataset. Tokens are lower-cased (as expected in spoken transcripts), no punctuations are present, and disfluent phenomena (EDITED, INTJ nodes) are common.

While parsing is well studied on written text to this day (Gómez-Rodríguez and Vilares, 2018; Kitaev and Klein, 2018; Kitaev et al., 2019), work on parsing conversational speech has been limited. Early work in parsing conversational speech made it clear that speech data poses challenges not present in written text, e.g. the lack of punctuation and the presence of disfluencies (Charniak and Johnson, 2001), and therefore most results seen in parsers trained on text do not transfer well to spoken language data.

Later studies incorporated prosodic features into parsing systems, but initial efforts in directly using raw acoustic features showed discouraging results (Gregory et al., 2004) or modest gains. In particular, Kahn et al. (2005) leveraged automatically predicted prosodic labels (trained on a smaller annotated set) in a statistical parser, achieving improvements in both parsing and disfluency detection. Similarly, Dreyer and Shafran (2007) also predicted prosodic break labels as latent annotations that enriched the parse grammar, leading to an F1 score improvement of 0.2%. In a more recent work, Kahn and Ostendorf (2012) showed that prosody was most useful when sentence boundaries were unknown, in the context of joint parsing and word recognition. These systems, however, assume the availability of human-annotated prosodic features, e.g. ToBI, or features from a system trained on these rich, expert-level annotations.

Another major challenge of parsing conversational speech is the presence of disfluencies, which are grammatical and prosodic interruptions. Disfluencies include repetitions ('I am + I am'), repairs ('I am + we are'), and restarts ('What I + Today is the...'), where the '+' corresponds to an interruption point. Charniak and Johnson (2001) and Johnson and Charniak (2004) suggested that disfluencies are different in character than other constituents, improving parsing performance by combining a PCFG parser with a separate module for disfluency detection. More recently, however, studies have shown that (retrained) SOTA constituency parsers still perform well on disfluent speech (Jamshid et al., 2019), and therefore are good disfluency detectors as a by-product (Jamshid and Johnson, 2020). These studies, however, only parsed transcript texts; prosodic features were not used.

2.3.3 Prosody in Dialog Act Recognition

Dialog act (DA) recognition is the task of identifying the category (speech act) of a spoken sentence unit, such as statement, question, agreement, backchannel, and more. Sentences make up turns, which are associated with a speaker in the conversation; a turn in a dialog consists of one or more sentence-level dialog acts. Some examples of dialog acts are shown in Table 2.2.[3]

Most works in DA recognition treat the task as text classification, focusing on sentence-level classification of a DA given a known (segmented) utterance. Early work (Stolcke et al., 2000) modeled discourse structure as HMM with DAs as emitted observations, where the discourse grammar is modeled via a combination of word n-grams and DA class probabilities produced by a neural network or decision tree classifier learned on prosodic features. The use of prosody was shown to be beneficial in these works, specifically in distinguishing questions from statements, and backchannels from agreements (Shriberg et al., 1998). For example, Jurafsky et al. (1998) found that, compared to agreements, backchannels are often shorter in duration and less intonationally marked (lower f0, energy). In these older studies, prosodic

[3]Taken from `http://compprag.christopherpotts.net/swda.html`.

Table 2.2: Example of the most frequent dialog acts in the SWDA corpus.

Dialog Act	Tag	Example
Statement-non-opinion	sd	Me, I'm in the legal department.
Acknowledge (Backchannel)	b	Uh-huh.
Statement-opinion	sv	I think it's great
Agree/Accept	aa	That's exactly it.
Abandoned or Turn-Exit	%	So, -
Appreciation	ba	I can imagine.

features include pauses, duration, and combinations of frame statistics such as mean/max f0, least-squares all-points regression over utterance and penultimate regions, etc.

More recent neural approaches have focused on modeling utterance-level or dialog-level representations for DA classification, commonly using CNNs (Kalchbrenner and Blunsom, 2013), LSTM-RNNs (Khanpour et al., 2016), or a combination of both (Lee and Dernoncourt, 2016). These studies additionally showed the importance of modeling history and context, as previous utterances are often good signals of the current utterance's speech act, e.g. a statement often follows a question. Along these lines, researchers have incorporated segmental dependencies in modeling DAs via: introducing another DA-level CNN or RNN layer (Ortega and Vu, 2017); using previous reference or predicted dialog act posteriors (Liu et al., 2017); extending both utterance- and dialog-level representations with character-level embedding features (Raheja and Tetreault, 2019) and high-quality pretrained embeddings (Ribeiro et al., 2019); or dynamic models of speakers (Cheng et al., 2019).

These more recent studies do not use prosodic features, with the exception of a few that have only explored basic acoustic features such as Mel-Frequency Cepstral Coefficients (MFCCs) statistics (Ortega and Vu, 2018) or features originally designed to capture paralinguistics elements or speaker characteristics like OpenSMILE (Eyben et al., 2010) in the work by Arsikere et al. (2016). He et al. (2018) achieved virtually the same performance with and without using only MFCCs, which were combined with the text modality using a CNN over all frames in an utterance, i.e. the authors did not enforce the alignment between acoustic frames and the word sequence. Moreover, these studies assume known turn boundaries, which is unrealistic in most dialog systems.

Earlier work that takes into account the problem of segmentation include the pipeline approach by Ang et al. (2005). For segmentation, pause was used as the main prosodic feature, reducing segmentation error rate by at least 10% over their language-model-only approach. For classification, prosodic features used include a small set of simple features such as average pitch, normalized last pitch, and utterance duration. This integration of prosody helped reduce DA classification error rate by around 2% over a lexical-only model. Most related to our work (that also performs segmentation) is the one by Zhao and Kawahara (2019), who studied the task of joint segmentation and classification. Zhao and Kawahara (2019) reported performance on a variety of modeling choices: a cascade pipeline that performs segmentation before classification, a neural sequence-tagging system that predicts joint labels, and a sequence-to-sequence encoder-decoder model with attention that allows for modeling dialog context. The authors found that the encoder-decoder model outperformed the cascade and sequence labeling systems on most metrics by up to 3% in segmentation error rate and 7% in macro F1 score. However, their study did not use prosodic features. A recent work by Dang et al. (2020) used acoustic features (mel-filter bank coefficients) to implicitly perform word recognition as an auxiliary task, but important prosodic features such as pitch and energy were not used. Further, both these works by Dang et al. (2020) and Zhao and Kawahara (2019) did not take advantage of recent advances in neural language representations, which we review next.

2.4 Neural Language Representations

Before the 2010s, successful NLP systems still largely employed bag-of-words (BOW) features or their extensions. The first popular word embeddings, i.e. the continuous, dense vector representations of words, were motivated by the distributional semantics theory and learned via a combination of co-occurrence statistics and dimension reduction (Dumais, 2004), or probabilistic generative latent models (Blei et al., 2003). As neural network language models gained in popularity, the hidden states in the feedforward network (FFN) naturally became the distributed representation of words (Bengio et al., 2003). When recurrent neural network (RNNs) language models overtook FFNs in popularity and trainability, RNNs (Mikolov et al., 2010), and later word2vec (Mikolov et al., 2013) became the standard word representations for NLP tasks. During those years, GloVe (Pennington et al., 2014), which learned representations through co-occurrence statistics, also emerged as a competitive option for representing words.

While useful in many tasks, these word vectors ultimately are static for each word type, i.e. they are unable to distinguish different word senses. Modeling contextual information, therefore, provided a way to ultimately learn such distinctions. Peters et al. (2018) were the first to demonstrate the success of contextualized word embeddings, their ELMo word representations achieved impressive SOTA results on a variety of tasks. Only a year later, Devlin et al. (2019) introduced BERT, which again improved over ELMo on many NLP tasks. The key difference between the neural architecture of ELMo vs. BERT is that the language model in ELMo is learned via Long-Short Term Memory (LSTM) networks, while BERT is learned using a transformer architecture (Vaswani et al., 2017) and two new objectives (masked language model and next sentence prediction). Since then, many variations to contextualized word representations have been proposed, including Transformer XL (Dai et al., 2019), RoBERTa (Liu et al., 2019), SpanBERT (Joshi et al., 2019), XLNet (Yang et al., 2019), GPT variants (Radford et al., 2019), T5 (Raffel et al., 2020), BART (Lewis et al., 2020), etc. and many others are constantly being developed.

While these models have consistently outperformed older word vectors such as word2vec and GloVe, it is worth noting that these large models were all pretrained on written/web-crawled data instead of spoken transcripts. We experiment with using these new word representations in our spoken language systems, and show that, perhaps surprisingly, they are also useful for spoken language data despite the domain mismatch, and serve as strong baselines for systems without prosody.

Chapter 3

COMPUTATIONAL MODELS FOR INTEGRATING PROSODY IN SPOKEN LANGUAGE UNDERSTANDING TASKS

Many NLP tasks can be formulated as encoder-decoder learning, where the encoder is trained to learn useful input representations, and the decoder to predict correct labels specific to a task. In our studies, we focus on two main types of encoders: RNN-based and transformer-based encoders. For decoders, we briefly review common approaches in literature, but they are not a focus of our studies as prosody integration is done on the encoder side. We then describe how we can improve on these by using prosody and give details on our proposed model, which is integrated in both types of encoders.

3.1 Neural Networks for Language Processing

In this section, we review neural architectures that will be modified in our work to incorporate prosodic features. These general frameworks are widely used in NLP and can be adapted to different tasks.

The encoder-decoder model takes as input a sequence of features $x = [x_1, \ldots, x_{T_{in}}]$ and learns to output another sequence $y = [y_1, \ldots, y_{T_{out}}]$. Inputs are usually word representations, and output vectors are often probabilities over output classes. For example, in language modeling, these output probabilities are over the vocabulary size, while in tagging tasks these probabilities are over the tag symbol vocabulary. Outputs for parsing vary depending on the representation of the parse tree structure and will be described below.

3.1.1 RNN-based models

In RNN-based models, both the encoder and decoder are composed of RNN cell units, most commonly the Long-Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) cell, but the Gated Recurrent Unit (GRU) (Cho et al., 2014) is also a popular option. Figure 3.1 shows the general architecture of RNN-based encoder-decoder models.

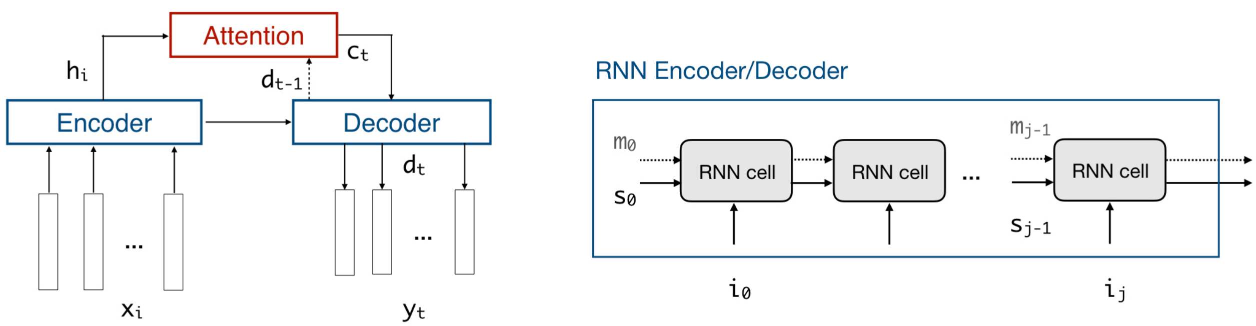


Figure 3.1: RNN-based architecture. Left: RNN encoder-decoder model overview; x_i is the sequence of input vectors (features), $i = 1, \ldots T_{in}$, and y_t is the sequence of output vectors, $t = 1, \ldots T_{out}$; T_{in} and T_{out} do not need to be equal. Right: the RNN encoder have the same form, which consists of RNN cells. For the encoder, $i_{(\cdot)} = x_{(\cdot)}$ and $s_{(\cdot)} = h_{(\cdot)}$; for the decoder, $i_{(\cdot)} = [d, c, m]_{(\cdot)}$ and $s_{(\cdot)} = d_{(\cdot)}$. Optionally, the encoder can be bi-directional, inducing two sets of RNN cells. In LSTMs, m is an additional input to the unit, which is not present in GRUs.

The RNN cells work by encoding the input vectors x into hidden states $h = [h_1, \ldots, h_{T_{in}}]$ where $h_i = \text{RNN}(x_i, h_{i-1})$. In the case of LSTM, the RNN function is described by:

$$f_t = \sigma(W_f[x_t, h_{t-1}] + b_f) \qquad i_t = \sigma(W_i[x_t, h_{t-1}] + b_i) \tag{3.1}$$

$$o_t = \sigma(W_o[x_t, h_{t-1}] + b_o) \qquad \tilde{m}_t = \sigma(W_m[x_t, h_{t-1}] + b_m) \tag{3.2}$$

$$m_t = f_t \odot m_{t-1} + i_t \odot \tilde{m}_t \qquad h_t = o_t \odot \tanh(m_t) \tag{3.3}$$

where the matrices $W_{(\cdot)}$ and bias vectors $b_{(\cdot)}$ are learnable parameters. In the case of GRU, the RNN function is defined by:

$$z_t = \sigma(W_z[h_{t-1}, x_t] + b_z) \qquad \tilde{h}_t = \tanh(W_h[r_t \odot h_{t-1}, x_t] + b_h) \tag{3.4}$$

$$r_t = \sigma(W_r[h_{t-1}, x_t] + b_r) \qquad h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{3.5}$$

where, again, $W_{(\cdot)}$ and bias vectors $b_{(\cdot)}$ are learned parameters of the network. This vanilla RNN encoder-decoder formulation has limitations, as the entire input sequence is represented by one vector $h_{T_{in}}$. Bahdanau et al. (2015) proposed an attention mechanism that enables the decoder to consider the whole input sequence in prediction: the posterior distribution of the output y_t at time step t is given by:

$$P(y_t|h, y_{<t}) = \text{softmax}(W_s[c_t; d_t] + b_s) \tag{3.6}$$

where c_t is referred to as a *context vector* that summarizes the encoder's output h; and d_t is the decoder hidden state at time step t, which captures the previous output sequence context $y_{<t}$.

The attention mechanism computes the context vector c_t as follows:

$$u_{it} = v^\top \tanh(W_1 h_i + W_2 d_t + b_a) \tag{3.7}$$

$$\alpha_t = \text{softmax}(u_t) \tag{3.8}$$

$$c_t = \sum_{i=1}^{T_{in}} \alpha_{ti} h_i \tag{3.9}$$

where vectors v, b_a and matrices W_1, W_2 are learnable parameters; u_t and α_t are the attention score and attention weight vector, respectively, for decoder time step t. This attention mechanism is only *content*-based, i.e. it is only dependent on h_i and d_t. It is not *location*-aware since it does not consider the "location" of the previous attention vector. Chorowski et al. (2015) proposed a convolutional attention scheme that models local phenomena for these context vectors as follows. A feature vector $f_t = F * \alpha_{t-1}$, where $F \in \mathbb{R}^{k \times r}$ represents k learnable convolution filters of width r, and is used in attention calculation. The filters are used for performing 1-D convolution over α_{t-1} to extract k features f_{ti} for each time step i

of the input sequence. The extracted features are then incorporated in the alignment score calculation as:

$$u_{it} = v^\top \tanh(W_1 h_i + W_2 d_t + W_f f_{ti} + b_a) \tag{3.10}$$

where W_f is another learnable parameter matrix.

Finally, the decoder hidden state d_t is computed as

$$d_t = \text{RNN}([\tilde{y}_{t-1}; c_{t-1}], d_{t-1}) \tag{3.11}$$

where $\tilde{y}_{t-1}$ is the embedding vector corresponding to the previous output symbol y_{t-1}, which is ground truth during training, and predicted at inference.

In constituency parsing, the RNN-based decoder learns to output a sequence of linearized parse symbols (more detailed explanation in Chapter 4); in DA recognition, the decoder learns to output a sequence of joint DA tags. Figure 3.2 illustrates this setup. There are also several architecture differences between two tasks: in parsing, the encoder RNN cells are forward-only LSTMs while in DA recognition they are bi-directional GRUs. Another difference in implementation is that the attention mechanism in DA recognition operates on the history vectors instead of the input word sequence (details in Chapter 5).

3.1.2 Transformer-based models

In the original transformer model proposed by Vaswani et al. (2017) for machine translation, both the encoder and decoder are composed from multihead self-attention neural networks. The transformer architecture, however, has shown success as an encoder alone (Kitaev et al., 2019; Devlin et al., 2019). In our studies, we focus on transformers' capability as encoders. Similar to RNN-based encoders, the transformer encoder maps input vectors x_i to a query vector q_i, a key vector k_i, and a value vector v_i. These key, query, and value vectors are then used to compute the probability of word i attending to word j as:

$$p(i \to j) \propto \exp\left(\frac{q_i k_j}{\sqrt{d_k}}\right) \tag{3.12}$$

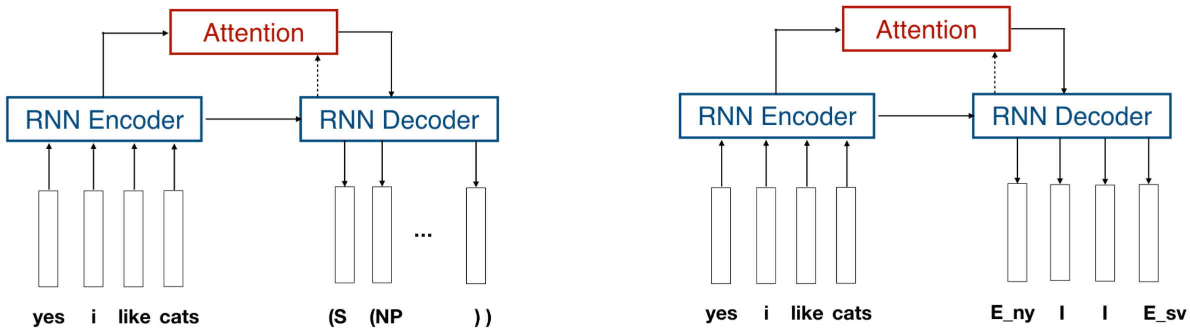


Figure 3.2: General setup for parsing (left) and DA recognition (right) in the RNN-based models. In both tasks, the input sequence is the sequence of word-level feature vectors. In parsing, the outputs are parse symbols obtained by linearizing parse trees; in DA recognition, the outputs are joint DA tags obtained by labeling each token in a turn with a symbol E_x (x = the utterance's DA) if the token is at the end of the utterance; the token is labeled as I otherwise.

for all words in the sequence; d_k denotes the dimension of the key, query, and value vectors. In aggregate, a single attention head for a sequence (sentence or turn) $X = [x_1, x_2, \dots, x_{T_{in}}] \in \mathbb{R}^{d_{model} \times T_{in}}$ is calculated as

$$\text{SingleHead(X)} = \left[\text{softmax}\left(\frac{XW_Q(XW_K)^\top}{\sqrt{d_k}} XW_V\right)\right] W_O \tag{3.13}$$

where W_O is an output projection matrix to map back to dimension d_{model}. All matrices $W_{(\cdot)}$ are learnable parameters.

The original transformer combines outputs of 8 heads over $N = 6$ layers. Specifically, with the first layer's output $Y^1 = \text{MultiHead}(X) = \sum_{n=1}^{8} \text{SingleHead}(X)$, the n^{th} layer output is

$$\begin{aligned} Y^n &= [y_1^n, y_2^n, \cdots y_{T_{in}}^n] && (3.14) \\ &= \text{LN}(\text{FF}(\text{LN}(\text{MultiHead}(Y^{n-1})))) && (3.15) \end{aligned}$$

where $n = 2, \dots, N$; LN denotes the layer normalization operation, and FF denotes the feedforward operation:

$$\mathrm{LN}(x) = a_{\mathrm{LN}} \frac{x - \mu}{\sqrt{\sigma + \epsilon}} + b_{\mathrm{LN}} \tag{3.16}$$

$$\mathrm{FF}(x) = W_{\mathrm{F1}} \mathrm{relu}(W_{\mathrm{F2}} x + b_{\mathrm{F2}}) + b_{\mathrm{F1}} \tag{3.17}$$

μ and σ are the mean and variance of the layer output x, and ϵ is usually set to 10^{-6}. Matrices $W_{(\cdot)}$ and bias vectors $b_{(\cdot)}$ are learnable parameters. Figure 3.3 summarizes the submodules in the transformer-based encoder.

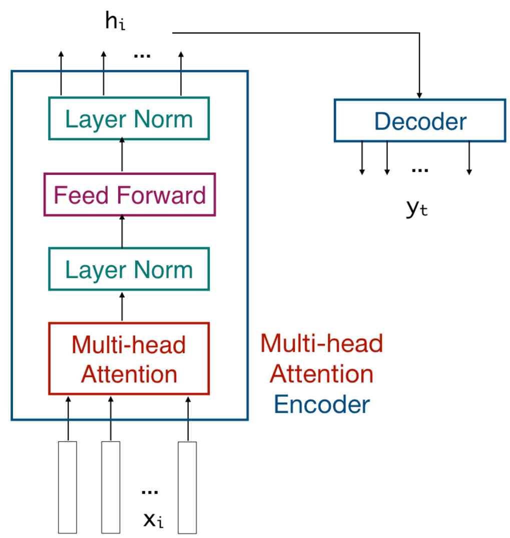
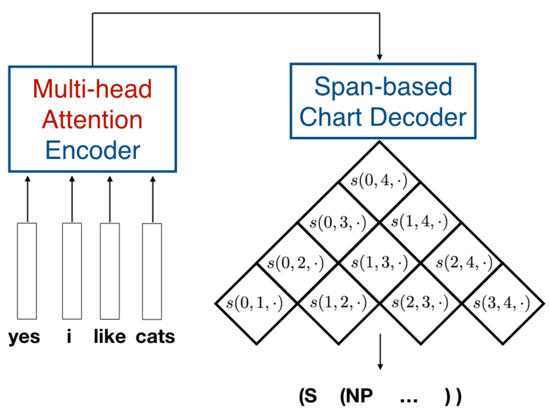


Figure 3.3: Transformer-based model with the multihead self-attention encoder, composed of multihead attention (on the input sequence *itself*), layer normalization, and feedforward blocks.

For decoding, parsing and DA recognition use different types of decoders. In parsing, the decoder is a span-based chart decoder, which follows the one from Stern et al. (2017). The decoder learns to predict a set of best-scoring labeled spans (a, b, l), where $a, b \in [0, T_{in}]$ are position indices, and $l \in V_p$ is a label in the constituent label vocabulary V_p. These span scores are computed as:

$$\mathrm{s}(a,b,\cdot) = M_2\mathrm{relu}(\mathrm{LN}(M_1 v + c_1)) + c_2 \tag{3.18}$$

where $v = [\overrightarrow{y}_b - \overrightarrow{y}_a; \overleftarrow{y}_{b+1} - \overleftarrow{y}_{a+1}]$ summarizes left and right position information of span $(a,b,\cdot)$. Following Kitaev and Klein (2018), $\overleftarrow{y}_t$ and $\overrightarrow{y}_t$ are obtained by splitting in half y_t^n from Y^N above; $M_{(\cdot)}$ and $c_{(\cdot)}$ are learnable parameters.



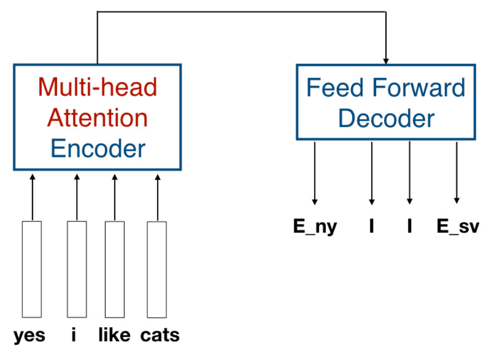


Figure 3.4: General setup for parsing (left) and DA recognition (right) in the transformer-based models. In both tasks, the input sequence is the sequence of word-level feature vectors. In parsing, the outputs are scores for each tuple of (a,b,l) span representations, from which a parse tree can be reconstructed. In DA recognition, the outputs are joint DA tags obtained by labeling each token in a turn with a symbol E_x (x = the utterance's DA) if the token is at the end of the utterance, and I otherwise.

In DA recognition, the decoder is a FF layer that learns to predict probabilities over the DA tag vocabulary $V_{da} = \{\mathsf{I}, \mathsf{E\_sd}, \mathsf{E\_sv}, \ldots\}$ for each word w_t given the final layer encoding y_t^N. Figure 3.4 illustrates the setups for parsing and DA recognition tasks with the transformer-based models.

3.2 Modeling Prosody

In previous work, prosody representation has mainly relied on gold/silver prosodic annotations such as ToBI, or simple averaging/stacking of frame statistics in a word. Symbolic representations are expensive to obtain, and frame statistics do not capture the dynamics of acoustic features in a word. We describe our approach to address these limitations.

3.2.1 Acoustic Features

We explore four types of features widely used in computational models of prosody and motivated by previous linguistics studies: pause, duration, fundamental frequency (f0), and energy (E). Since prosodic cues are at sub- and multi-word time scales, they are integrated with the encoder using different mechanisms.

All features are extracted from transcriptions that are time-aligned at the word level. Time alignments are provided in our SWBD corpus, or can be obtained from forced alignment. In automatically recognized transcripts, time alignments can be a by-product of the systems. In a small number of cases, the time alignment for a particular word boundary is missing. Some cases are due to tokenization. For example, contractions, such as *don't* in the original transcript, are treated as separated words for the parser (*do* and *n't*), and the internal word boundary time is missing. In such cases, these internal times are estimated. In other cases, there are transcription mismatches that lead to missing time alignments, where we cannot estimate times.[1] For the roughly 1% of sentences where time alignments are missing, we simply back off to the parser not learned on prosody. In our later DA recognition experiments, we revised the time alignment estimation to simply copy the start and end times of contractions to each element of the tokenized sequence. This estimation is also done for subword tokens as the BERT model has its own tokenizer.

---

[1] Time alignments are based on a different (corrected) transcript version than used in annotations.

Pause. Given a raw pause duration q, we consider several ways to use it in our system. The pause embedding feature vector $r_{e,i}$ for word i is the concatenation of pre-word pause feature $r_{e,pre,i}$ and post-word pause feature $r_{r,post,i}$, where each subvector is a learned embedding for 6 pause categories: no pause, missing, $0 < q \leq 0.05$ s, 0.05 s $< q \leq 0.2$ s, $0.2 < q \leq 1$ s, and $q > 1$ s (including turn boundaries). The bins are chosen based on the observed pause length distribution (see Appendix A). This way of modeling pause as embeddings was motivated by two main reasons: (1) to handle missing time alignments (in parsing); and (2) duration of pause does not matter beyond a threshold (e.g. $q > 1$ s). However, in later experiments (in DA recognition), we also use raw pause features $r_i = [r_{pre,i}, r_{post,i}]$, which is the concatenation of pre- and post-word normalized pauses, computed as $r_{pre|post,i} = min(1, \ln(1 + q_{pre|post,i}))$, where $q_{pre|post,i}$ is the raw pause duration preceding/following word i.

Word duration. Both word duration and word-final duration lengthening are strong cues to prosodic phrase boundaries (Wightman et al., 1992; Pate and Goldwater, 2013). The word duration feature $\delta_i = [d_{gi}, d_{li}]$ consists of two normalized word durations: global d_{gi} and local d_{li}. The globally normalized word duration d_{gi} is computed as $min\left(5, \frac{\text{wd}_i}{\mu_i}\right)$, where the threshold 5 is used to limit the effect of abnormally long durations possibly due to time alignment errors, and μ_i is the mean duration of the word type; $d_{li} = \frac{\text{wd}_i}{max_u(\text{wd}_i)}$ where wd_i is the raw word duration, and $max_u(\text{wd}_i)$ is the max word duration of all words in that utterance or turn u. The sample mean is used for frequent words (count ≥ 15). For infrequent words we estimate the mean as the sum over the sample means for the phonemes in the word's dictionary pronunciation.

Fundamental frequency (f0) and Energy (E) contours (f0/E). The contour features are extracted from 25-ms frames with 10-ms hops using Kaldi (Povey et al., 2011). Three f0 features are used: warped Normalized Cross Correlation Function (NCCF), log-pitch with Probability of Voicing (POV)-weighted mean subtraction over a 1.5-second window, and the estimated derivative (delta) of the raw log pitch. Three energy features are extracted from the Kaldi 40-mel-frequency filter bank features: E_{total}, the log of total energy normalized by dividing by the speaker side's max total energy; E_{low}, the log of total energy in the lower 20 mel-frequency bands, normalized by total energy, and E_{high}, the log of total energy in the higher 20 mel-frequency bands, normalized by total energy. Multi-band energy features are used as a simple mechanism to capture articulatory strengthening at prosodic constituent onsets (Fourgeron and Keating, 1997). Concatenating f0 and energy features gives a 6-dimensional vector computed at a 10-ms frame rate. To summarize these contour features to a fixed vector for a word, we use a CNN as described in the next section.

3.2.2 Convolutional Neural Network for Acoustic Features

The model described here was introduced in (Tran et al., 2018) and later used in (Tran et al., 2019). Figure 3.5 summarizes the feature learning approach for representing fundamental frequency and energy contours in word-level vectors. Each sequence of f0/E frames corresponding to a time-aligned word (and potentially its surrounding context) is convolved with N filters of m sizes (a total of mN filters). The motivation for the multiple filter sizes is to enable the computation of features that capture information on different time scales. For each filter, we perform a 1-D convolution over the 6-dimensional f0/E features with a stride of 1. Each filter output is max-pooled, resulting in mN-dimensional speech features s_i for word i.

Implementation-wise, for each word i we convolve a fixed window of M frames based on the center time alignment of the words with the CNN filters. In our experiments, $M = 100$ based on the distribution of frame lengths for words in our corpus. Specifically, the average frame length of an word is 25 frames, so a CNN filter of widths 5, 10 are meant to capture sub-word f0/E characteristics, while larger filter widths such as 50, 100 are meant to capture those of the word's surrounding context. Our overall acoustic-prosodic feature vector is the concatenation of pause features $r_{(e),i}$, duration features δ_i, and f0/energy features s_i in various combinations. To simplify notations, we use $\phi_i = [r_{(e),i}, \delta_i]$ to denote the concatenation of pause and duration features for word i.

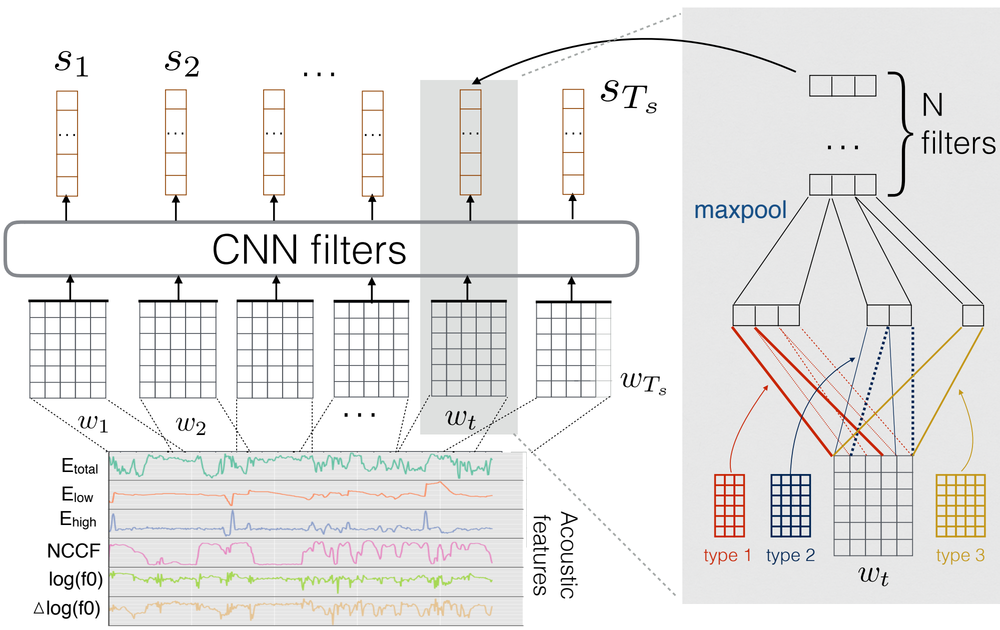


Figure 3.5: CNN module for learning acoustic-prosodic features, in particular f0 and energy features. For each word, we convolve a fixed window of M frames ($M = 100$) based on the time alignment of the words with m filters of widths h_i. Here the illustrated CNN filter parameters are $m = 3$ and $h = [3, 4, 5]$.

In a complete parser/DA recognizer system, each word i has an associated feature vector $x_i = f(e_i, \phi_i, s_i, p_i)$, where the input components e_i, ϕ_i, s_i are word embeddings, pause- and duration-based features, and CNN-learned features, respectively. For the transformer encoder case, to capture the timing information without recurrent connections, the transformer encoder input also includes positional embeddings p_i. The function $f(\cdot)$ that combines these different types of inputs can be simple addition or explicit factorization as detailed in Kitaev and Klein (2018). In our case, we extend the lexical-positional factorization in Kitaev and Klein (2018) to lexical-positional-prosodic factorization. In particular, we learn separate key, query, and value mappings for each component of the input: e_i, p_i, and $[\phi_i, s_i]$.

Chapter 4

CONSTITUENCY PARSING AND PROSODY

To assess the usefulness of our proposed approach, we first study the use of prosody in constituency parsing — the task of identifying the syntactic structure of a sentence. In recent encoder-decoder neural parsers, the encoder learns the input sentence representation and the decoder learns to predict a parse tree. While the input is commonly represented via a sequence of word-level features, representation for the output trees varies: as a sequence of parse symbols (Vinyals et al., 2015), set of spans (Stern et al., 2017; Gaddy et al., 2018), syntactic distances (Shen et al., 2018), or per-word structure-rich labels (Gómez-Rodríguez and Vilares, 2018). A key characteristic in many of these neural parsers is the recurrent network structure, particularly Long Short-Term Memory networks (LSTMs), but Kitaev and Klein (2018) have shown that a non-recurrent encoder such as the Transformer network introduced in Vaswani et al. (2017) is also capable of encoding timing information through self-attention mechanisms, achieving state-of-the-art parse results on the Treebank WSJ dataset.

4.1 Models

We focus on two neural constituency parsing models: **RNN-seq** and **Self-attn**, which we modify to integrate our prosody learning module. Both models accept a sequence of T_{in} word-level features as inputs: $x_1, \cdots, x_{T_{in}}$, where $x_i = [e_i \;\phi_i \;s_i, p_i]$ is composed of word embeddings e_i, position encodings p_i (depending on the model), pause and duration features $\phi_i = [r_{e,i}, \delta_i]$, and a learned representation of f0/E contours s_i — as described in Chapter 3. Figure 4.1 gives an overview of two architectures, with the common acoustic-prosodic feature learning module.

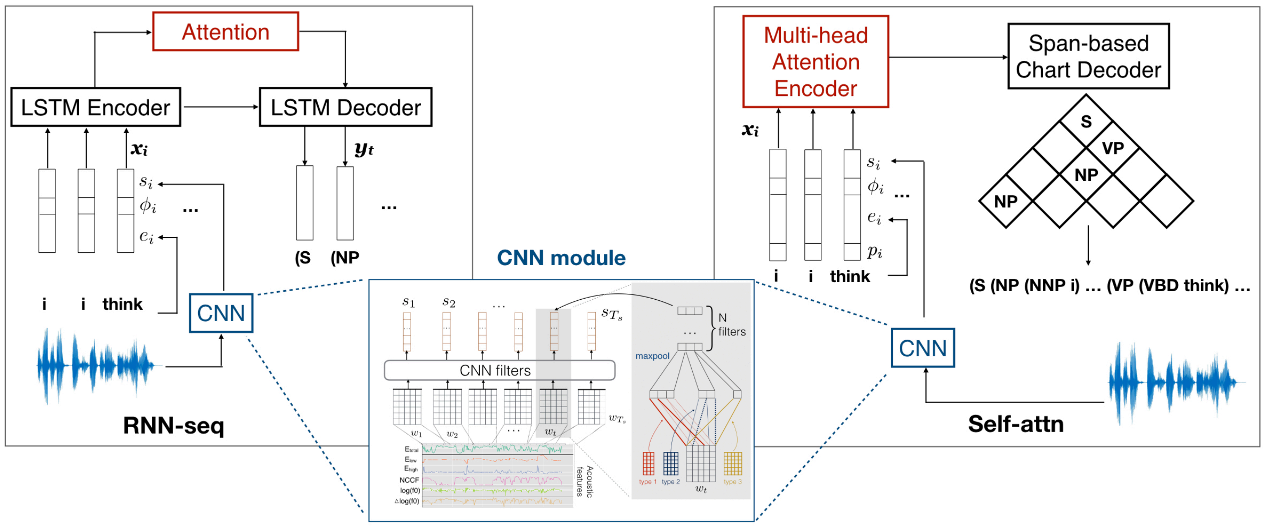


Figure 4.1: Parser models overview. Left: the RNN-seq model; Right: the Self-attn model; Center: common CNN module for learning acoustic-prosodic features. Both models take word-level features as inputs: $x_1, \cdots, x_{T_1}$, where $x_i = [e_i\ \phi_i\ s_i]$ is composed of word embeddings e_i, pause- and duration-based features ϕ_i, and CNN-based features s_i.

Original parse tree

```
                 ┌ INTJ ── UH ── uh
S ── FRAG ───────┤
                 │         ┌ IN ── about
                 └ PP ─────┤
                           └ NP ── PRP ──yourself
```

Linearized parse tree

`(S (FRAG (INTJ (UH uh)) (PP (IN about)`
`(NP (PRP yourself) ))))`

Final POS-normalized linearized parse tree

`(S (FRAG (INTJ XX) (PP XX (NP XX))))`

Figure 4.2: Data preprocessing. Trees are linearized; POS tags (pre-terminals) are normalized as “XX” and merged with input words at the postprocessing step for scoring purposes.

RNN-seq Our baseline RNN-seq model follows the setup of Vinyals et al. (2015). Figure 4.2 illustrates the data preprocessing step in this setup.[1] Specifically, RNN-seq learns a mapping from a sequence of T_{in} word-level features x_i to a linearized sequence of T_{out} parse symbols $z_1, z_2, \cdots z_t, \cdots z_{T_{out}}$, using LSTMs for both the encoder and decoder. In addition, we employ the location-aware attention mechanism proposed in Chorowski et al. (2015), reviewed in Section 3.1, and extend the encoder with the prosodic feature learning module described in Section 3.2.

Self-attn The Self-attn model extends the self-attentive encoder chart decoder of Kitaev and Klein (2018) with the acoustic-prosodic feature learning module as described in Section 3.2. The self-attentive encoder follows the multihead self-attention architecture of Vaswani et al. (2017) and the span-based chart decoder follows the decoder from Gaddy et al. (2018), as reviewed in Section 3.1. The span-based chart decoder in essence works the same way as CKY chart decoding, where, instead of PCFG production probabilities, the scores are span scores $s(a, b, l)$. Because of this setup, the parse trees reconstructed from the chart are guaranteed to be valid.

4.2 Research Questions and Datasets

The goal of this study is to answer the following questions:

1. In assessing the use of neural parsers designed for written text, which architecture also works for speech? We compare Self-attn vs. RNN-seq, and contextualized embedding vs. non-contextualized embeddings.

2. Does prosody improve further on top of the rich text information in neural parsers for spontaneous speech? If so, where does prosody benefit most?

[1]On the decoder end, we also use a post-processing step that merges the original sentence with the decoder output to obtain the standard constituent tree representation. During inference, in rare cases (and virtually none as our models converge), the decoder does not generate a valid parse sequence, due to the mismatch in brackets and/or the mismatch in the number of pre-terminals and terminals, i.e., num(XX) $\neq$ num(tokens). In such cases, we simply add/remove brackets from either end of the parse, or add/remove pre-terminal symbols XX in the middle of the parse to match the number of input tokens.

3. How is the use of prosody affected by mismatch between read and spontaneous speech styles?

To answer these questions, we use several datasets described below, mainly evaluating on the treebanked subset of Switchboard conversational speech data (Section 2.3.1), but including some results on the read version of the treebanked data. Table 4.1 summarizes the different datasets we used: some sets have both audio and parse trees available, while others have only either audio or parse trees.

Table 4.1: Summary of datasets used in parsing experiments.

Data	Style	Available material	Used for	# sentences
WSJ	news text	(gold) parses	train, dev	40k
SWBD	conv. speech	audio, (gold) parses	train, dev, test	96k
CSR	read news text	audio, (silver) parses	train (fine-tune), dev	8k
GT-N	read article text	audio, (gold) parses	test	6k (3k unique)
GT-SW	read SWBD	audio, (gold) parses	test/analysis	31 (13 unique)

We use two primary corpora for training and development: the Wall Street Journal (**WSJ**) corpus of treebanked news articles (Marcus et al., 1999) and the Switchboard (**SWBD**) corpus of telephone speech conversations (Godfrey and Holliman, 1993; Marcus et al., 1999), which are the two standard corpora for constituency parsing studies on written text and conversational speech, respectively. SWBD includes audio files with time-aligned transcripts.

Wall Street Journal (**WSJ**) (Marcus et al., 1999) is a standard corpus of news articles with parse trees used for constituency parsing studies. We use this corpus for assessing the utility of written text parses in training a parser for spontaneous speech transcripts (Question 1). Switchboard (**SWBD**) (Godfrey and Holliman, 1993) is a corpus of conversational speech, which has audio, time-aligned transcripts, and constituency trees (Marcus et al., 1999). We use this set for most of our experiments, assessing both the utility of various aspects of information available to a parser (transcript vs. transcript and prosodic features).

In order to train a parser with prosodic features matched to the read speech style, we use the common read subset of the CSR-I corpus (**CSR**) (Garofolo et al., 1993), which includes read Wall Street Journal sentences (but does not overlap with **WSJ** sentences). CSR is used to fine-tune a pretrained SWBD parser (instead of training from scratch), since the corpus is much smaller than SWBD. The Penn Phonetics Lab Forced Aligner (P2FA) (Yuan and Liberman, 2008) was used to get time alignments. Since the CSR sentences are not covered in the WSJ set, we used a pretrained SOTA parser for written text (Kitaev and Klein, 2018) to obtain silver trees. To verify the quality of the automatically parsed trees, we recruited two linguists to hand-correct a random subset of 100 trees. The annotator agreement is high: the F1 score between annotators' trees is 97.2%. Among the 100 trees, both annotators confirmed that the parser got the perfect tree in 72 sentences, and the rest have minor errors.

To assess parser performance in style mismatch (Question 3), we use two subsets of the GlobalTIMIT dataset (Chanchaochai et al., 2018): **GT-N** and **GT-SW**. **GT-N** contains 3207 news sentences read by 50 speakers, some were read by multiple speakers, totaling 6k read sentences; **GT-SW** contains the read version of 13 Switchboard sentences, read by 29 speakers, totaling 31 read sentences.[2] These sentences were selected from the Treebank3 corpus (Marcus et al., 1999), so they have gold parse trees; we use this set for evaluation and analysis only.

4.3 Results and Discussion

For the RNN-seq parser, we re-implemented the model in Vinyals et al. (2015); for the Self-attn parser, we modify the implementation in Kitaev and Klein (2018) to include the acoustic-prosodic feature learning module and the corresponding factorization.

Because random seeds can lead to different results as demonstrated in Reimers and

[2]The number of read conversational sentences is limited, because we chose to use a standard corpus.

Gurevych (2017), we train and tune each model configuration initialized with 5 random seeds, and report the median prediction as our final result. For both RNN-seq and Self-attn, we used the same optimizer, Adam (Kingma and Ba, 2014), with the same learning schedule as the provided implementations. All models are evaluated using EVALB,[3] i.e. we report standard parseval F1 scores, which is F1 on correctly predicted tuples (a, b, l). Statistical significance was assessed using the paired bootstrap test as described in Berg-Kirkpatrick et al. (2012).

4.3.1 Assessing Transcript-only Parser Models

To assess the impact of different types of text representations in parsing speech transcripts, we train and evaluate our parser on SWBD data, comparing several methods of using/learning word embeddings e_i. These embeddings can be learned jointly with the parsing task, or extracted from pretrained models and then used as features. For pretrained embeddings, we consider the following representations: GloVe (Pennington et al., 2014) embeddings are learned from co-occurrence statistics and have little context information. The standard version (GloVe-Gigaword) was pretrained on a large corpus of 6B tokens (Wikipedia & Gigaword 5). We additionally trained GloVe embeddings on a dataset with style more similar to spontaneous speech, the Fisher corpus (Cieri et al., 2004) and consider the effect of these embeddings on parsing (GloVe-Fisher). Contextualized embeddings such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019) are recent neural models that have been pretrained on a large amount of written text data, capturing larger context information with language modeling auxiliary tasks via bi-LSTM (ELMo) or transformer networks (BERT). Both ELMo and BERT have been reported to benefit a variety of NLP tasks.

Table 4.2 compares performance of different models in combination with different embeddings on the SWBD dev set: the transformer-based model outperforms RNN-seq by a large margin, even without pretrained embeddings. Using pretrained embeddings outper-

[3] `https://nlp.cs.nyu.edu/evalb/`

forms embeddings learned jointly with parsing, even though most pretrained models were on on written text. Further, there is negligible difference between GloVe-Gigaword and the better matched GloVe-Fisher. This suggests that text features pretrained on large written text data do benefit parsing on speech transcripts, with comparable results to text features pretrained on a dataset more similar in style to SWBD like GloVe-Fisher.

Table 4.2: Parsing results (F1 scores) on the SWBD dev set, using only text information, comparing different types of embeddings; all parsers were trained on the SWBD train set. Differences between BERT vs. ELMo, and those between BERT/ELMo vs. others are statistically significant with p-val < 0.01.

Model	Embedding	F1
RNN-seq	Learned	0.880
	GloVe - Gigaword (Pennington et al., 2014)	0.886
Self-attn	Learned	0.910
	GloVe - Gigaword (Pennington et al., 2014)	0.912
	GloVe - Fisher	0.910
	ELMo (Peters et al., 2018)	0.927
	BERT (Devlin et al., 2019)	0.932

Both contextualized models outperform GloVe models by a large margin (p-val < 0.01), with BERT showing the best F1 scores, outperforming ELMo with statistical significance (p-val < 0.01). This is consistent with results in other NLP tasks, confirming that contextualized embeddings are a powerful tool in a range of applications. All embeddings here are used as features, without further fine-tuning the embedding weights. We also ran several experiments where the embedding weights were jointly trained, but the results were worse, probably due to the large number of weights and the limited amount of speech transcripts.

Similar to comparing different types of embeddings, we also assess the effect of using different datasets on parsing speech transcripts. Table 4.3 presents these results. Unsurprisingly, simply training on written text data performs poorly on speech transcripts. Training on additional text-only data (SWBD+WSJ) provides marginal improvement in parsing conversational speech, suggesting that substantial benefit can be obtained with pretrained embeddings, but the dataset for the main task still requires a style match.

Table 4.3: Parsing results (F1 scores) on the SWBD dev set, using only text information, comparing different types of training data. The differences between SWBD and SWBD+WSJ are not significant.

Trained on	ELMo	BERT
WSJ	0.760	0.775
SWBD	0.927	0.932
SWBD + WSJ	0.927	0.934

4.3.2 The Role of Prosody

For this question, we only consider the two best-performing models on transcript-only data: Self-attn with ELMo vs. BERT. Table 4.4 presents the results on SWBD test set, separating results by fluent vs. disfluent (sentences with EDITED and/or INTJ nodes) subsets of sentences.

Comparing transcript-only and transcript+prosody models, prosody helps in both ELMo and BERT. ELMo results are consistent with results on the RNN-seq models: most gains seem to be from disfluent sentences. For BERT, the gains are statistically significant in fluent sentences, but not in disfluent ones. Comparing BERT and ELMo models, BERT-transcript improves over ELMo-transcript with p-val < 0.05 in disfluent sentences and overall, but

Table 4.4: Parsing results (F1 scores) on the SWBD test set (3823 disfluent + 2078 fluent sentences): using only transcript information vs. adding acoustic-prosodic features. Comparing transcript+prosody and transcript-only models, statistical significance is denoted as: (*) p-val < 0.02; (†) p-val < 0.05.

Model	Embedding	all	disfluent	fluent
transcript only	ELMo	0.925	0.915	0.946
	BERT	0.929	0.919	0.949
transcript+prosody	ELMo	0.927*	0.917*	0.949†
	BERT	0.930*	0.921	0.952*

not in fluent sentences. This is likely why BERT-prosody does not improve over BERT-transcript with statistical significance in disfluent sentences, since BERT-transcript itself is already good. BERT-prosody improves over ELMo-prosody in all cases with p-val < 0.05. Additionaly, Table 4.5 shows the parse scores for subsets of sentences grouped by length. For both ELMo and BERT, prosody benefits parsing more for longer sentences than short ones.

We also analyze parse error types each parser makes or improves on. We use the Berkeley Parse Analyzer (Kummerfeld et al., 2012) to categorize the common error types in constituency parsing. Table 4.6 shows the relative error reduction when using prosody vs. using only transcripts, and similarly when using BERT vs. ELMo. For both ELMo and BERT, VP attachment errors are most reduced when using prosody. Figure 4.3 shows an example sentence where prosodic features (pause) help avoid the attachment error made by the parser using only transcript features.

Cases where prosody seems to hurt BERT (Coordination, Clause Attachment, and possibly Modifier Attachment) are contexts where the transcript-only BERT and ELMo models

Table 4.5: Test set F1 scores for different sentence lengths. Prosody shows the most benefit in long sentences.

Embedding	Model	Sentence lengths (# sents) [0, 5] (2885)	[6, 10] (1339)	[11, -] (1677)
ELMo	transcript	0.966	0.963	0.905
	transcript+prosody	0.967	0.964	0.908
BERT	transcript	0.965	0.965	0.911
	transcript+prosody	0.966	0.967	0.913

Table 4.6: Percentage of error reduction counts from transcript to transcript+prosody models (first 2 columns) and from ELMo to BERT models (last 2 columns).

Error Type	Δ(+prosody, transcript)		Δ(BERT, ELMo)	
	ELMo	BERT	transcript	+prosody
Co-ordination	-1.0	-5.1	18.2	14.9
PP Attachment	1.2	1.0	1.2	1.0
NP Attachment	-7.5	0.0	6.0	12.5
VP Attachment	19.2	19.6	-7.7	-7.1
Clause Attachment	8.3	-8.1	11.4	-4.4
Modifier Attachment	7.9	-1.4	11.8	3.0
NP Internal	2.7	7.0	6.5	10.6
Single-Word Phrase	5.2	2.3	-3.5	-6.6
Different Label	1.0	7.3	-2.4	4.1

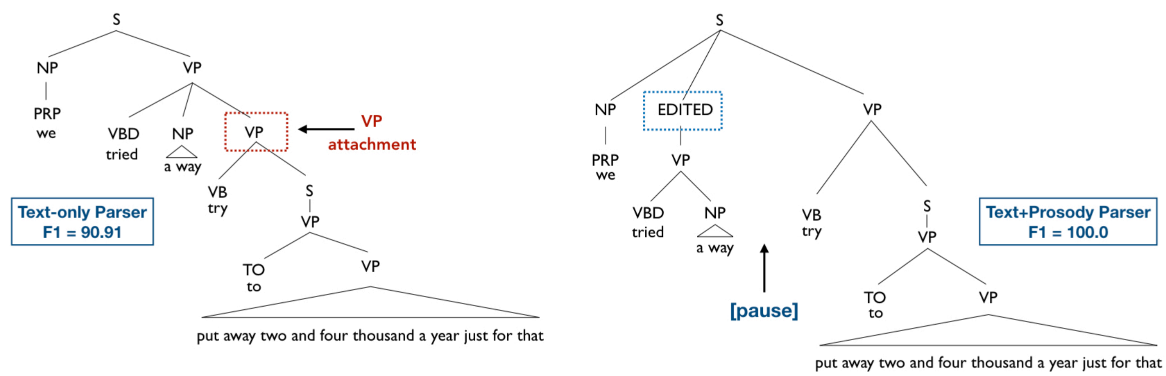


Figure 4.3: Predicted tree by a parser using only text (left) made a VP attachment error and missed the disfluency (EDITED) node, whereas the parser with prosody (right) avoided, likely thanks to the presence of a pause.

have the greatest difference. For the main case where prosody hurts ELMo (NP Attachment), there is no benefit to BERT. These may simply be contexts where there is little need for prosody given well-trained transcript-only models. For Clause Attachment errors, ELMo-speech seems to improve over ELMo-text significantly while the opposite is true for BERT. This is likely because BERT-transcript (3rd column) already significantly outperforms ELMo-transcript, so it is harder for BERT-prosody to improve further over BERT-transcript. This trend also shows up in other types of errors, such as NP attachment and Modifier attachment.

4.3.3 Spontaneous vs. Read Speech

For this experiment, we only consider the models with BERT. Table 4.7 presents parsing results in mismatched tuning-testing conditions. In all settings, training on conversational speech degrades results on read speech minimally, but training on read speech degrades results on conversational speech significantly. Further, prosody consistently helps when the parser is trained on conversational speech, both when testing with matched and mismatched styles. This suggests that conversational speech data is more useful for general purpose parser training, likely because of the diversity in prosodic characteristics available in spontaneous speech, on top of tail-end phenomena (disfluencies) likely captured by the contextualized embeddings.

When testing on conversational speech (SWBD column), the biggest effect of mismatch is associated with the word sequence; the degradation from prosody mismatch seems to have a smaller but still significant impact (p-val < 0.05). However, when testing on read news (GT-N column), the BERT model with prosody tuned on read speech sees a performance gain (p-val < 0.01). These results are consistent with the hypothesis that use of prosody differs in read vs. conversational speech, i.e. the style mismatch is both in terms of words and acoustic cues.

Table 4.7: Parsing results (F1 scores) for mismatched tuning-testing conditions: conversational (C) vs. read (R) vs. read conversational transcripts (RC). Comparing the improvement of text+prosody over text models, statistical significance is denoted as: (*) p-val < 0.02.

		Test data		
Train/tuning data	Model	SWBD (C)	GT-N (R)	GT-SW (RC)
SWBD (C)	transcript	0.929	0.924	0.980
CSR (R)	transcript	0.806	0.939	0.914
SWBD (C)	transcript+prosody	0.930*	0.926*	0.980
CSR (R)	transcript+prosody	0.804	0.942*	0.903

To further explore this question, we ran experiments on the GT-SW sentences. The results in Table 4.7 (GT-SW column) are anecdotal but consistent with the other results. On these sentences, with text-only models, further tuning on read style data degrades performance significantly. For the parsers using prosody, the version trained on spontaneous speech seems to be able to handle the read version of Switchboard sentences, but the one fine-tuned on read text further degrades. It may be that the prosody associated with reading conversation transcripts is not like that associated with reading more formal written text.

4.4 Summary of Findings

In this chapter, we explored the the task of constituency parsing on spoken language, studying the effects of prosodic features and variations in speaking style (read vs. spontaneous). Following a series of empirical experiments, we first showed that contextualized word representations, despite being pretrained on written text, are still useful in parsing speech transcripts. Regarding the use of prosody, we showed that our approach to integrating acoustic-prosodic features further benefits parsing, improving over the strong transcript-only baselines. Our analyses revealed that prosody is especially helpful in longer sentences, reducing attachment errors, and detecting disfluent nodes. Finally, our experiments regarding mismatch in speaking styles showed a minimal degradation when parsers were trained on spontaneous speech and evaluated on read speech, but a more significant degradation vice versa. This finding suggests that conversational speech is generally more useful than read speech, which we hypothesize is in part due to the more diverse prosody, further supporting the importance of using spontaneous speech in developing language systems.

Chapter 5

DIALOG ACT RECOGNITION AND PROSODY

Dialog act (DA) recognition is the task of identifying the dialog act category of a speech segment, such as statement, question, agreement, backchannel, and more. Most recent work, e.g. Ribeiro et al. (2019), achieved high accuracies in DA classification, while assuming known segment boundaries. However, such an assumption is unrealistic, especially in practical spoken language systems. In this chapter, we explore models that perform joint segmentation and DA classification, which we refer to as *DA recognition*, for short.

Conversations involve multiple people talking. Typically, one person has the floor at a time, but there can be speech overlaps associated with interruptions and backchannels (verbal encouragement for the other party to keep speaking). A conversation consists of *turns*, which are speech units spoken by a speaker in the dialog. We define a turn as a segment of speech from a single speaker, bounded by long pauses and/or floor change. Within each turn, there could be one or more *dialog acts*. An example from the Switchboard Dialog Act corpus (SWDA), annotated by Jurafsky et al. (1997), is shown in Table 5.1. In this example, "turns" are defined based on the original transcription guidelines aimed at preserving the timing of speaker interactions, but this often splits up dialog acts. These split DAs are indicated with the "+" tag (for "continuation"), but in itself it is not a meaningful DA category and there will be no prosodic or syntactic cues to the boundary. Following most work in DA classification (Stolcke et al., 2000; Raheja and Tetreault, 2019; Cheng et al., 2019; Ribeiro et al., 2019), we perform a preprocessing step where continuation segments are merged with the immediate previous segment by the same speaker to form a complete DA. This step is illustrated in Table 5.2; this processing results in a different segmentation of speaker sides into turns.

Table 5.1: An example of a (partial) dialog in SWDA original form. The "+" tag is used when there is speech overlap between speaker sides.

Turn#	Speaker	DA#	DA Tag	DA	Words
			...		
3	A	4	aa	accept/agree	I know
3	A	5	sv	opinion	I guess that I guess you consider just things that every day that would you would think of about
3	A	6	sd	statement	see I'm a college student
3	A	7	sd	statement	so I can think of lots of things that my roommate does that bother me
4	B	8	b	backchannel	yeah
5	A	9	+	continued	you know that I think's like is an invasion of my privacy stuff like that
5	A	10	sv	opinion	but I think
6	B	11	b	backchannel	yeah
7	A	12	+	continued	it'd be it is kind of a tough topic
			...		

In joint DA segmentation and classification (DA recognition), we are interested in identifying the boundaries and categories of dialog acts within a turn, assuming known turn boundaries. Given multi-channel recordings, turn boundaries can more reasonably assumed to be known than sentence boundaries, as they are associated with distinctive acoustic cues.

Table 5.2: Example of the same partial dialog in Table 5.1, with continuations merged into the same turn.

Turn#	Speaker	DA#	DA Tag	DA Type	Words
				...	
3	A	4	aa	accept/agree	I know
3	A	5	sv	opinion	I guess that I guess you consider just things that every day that would you would think of about
3	A	6	sd	statement	see I'm a college student
3	A	7	sd	statement	so I can think of lots of things that my roommate does that bother me you know that I think's like is an invasion of my privacy stuff like that
4	B	8	b	backchannel	yeah
5	A	10	sv	opinion	but I think it'd be it is kind of a tough topic
6	B	11	b	backchannel	yeah
				...	

Following (Zhao and Kawahara, 2019), our joint DA recognition task setup is as follows. Given a transcript of a turn (with time-segmented audio), each token in the turn is labeled

Table 5.3: Example partial dialog in Tables 5.1 and 5.2 after preprocessing.

Turn#	Speaker	Word Sequence (Inputs)	Joint Tag Sequence (Labels)
		...	
3	A	i know i ... like that	I E_aa I ... E_sv ... E_sd ... E_sd
4	B	yeah	E_b
5	A	but i think ... tough topic	I I I ... I E_sv
6	B	yeah	E_b
		...	

as E_x if it is the final word in the DA, where x denotes the DA of that utterance; the token is labeled I otherwise. This resulted in an overall tag vocabulary size of 42: I + E_x for x $\in$ 41 DA tags (Jurafsky et al., 1997). In this setup, our joint DA recognition task is essentially a sequence labeling task. In addition to joint DA tag labeling, additional preprocessing steps that we did include:

- Remove non-verbal tokens such as [laughter], [noise], [lipsmack]; i.e. we are not predicting the non-verbal tag "x" (it is not clear in previous work if this was predicted).[1]
- Lowercase all tokens and remove punctuations (similar to parsing).

Table 5.3 shows the same example dialog in Tables 5.1 and 5.2, after these preprocessing steps.

The dialog act sequence of one speaker depends on the previous dialog acts of the other, e.g. it is common for a statement to follow a question. Incorporating dialog history can therefore lead to performance improvement. In joint DA segmentation and classification, DA boundaries are not given, so the context is represented in terms of previous turns. In our work, because of the continuation merging, a previous turn can be overlapping and extend

[1]Based on the implementation we are following, the authors do not predict non-verbal tags either.

beyond the current turn but the full turn is still used as context. For example, in Table 5.3, when predicting segmentation and categories for DAs in the 6th turn, a history of 2 means using turns 4 and 5 as context.

5.1 Models

Similar to parsing experiments, we explore two types of encoder architectures: RNN-seq and transformers. The RNN-seq model is the best performing model from Zhao and Kawahara (2019), extended with the CNN module for learning acoustic-prosodic features as described in Chapter 3. The transformer encoder in our experiments either uses BERT outputs and CNN outputs as features to a feedforward decoder, or includes another full transformer to encode these BERT+CNN features. The models are illustrated in Figure 5.1.

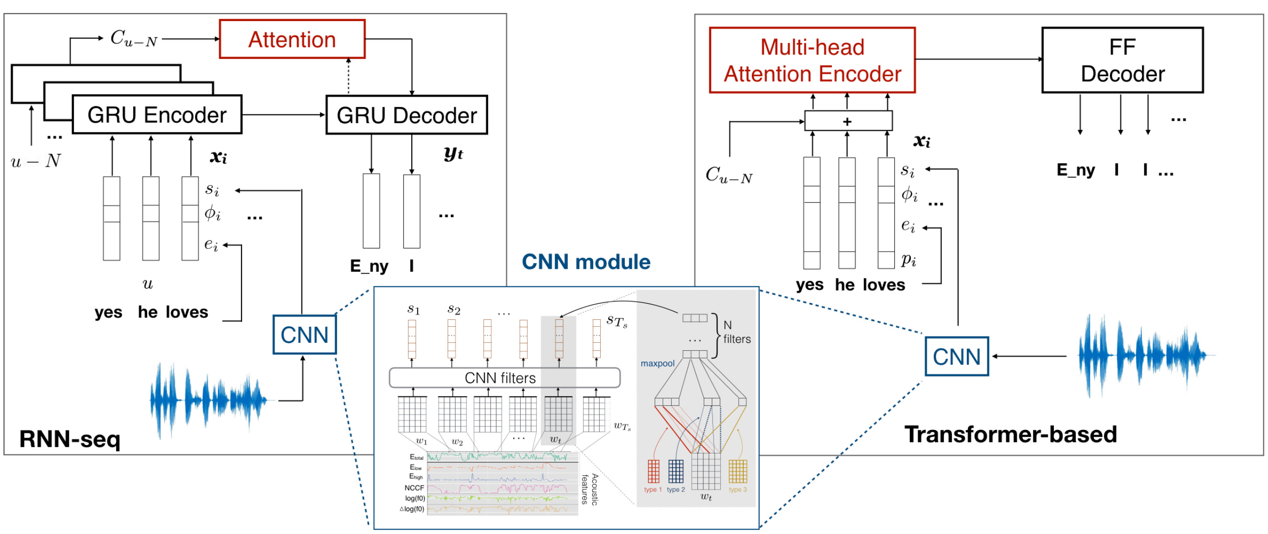


Figure 5.1: Joint dialog act recognition models used. C_{u-N} denotes the context vector, i.e. encoded history from previous turns. In the RNN-seq models, C_{u-N} is obtained from the mean-pooled hidden states of another RNN that was run on previous N turns. For the transformer-based models, C_{u-N} can be obtained by mean- and max-pooling of word features in the previous N turns, then concatenated with word features of the current turn.

As dialog context is important in predicting the current DA category, we also allow for incorporating dialog history. Specifically, for a turn u, the context sequence $T_{u-k}, k \in \{1, \ldots, N\}$ is obtained from another RNN encoder which was run on previous N turns, where T_{u-k} denotes the mean-pooled hidden states of the tokens in turn $u-k$. The attention mechanism operates on the context sequence T_{u-k}, i.e.

$$C_{u-N} = \alpha_k \sum_{k=1}^{N} T_{u-k} \tag{5.1}$$

This history vector C_{u-N} is then concatenated with encoder hidden states h_t for use in decoding, i.e. the context vector c_{t-1} in Equation 3.11 is now:

$$c_{t-1} = FF([C_{u-N}, h_{t-1}]) \tag{5.2}$$

The rest of the operations follow similarly to Equations 3.6 through 3.10 in Section 3.1.

For the transformer-based models, C_{u-N} can be obtained by mean- and max-pooling of word features in the previous N turns. The input to the multihead self-attention encoder is then the concatenation $[C_{u-N}, x_t]$ for all words x_t in the current turn.[2]

5.2 Research Questions and Datasets

The goal of this study is to answer the following questions:

1. Which architecture and word representations work best for joint DA recognition of spoken transcripts? We compare transformer-based vs. RNN-based models, and contextualized embeddings vs. non-contextualized embeddings.
2. Does prosody improve further on top of these strong neural DA recognizers for spontaneous speech? If so, where does prosody benefit most?
3. How does performance on segmentation differ from DA recognition, and what are the error patterns?

[2]Results with context in the transformer-based models were poor so they are not included in the current study. However, we describe one approach where context can be incorporated into transformer encoders for completeness.

For this task, we use the portion of Switchboard annotated with dialog acts (Jurafsky et al., 1997). This subset consists of 1,155 conversations, with train/dev/test splits of 1,115/21/19 conversations. This split does not follow the same convention with standard parsing splits (i.e. conversations numbers 2000s and 3000s for training), but is used in all DA classification studies, e.g. (Stolcke et al., 2000; Raheja and Tetreault, 2019; Cheng et al., 2019; Ribeiro et al., 2019). On average, each conversation has 96.8 turns (min = 14; max = 313; median = 88); and each turn has on average 1.8 DAs (min = 1; max = 30; median = 1).

Time alignment for turns were transferred from MS-State transcripts. Specifically, we ran a token-level sequence matching algorithm[3] to align MS-State tokens and SWDA tokens for each speaker side. The start and end times of MS-State tokens are transferred to SWDA tokens using the following heuristics:

- Error-free tokens or substituted tokens: get the same corresponding start and end times.
- Deleted tokens (present in MS-State, not present in SWDA): no times to be aligned.
- Inserted tokens (not present in MS-State, present in SWDA): get the start time as the end time of the previous SWDA token, and the end time as the start time from the following SWDA token.

Anecdotally, we found few problems with this heuristics, based on later ASR experiments. Briefly, these time alignments were used to extract relevant audio portions to use as inputs to our ASR system, and we observed reasonable WERs.

5.3 Results and Discussion

For both RNN-seq and transformer models, we used the same optimizer, AdamW (Loshchilov and Hutter, 2019), with the same learning schedule as the provided implementation. We report results on all models using the following metrics.

---

[3] `https://docs.python.org/3.6/library/difflib.html`

- DSER: dialog act segmentation error rate, computed as the number of segments wrongly detected divided by number of segments in the reference turn. A segment is said to be correct if all tokens in the reference segment are included in the predicted segment.
- DER: dialog act error rate, computed similarly to DSER, but also taking into account the DA category of the segments detected.
- Macro F1: macro F1 score over joint DA tags in the predicted sequence.
- SLER: Segment Label Error Rate, computed as the word error rate (i.e. edit distance divided by number of reference segments) for the sequence of joint tags, ignoring I tags.

DER, DSER, and F1 scores are used in Zhao and Kawahara (2019), and we report these for comparison. We do nor report their WER scores because these benefit from ASR deletion errors. Instead, we introduce SLER as a measure better suited for ASR transcripts, where the predicted and reference turns might not have a one-to-one alignment. Moreover, in spoken dialog systems, it is often more useful to know the identities of the dialog acts in a turn, regardless of where such speech acts start or end. Tables 5.4 and 5.5 provide examples with calculation of these metrics.

Table 5.4: Example for computing metrics on transcripts. Here DSER $= 2/3 = 0.67$ and DER $= 3/3 = 1$. For SLER, the edit distance is 1 (error in red), there are 3 reference segments, so SLER $= 1/3 = 0.33$.

Reference tags	E_b	I	E_sv	I	E_sd
Predicted tags	E_aa	I	I	E_sv	E_sd
DSER Error	0		1		1
DER Error	1		1		1
Reference tags - utterance level		E_b	E_sv	E_sd	
Predicted tags - utterance level		E_aa	E_sv	E_sd	

Table 5.5: Computation of micro and macro F1 on the same example in Table 5.4. Per-instance F1 is computed as $F1 = 2 * \text{Match}/(\text{Reference} + \text{Predicted})$.

Tag	Match	Reference	Predicted	F1
E_b	0	1	0	0
E_sv	0	1	1	0
E_sd	1	1	1	1
I	1	2	2	0.5
E_aa	0	0	1	0
total (micro F1)	2	5	5	0.4
macro F1				0.3

5.3.1 Assessing Transcript-only Dialog Act Recognition Models

We first study which type of model and embeddings work better for our DA recognition task. Table 5.6 shows the results on SWDA dev set. Our baseline is the best system in Zhao and Kawahara (2019), which learns embeddings jointly with the task and used the RNN-seq model. The authors also considered a longer history (9 previous turns) than we did. Non-contextualized embeddings like GloVe, without enough history length, still underperforms the baseline with non-pretrained embeddings. Similar to parsing results, using contextualized embeddings outperforms learned and non-contextualized embeddings by a large margin in all metrics, even with only the current turn as context (History = 0). The longer context window (History = 2) generally benefits DA recognition, except on the DSER metric; i.e. dialog context seems to benefit DA classification but not segmentation. A possible explanation for this result is that segmentation identification is more local, whereas DA classification can benefit from history, e.g. knowing a question was asked in the previous turn may help predict the statement DA for the current turn. SLER is lower than DER, as expected, since it is a less strict measure but the relative differences between configurations are similar on the two metrics.

Table 5.6: DA recognition results (error rates and macro F1) on SWDA development set. "Baseline" denotes the best system by Zhao and Kawahara (2019), reimplemented as the original paper used a different data split. "BERT" denotes using BERT embeddings as features (no further fine-tuning); "BERT + top layer" denotes fine-tuning the last layer of the BERT model with the DA recognition task; "BERT + transformer" denotes using BERT as features (no fine-tuning) with another transformer encoder before the decoder; "BERT + transformer + top layer" similarly denotes additionally fine-tuning the last layer of BERT.

Model	Embedding	History	DSER	DER	F1	SLER
RNN-seq	Baseline	9	13.9	30.8	0.479	28.6
	GloVe	0	14.1	33.2	0.417	30.9
	GloVe	2	13.9	31.8	0.442	29.2
	BERT	0	**9.8**	28.7	0.429	27.1
	BERT	2	11.9	**27.4**	**0.489**	**25.8**
Transformer-based	BERT	0	24.9	44.1	0.304	42.3
	+ top layer	0	11.2	30.9	0.384	29.9
	+ transformer	0	10.7	30.3	0.367	29.0
	+ transformer + top layer	0	11.1	31.2	0.353	29.6

Compared to RNN-seq, the transformer-based models generally underperformed, even with fine-tuned BERT embeddings and an additional transformer encoder layer. It is possible that our hyperparameter search was not exhaustive enough, especially as transformer models generally require more tuning. Since training transformer models was more computationally expensive than RNN-seq models,[4] we did not explore this model further.

5.3.2 The Role of Prosody

To explore the utility of prosody in DA recognition, we compare model performance with and without prosody features. For all following experiments, the model is the RNN-seq with BERT embeddings. The results are shown in Table 5.7. Compared to models trained on only transcripts, the models using prosody outperform in most metrics, except macro F1. Longer history also helps improve F1, SLER, and DER, but seems to hurt DSER.

Table 5.7: DA recognition results (error rates and macro F1) on the development set, comparing with and without using prosody features. For the model with prosody, the feature set used here is the same as those in parsing (also described in detail in Chapter 3): pitch (f0), energy (E), pause embeddings (r_e), raw pause (r), word duration (δ).

Model	History	DSER	DER	F1	SLER
transcript	0	9.8	28.7	0.429	27.1
transcript+prosody	0	**9.6**	27.5	0.448	26.3
transcript	2	11.9	27.4	**0.489**	25.8
transcript+prosody	2	11.6	**26.9**	0.483	**25.7**

We also studied feature ablation; the performance of the prosody models are shown Table 5.8. Overall, most feature sets gave similar results. However, raw pause duration seems to be more useful than pause embeddings, and word duration is the least useful, likely due to errors in time alignments.

[4]With a sequence length N and hidden dimension size d_h, for each layer, the time complexity is $O(Nd_h^2)$ for RNNs while it is $O(N^2 d_h)$ for transformers. In the parsing case, the sequences were on the sentence (segment) level, so $N << d_h$. In DA recognition, since the sequences are now turns (optionally with history), $N \approx d_h$, making the training much more costly.

Table 5.8: DA recognition ablation results (error rates and macro F1) on the model trained with prosody and no context on SWDA dev set. f0 denotes pitch, E denotes energy, r_e denotes pause embeddings, r denotes raw pause features, and δ denotes word duration features.

Features	DSER	DER	F1	SLER
f0, E, r_e, r, δ	9.8	28.9	0.408	27.1
f0, E, r_e, δ	**9.2**	28.0	0.404	26.5
f0, E, r, δ	9.5	28.1	0.417	26.4
f0, E, δ	10.1	27.5	0.424	**26.3**
f0, E, r_e,	10.0	28.6	0.429	27.1
f0, E, r	9.6	**27.5**	**0.448**	**26.3**
r_e, r	9.8	28.0	0.425	26.6
r_e, r, δ	9.8	28.4	0.434	27.1

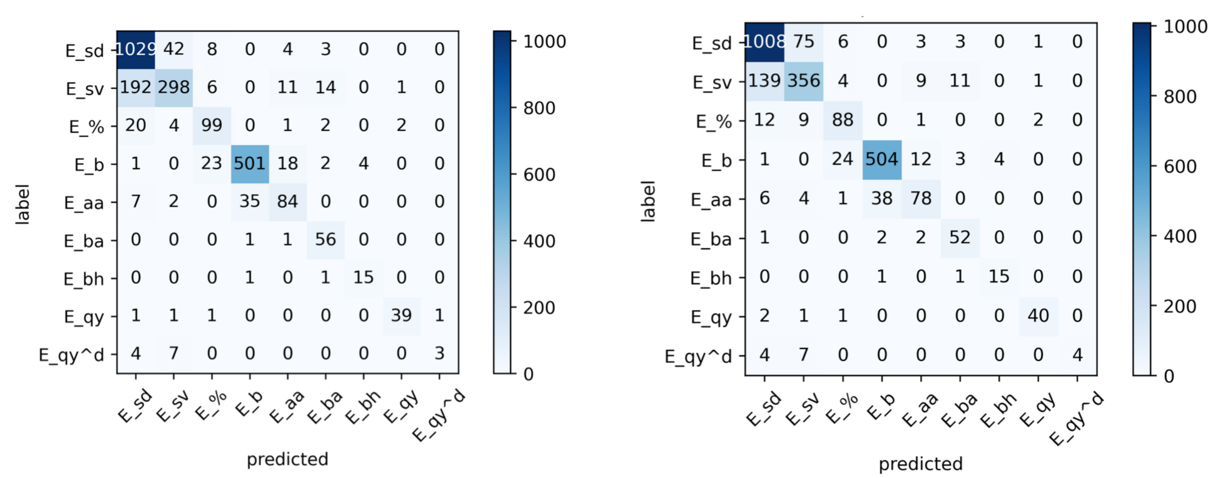
Table 5.9 presents results on SWDA test set using our best models (for models with prosody, the features are f0, E, and r). While prosody helps in the no-context case, it hurts when history is considered. It could be the case that the model trained with prosody is overfitting when history is used, or context should be modeled differently in combination with prosody. This result also suggests that dialog history (transcripts) and prosody are somewhat complementary. That is, prosody helps predict DA segmentation and category more when there is not enough context information. When prosody does help (both in test and dev set), the gain is most prominent for segmentation-related metrics: DSER (4.6% relative improvement in test, 2.1% in dev), and DER (2.4% in test, 4.4% in dev).

5.3.3 Error Analysis

Figure 5.2 shows the confusion matrices on the dev set based on DER errors, from the best performing DA predictors with no context on transcript only (5.2a) and with prosody

Table 5.9: DA recognition results (error rates and macro F1) on test set. Prosody models are those with the best feature set (raw pause, energy, and pitch).

Model	History	DSER	DER	F1	SLER
transcript	0	8.3	30.4	0.418	29.3
transcript+prosody	0	**7.9**	29.7	0.423	28.8
transcript	2	8.6	**26.6**	**0.497**	**25.6**
transcript+prosody	2	9.1	27.1	0.413	26.3



(a) Transcript-only model

(b) Transcript+prosody model

Figure 5.2: Confusion matrices for the for the most common DA classes, comparing the model trained only on transcript (left) and the one trained with prosody (right). Results are on the dev set, model with no context, labels from DER scoring.

(5.2b). Only the most common (and most commonly confused) DA tags are shown. Overall, the model trained with prosody makes similar types of mistakes to those made by the one trained on only transcripts, i.e. the most confusable tags are still statement opinion (sv) vs. non-opinion (sd), and accept/acknowledge (aa) vs. backchannel (b). This is consistent with findings by Jurafsky et al. (1998) and these are DA classes that human annotators also often confuse. The model with prosody, while still not completely eliminating these mistakes, most notably improved over the transcript-only model in the statement (non-opinion) vs. opinion categories.

Some anecdotal examples are shown in Tables 5.10, 5.11, and 5.12, suggesting that prosody in the model helps correct the segmentation error made by the model relying only on transcript. From listening to these samples, this result is likely thanks to the lack of pause and pitch reset in at the confusable word in each instance.

Table 5.10: Example where prosody helped avoid a segmentation error. "sd" is the "statement (non-opinion)" dialog act.

words	i	was	just	like	but	i'm	wasting	my	time
reference tags	I	I	I	I	I	I	I	I	E_sd
predicted tags (transcript)	I	I	I	E_sd	I	I	I	I	E_sd
predicted tags (+prosody)	I	I	I	I	I	I	I	I	E_sd

Specific to segmentation errors, both the DA recognizers with and without prosody tend to misidentify segment boundaries at similar tokens: tokens associated with spontaneous speech phenomena such as fillers, disfluencies, and discourse cues. In particular, out of 3,288 dev segments, the model with prosody missed (wrongly predicted the tag I) in 477 instances, where the most commonly associated tokens are 'uh,' 'know' (from the discourse cue "you know"), and 'it.' The results are similar for the model using only transcripts: it missed 452 segments, in which the most commonly associated tokens are also 'uh,' 'know,' and 'it.'. On

Table 5.11: Example where prosody helped avoid a segmentation error. "%" is the "incomplete/abandon" dialog act, and "qy" is the "yes/no question" dialog act.

words	it's	uh	is your cat an indoor cat or an outdoor	cat
reference tags	I	I	I ... I	E_qy
predicted tags (transcript)	I	E_%	I ... I	E_qy
predicted tags (+prosody)	I	I	I ... I	E_qy

the other hand, there are comparatively fewer boundary insertion errors (wrongly predicting a E_x tag) in both models, though the transcript-only model seems to make more of this type of error: 155 inserted segments vs. 116 by the model with prosody. Within these errors, again the insertion is often associated with spontaneous speech phenomena, such as 'know,' 'uh,' and 'yeah' in both models.

Table 5.12: Example where prosody helped avoid a segmentation error. "ˆ2" is the "collaborative completion" dialog act, and "aa" is the "accept/acknowledge" dialog act.

words	just	sit	around	that	that's	true
reference tags	I	I	E_ˆ2	I	I	E_aa
predicted tags (transcript)	I	I	I	E_aa	I	E_aa
predicted tags (+prosody)	I	I	E_aa	I	I	E_aa

5.4 Summary of Findings

In this chapter, we explored the the task of joint dialog act segmentation and classification (which we refer to as DA recognition). Similar to parsing results, we found that contextualized word representations are useful in yet another task, DA recognition, outperforming non-contextualized representations. An RNN-seq architecture with BERT embeddings improved over the baseline system in Zhao and Kawahara (2019) in all metrics, where the largest gains are in segmentation metrics. Regarding the use of prosody, we showed that our approach for incorporating prosody into encoders helps improve DA recognition further when no dialog history is used, with most gains also from segmentation error improvement. Prosody also seems to help reducing common errors such as opinion vs. statement. When the models misidentify a segment boundary, the associated tokens are often tokens associated with spontaneous speech phenomena, such as disfluencies, fillers, and discourse cues: 'uh,' 'know,' and 'yeah.' Overall, prosody and dialog history seem to be complementary as prosody benefits segmentation while history benefits classification. However, the current framework does not give benefit from combining these on the test data. A factored attention model or some other architecture change might better take advantage of the two components together.

Chapter 6

EFFECTS OF IMPERFECT TRANSCRIPTS

Our experiments so far have been on human-annotated transcripts, which is an unrealistic assumption in most applications. In this chapter, we explore the effect of imperfect transcripts, i.e. ASR output, on our approach.

Prior work in parsing ASR outputs has been limited. One study by Kahn and Ostendorf (2012) explored joint parsing and word recognition by re-ranking ASR hypotheses based on parse features, showing an improvement in word recognition, as measured by word error rate (WER). Another study (Marin and Ostendorf, 2014) explored parsing in the context of domain adaptation and ASR name error detection. The authors showed that using output parse features improved re-scoring word confusion networks (WCN) and benefited the detection of ASR errors and out-of-vocabulary regions. Recent work by Yoshikawa et al. (2016) studied joint parsing with disfluency detection on ASR transcripts. However, they looked at dependency parsing and the method required extending the label set with speech-specific dependency type labels to handle mismatched words. All these studies only used ASR transcripts; prosodic features were not used.

Research in DA recognition on ASR outputs has also not been well studied. In Stolcke et al. (2000), a few experiments looking at joint ASR and DA classification were studied, but improvement on WER was minimal, likely due to the skewed distribution towards statement dialog types. The work by Ang et al. (2005) used a pipeline approach for segmentation and classification. Applying their system on ASR transcripts still saw benefit of using prosodic features, but relatively less than when used on human transcripts. More recently, He et al. (2018) also looked at DA classification on ASR, but not jointly with segmentation. They applied a CNN on segment-level MFCCs, and improved classification accuracy by 2% over classifying only on ASR transcripts. Dang et al. (2020) trained a joint DA segmentation and classification system with an acoustic-to-word model, implicitly providing distributed representations of word-level ASR decoding information. Acoustic features were used but to a limited extent in this work. Specifically, mean and variance of mel filter bank features were the only source of acoustic information. Additionally, it was not clear where performance most suffered by using imperfect transcripts.

In this chapter, we assess our models, which so far have been developed with available human transcripts, on typical ASR system outputs. We first describe the ASR system used, then present our studies on the two tasks, parsing and DA recognition, now with imperfect transcripts.

6.1 Automatic Speech Recognizer

Common to both tasks, we use an off-the-shelf ASR system, ASPiRE (Povey et al., 2016), which was trained on Fisher conversational speech data (Cieri et al., 2004), available in Kaldi's model suite.[1] Briefly, the ASPiRE system was trained using a lattice-free maximum mutual information (LF-MMI) criterion, with computation efficiencies enabled by a phone-level language model, outputs at one third the standard frame rate, and a simpler HMM topology.

For parsing, ASR is run on Treebank sentence units; for DA recognition, ASR is run on turns. The speech segmentation times are based on word times in the hand-corrected Mississippi State (MS) transcripts, using an alignment of Treebank words to the MS transcript words. For each sentence or turn, we retain a set of (up to) 10 best ASR hypotheses (shorter sentences often had fewer hypotheses). In parsing, we use these N-best hypotheses in our experiments; in DA recognition, we only use the 1-best output. Word-level time alignments are a by-product of the ASR system. Table 6.1 presents the WER for dev and test splits in each task.

[1] `https://kaldi-asr.org/models/m1`

Table 6.1: WER (on 1-best) ASR transcripts for each split and task.

Split	Parsing	DA
dev	18.6%	20.9%
test	19.4%	23.6%

6.2 Constituency Parsing Experiments

We explore the problem of parsing ASR outputs by combining previous SOTA parsing systems: a high-quality constituency parser that integrates automatically learned prosodic features, in addition to using powerful contextualized word representations, now applied to imperfect transcripts.

For evaluation, we use F1 score on dependencies and brackets, as implemented in SParseval (Roark et al., 2006). For bracket F1, SParseval requires an alignment between word sequences of the gold and predicted parses. We obtain this alignment with Gestalt pattern matching implemented in python's `difflib` package.[2] SParseval also has the option to compute dependency F1, which does not require the reference and predicted sequences to have the same words, as this measure is based on head-percolated tuples of (h, d, r) where h is the head word, d is the dependent, and r is the relation between h and d. We present F1 scores for both bracket and dependency F1, but will focus on bracket scores as this was the training objective of the original parser.

Comparison with previous work is not straightforward. For example, work by Marin and Ostendorf (2014) used a different dataset; Yoshikawa et al. (2016) reported dependency F1 but not bracket F1, in addition to using a different metric from SParseval; and Kahn and Ostendorf (2012) used automatic sentence segmentation with parse scoring based on the whole turn instead of sentence units. Additionally, each of these works used a different

[2] `https://docs.python.org/3.6/library/difflib.html`

(older) ASR system to generate automatic transcripts, different ranking algorithms, and potentially different time alignments. However, we will mention relevant previous results that are most comparable, e.g. constituency parsing on Switchboard.

Research Questions Our study aims to explore the following questions:

1. What features and ranking methods are useful for selecting better parse hypotheses?
2. How does parsing with multiple ASR hypotheses improve overall parsing performance?
3. Does prosody also help parsing ASR transcripts as it did in human transcripts? What is the impact of considering multiple hypotheses?
4. How does parsing-based selection of ASR hypotheses affect WER? What types of word changes are involved as the parser/ranker chooses a different hypothesis from the 1-best?

Rankers. Given a set of (up to) 10 ASR hypotheses for an utterance,[3] we parse each hypothesis and train a ranker to select the hypothesis with the best F1 score. This process is formulated as a binary classification problem, based on Burges (2010). Specifically, for each set of hypotheses, two sentences a, b are selected as a paired sample with features $F_{ab} = [f_{1a} - f_{1b}, \cdots, f_{Na} - f_{Nb}]$, where f_{ix} is the i-th feature of a sentence $x \in \{a, b\}$, including utterance length, number of disfluent nodes, parser output score, and ASR output score. The corresponding label is $Y_{ab} = 1$ for that pair if the F1 score $s(a)$ of sentence a is greater than that of sentence b, $s(b)$; $Y_{ab} = 0$ otherwise. In constructing the training set, we make sure to always select the pairs with highest F1 score difference, and 10 other random pairs. The ranker is the classifier $C()$ that learns to predict $\hat{Y}_{ab} = C(F_{ab})$. For each type of F1 score, i.e. $s() \in$ {labeled, unlabeled} $\times$ {dependency, bracket}, we trained a separate classifier to optimize for that score.

At test time, two ranking methods were used: point-wise and pair-wise. For point-wise ranking, each hypothesis sentence a is considered individually to produce the probability

[3] 62% of the sentences have < 10 hypotheses; 24% have < 5.

score $P(a) = C(X_a)$ (where X_a is the feature vector associated with pairing sentence a with a sentence of all feature values 0). The best hypothesis is chosen by $\hat{a} = argmax_a P(a)$. We use micro F1 to evaluate the score s_{point} for the set of hypotheses chosen this way. For the pair-wise ranking method, two hypotheses are selected at a time, where the hypothesis for the next round of comparison is chosen based on its higher score. Similarly, micro F1 is used as the score s_{point} to evaluate the hypotheses chosen this way.

We experimented with two types of binary classifiers: logistic regression (LR) and support vector machine classifier (SVC). Hyperparameters of each classifier were tuned on the development set's F1 scores. While many more complex ranking approaches have been proposed (e.g. see Burges et al. (2008)), our focus is to understand what improvements can be made over the 1-best baseline, even with a simple pairwise ranker. More complex ranking algorithms are left for future work.

6.2.1 Results and Discussion

Ranking Features

Table 6.2 shows labeled dependency and bracket F1 scores on the development set, comparing different feature sets, parsing with vs. without prosody, and ranking classifiers. In all settings, the simple LR ranker outperforms SVC, achieving the best dependency F1 score of 0.520 and bracket F1 score of 0.713.

Within LR results, the best performing feature set consists of parse score (raw and normalized by length), ASR score (raw and normalized by length), sentence length, tree depth, and the number of certain types of constituents in the predicted parse: EDITED, INTJ, PP, VP, NP. Between parsing with and without prosody, the parser trained with prosody data slightly outperforms the transcript-based one: 0.713 vs. 0.707 for bracket F1, and 0.520 vs. 0.518 for dependency F1. For the remaining results, we focus on this configuration: LR ranker with the full feature set.

Table 6.2: Labeled dependency and labeled bracket F1 scores on the development set: "core set" denotes the set of features: parser output score, ASR hypothesis score, sentence length, and number of EDITED nodes. "depth" denotes parse tree depth and "*P" denotes the counts of various constituents in the predicted parse (NP, VP, PP, INTJ)

	Ranker	LR		SVC	
Model	feature set	dependency	bracket	dependency	bracket
transcript	core set	0.514	0.701	0.488	0.665
	+ depth	0.512	0.698	0.513	0.649
	+ depth + *P	0.518	0.707	0.470	0.641
+prosody	core set	0.517	0.705	0.483	0.652
	+ depth	0.513	0.706	0.515	0.704
	+ depth + *P	**0.520**	**0.713**	0.481	0.663

Parsing ASR hypotheses vs. 1-best

Table 6.3 presents results comparing the baseline (1-best hypothesis) result with our re-ranked parser as well as several oracle sentence selection schemes. While using only parse score underperforms using the 1-best hypothesis, the re-ranking using parse features improves over the 1-best baseline in both transcript- and transcript+prosody parsers, for all types of evaluations (labeled vs. unlabeled dependency vs. bracket F1). The differences are statistically significant at $p < 0.01$ using the bootstrap test (Efron and Tibshirani, 1993).

The Use of Prosody

SParseval by default does not include EDITED (disfluent) nodes in evaluation. This is a disadvantage for our parser as it was trained to explicitly detect EDITED nodes. We modified SParseval's setting to consider EDITED nodes, and the effect is as large as 0.5%.

Table 6.3: F1 scores on the development set across different sentence selection settings.

	selection by	unlabeled		labeled	
	sentence's	dependency	bracket	dependency	bracket
	1-best ASR	0.624	0.723	0.513	0.699
transcript	parse score	0.588	0.698	0.499	0.664
	best ranker	0.627	0.736	0.518	0.707
+prosody	parse score	0.594	0.706	0.502	0.670
	best ranker	**0.629**	**0.740**	**0.520**	**0.713**
	oracle WER	0.674	0.788	0.555	0.770
	oracle F1	0.702	0.822	0.587	0.798
	gold trans.	0.933	0.938	0.909	0.928

We report our F1 scores in this setting, where disfluent nodes are included in scoring.

Between parsing with and without prosody, using prosody consistently gives better performance, as shown in Table 6.3. Focusing on labeled bracket F1, on the test set (Table 6.4), the relative improvement from using a ranker over the 1-best hypothesis is 1.5% for the best transcript-only parser, and 2% for the prosody parser. Achievable improvement in relation to the gap between oracle F1 score (sentence selected by best F1 score), the prosody parser helps cover 12.4% of the gap, compared to 9.8% by the transcript-only parser.

The closest point of comparison is the study by Kahn and Ostendorf (2012), which reports results on Switchboard using an ASR system. They achieved 24.1% 1-best WER (16.2% N-best oracle WER, N = 50) on the test set. Using reference sentence segmentations (similar to our scenario), they reported an unlabeled dependency F1 score of 0.734 with the oracle result of 0.823. The higher scores (despite the higher WER compared to our system) reflect differences in a parse scoring implementation that incorporates sentence segmentation, and

Table 6.4: Test set F1 scores: "gain" denotes the relative improvement of the system over the 1-best hypothesis; "gap" denotes the gain achieved relative to the oracle score.

	unlabeled		labeled	
	dependency	bracket	dependency	bracket
1-best ASR	0.612	0.700	0.491	0.676
best, transcript	0.619	0.714	0.494	0.687
best, +prosody	0.622	0.715	0.504	0.690
oracle F1	0.704	0.807	0.576	0.783
% gain, transcript	1.1%	2.0%	0.7%	1.5%
% gain, +prosody	1.7%	2.2%	2.5%	2.0%
% gap, transcript	7.2%	13.0%	4.0%	9.8%
% gap, +prosody	11.1%	14.2%	14.7%	12.4%

potentially the exclusion of EDITED nodes as implemented in default SParseval.

Effects on WER

Table 6.5 shows the corresponding test set WER on each setting. While the oracle parser has lower WER, no significant improvement is observed for the parser-rankers over WER of the 1-best hypothesis, which is not surprising as the training objective was not to directly minimize WER.

For further analysis, we compare hypotheses selected by the best (transcript+prosody) parser/re-ranker and the 1-best hypothesis. The best system overall results in a higher WER, but slightly improves in sentences where all 10 hypotheses are available. This result could be because most of the sentences are short (mean = 1.8–3 tokens) for those not producing all 10 hypotheses; only longer sentences (mean = 12.7 tokens) have the full 10 hypotheses.

In sentences where the prosody parser/re-ranker outperformed the 1-best hypothesis,

Table 6.5: WER on SWBD test set, computed depending on the way a hypothesis is selected: the baseline is ASR 1-best hypothesis; the oracle is WER 1-best selection.

			Ranker			
			unlabeled		labeled	
Parser	score	1-best	dependency	bracket	dependency	bracket
-	ASR	0.193	-	-	-	-
transcript	parse	0.243	0.195	0.192	0.201	0.192
+prosody	parse	0.240	0.195	0.201	0.194	0.192
oracle	parse	-	0.159	0.167	0.170	0.160
-	WER	0.115	-	-	-	-

35% of these are associated with better WER, and 23% with worse WER. In both cases, the majority of words involved are function words (82% when WER improved, 77% when WER degraded). Some anecdotal (but common) examples are shown below; **bold text** denotes words corrected by the prosody parser/re-ranker that were otherwise wrong (~~strike out text~~) or missed in the 1-best hypothesis or the transcript-only parser/re-ranker. The better parser appears to favor grammatically correct sentences.

- **and** uh really we 're not doing much at all
- i mean that 's better than george bush ~~you~~ **who** came out and said no
- **do** you like rap music
- **it 's** bigger than just the benefits
- ~~learn~~ **i learned** not necessarily be the center of attention

Finally, we considered whether human transcription errors (Tran et al., 2018; Zayats et al., 2019) could be a confounding factor. Within 5854 test sentences, 1616 have at least one transcription error based on the MS-State corrections. Indeed, as Table 6.6 shows,

Table 6.6: F1 score and WER on the test set, grouped by sentences with and without human transcription errors (based on MS-State corrections).

	bracket F1		WER	
Sentences:	1-best	ranker	1-best	ranker
with error	0.648	0.660	0.235	0.237
without error	0.693	0.707	0.169	0.181

the bracket F1 score in sentences without human transcription errors are higher both for the parser/re-ranker (0.707 vs. 0.660) and the 1-best hypothesis system (0.693 vs. 0.648). Similarly, the WER is lower in sentences without human transcription errors.

6.3 Dialog Act Recognition Experiments

In DA recognition experiments with ASR transcripts, we use the RNN-seq model without dialog history, comparing versions with and without prosody. In the prosody model, we use the best performing set of features: pitch, energy, and raw pause. With ASR output, the number of words in the reference sequence does not necessarily match those in the ASR output. We report the following metrics to evaluate the performance on ASR transcripts; except for SLER and ASER, the following metrics were also used in Dang et al. (2020). Table 6.7 illustrates these computations.

- LER: Label Error Rate, computed as the word-level DA label error rate (i.e. edit distance divided by number of reference tokens).
- SLER: Segment Label Error Rate, computed as the similarly to LER, but with the sequence of collapsed labels, i.e. I labels are ignored.
- DAER: Dialog Act Error Rate, computed similarly to LER, i.e. DAER is also a word error rate on the sequence of labeled tokens but also taking into account the identity

of the dialog act x in the current segment, so I and E_x labels are converted to x for all tokens in that segment.

- SER: Segment Error Rate, computed as the normalized sum of minimum distances between indices of segment positions between the reference and the predicted turn:

$$\text{SER} = \frac{1}{2N_G}\left(\sum_{g\in G} \underset{p\in P}{min}|p-g| + \sum_{p\in P} \underset{g\in G}{min}|p-g|\right)$$

  where G, P are sets of segmentation token indices in reference and prediction, respectively; N_G is the number of tokens in the reference turn.

- ASER: Aligned Segment Error Rate, computed similarly as SER, but after aligning reference and ASR transcript tokens to obtain sequences of the same length. Insertion errors are included in the reference sequence, and deletion errors are included in the ASR sequence, so that each token has an index for SER computation.

- NSER: Segmentation count Error Rate, computed as the difference in number of predicted and reference segment counts, normalized by the number of reference segments.

$$\text{NSER} = \frac{|N_P - N_G|}{N_G}$$

Our results are not comparable with those in (Dang et al., 2020), as they used a different data split for train/dev/test, and it was not clear which conversations were used for which split. Additionally, the ASR system is different, as they report a much higher WER (34% on their test set). Since this is the only work so far using ASR transcripts in joint DA segmentation and classification, we report their results but note that there are many discrepancies.

Research Questions Our study aims to explore the following questions:

1. How is DA recognition affected by imperfect transcripts? Which aspect is more affected: segmentation or classification?
2. How does prosody help DA recognition on ASR transcripts, if at all?

Table 6.7: Example computations of metrics on ASR transcripts. For LER, the label errors are shown in red ("Predicted tags" row); the edit distance here is 4, so LER $= 4/5 = 0.8$. For DAER, the errors also shown in red illustrate edit distance is again 4, but contributed by different tokens, and also result in DAER$= 0.8$. LWER here is $2/3 = 0.67$, NSER $= (4-3)/3 = 0.33$, SER $= \frac{(0+1+0)+(0+1+0+1)}{2*5} = 0.3$. ASER $= \frac{(1+0+0)+(0+1+0+1)}{2*5} = 0.3$.

Reference transcript	right	yes	he	-	loves	cats	-
Reference tags	E_b	E_sv	I	-	I	E_sd	-
ASR transcript	-	yes	she	she	loves	cats	yes
Predicted tags	-	E_ny	I	I	E_sv	E_sd	E_ny
DAER reference sequence		b	sv	sd	sd	sd	
DAER predicted sequence		ny	sv	sv	sv	sd	ny
Reference tags - utterance level		E_b	E_sv	E_sd			
Predicted tags - utterance level		E_ny	E_sv	E_sd	E_ny		
Reference segment indices (G)		0	1	4			
Predicted segment indices (P)		0	3	4	5		
Aligned reference segment indices (G')	0	1				5	
Aligned predicted segment indices (P')		1			4	5	6

6.3.1 Results and Discussion

DA Recognition on Imperfect Transcripts

Table 6.8 and present results of DA recognition on ASR output, compared to the same metrics computed on human transcripts. While predicting the number of segments (macro NSER) and joint labels (LER) suffered the most degradation, classification of DA labels suffered relatively less loss (SLER, DAER, and micro NSER). As expected, the performance on imperfect transcripts is significant worse, especially in metrics that take into account the number of tokens in the sequence, i.e. SER and LER are overly impacted by ASR errors. Our ASER metric is more informative considering such errors, so we will report only on NSER, SLER, DAER, and ASER in the following analyses.

Between segmentation-focused metrics (ASER, NSER) and classification-focused metrics (SLER, DAER), segmentation tends to be more affected (degrades more) compared to classification when imperfect transcripts are used. This further motivates the importance of looking at the segmentation problem, as previous works that only consider classification might be underestimating the challenge in this DA recognition task.

Table 6.8: Macro and micro DA recognition results (error rates) on dev set, comparing DA recognition on human vs. ASR transcripts. LER and SER are overly sensitive to ASR errors.

F1	Model	SLER	LER	DAER	NSER	SER	ASER
Macro	transcript	0.271	0.084	0.220	0.079	0.041	
	asr	0.405	0.251	0.418	0.110	0.177	0.117
	%Δ(asr, trans)	49.4%	198.1%	90.0%	38.0%	333.8%	187.2%
Micro	transcript	0.291	0.042	0.213	0.090	0.061	
	asr	0.372	0.109	0.269	0.105	1.244	0.107
	%Δ(asr, trans)	27.8%	157.8%	26.2%	15.8%	1938.5%	74.8%

The Role of Prosody

Table 6.9 presents our DA recognition results with and without prosody. The models with prosody improve over those without on most metrics, except NSER and ASER, both for human and ASR transcripts. The NSER only takes into account the number of segments in respective turns, so it is sensitive to missed or inserted segment tags. Most importantly, using prosody in the ASR setups gives a larger gain (or smaller loss) than in the perfect transcript setup: improving macro SLER and DAER by 15-17% on ASR but only 2-3% on perfect transcripts. Similarly, micro SLER improves by 9% on ASR, using prosody, but only 1% on human transcripts.

Table 6.9: DA recognition results (error rates) on dev set, comparing DA recognition on human vs. ASR transcripts using the model trained with and without prosody.

F1	Model	SLER	DAER	NSER	ASER
Macro	transcript	0.271	0.220	0.079	0.041
	transcript+prosody	0.263	0.214	0.083	0.043
	%Δ(+prosody, transcript)	3.0%	2.8%	-4.0%	-5.8%
	asr	0.405	0.418	0.110	0.117
	asr+prosody	0.333	0.347	0.115	0.118
	%Δ(+prosody, asr)	17.7%	17.1%	-5.0%	-0.5%
Micro	transcript	0.291	0.213	0.090	0.061
	transcript+prosody	0.288	0.206	0.110	0.066
	%Δ(+prosody, transcript)	0.9%	3.1%	-21.6%	-7.6%
	asr	0.372	0.269	0.105	0.107
	asr+prosody	0.337	0.259	0.113	0.105
	%Δ(+prosody, asr)	9.3%	3.8%	-8.4%	1.8%

Comparing the degradation due to imperfect transcript, Table 6.10 suggests that using prosody leads to a less severe performance drop compared to using only ASR transcripts: relative error increase is smaller for all metrics (except macro NSER) when prosody is used with ASR transcripts.

Table 6.10: Relative differences in macro and micro DA recognition results on dev set, with and without prosody.

F1	Model	SLER	DAER	NSER	ASER
Macro	%Δ(asr, transcript)	49.4%	90.0%	38.0%	187.2%
	%Δ(asr, transcript) + prosody	26.7%	62.1%	39.4%	172.6%
Micro	%Δ(asr, transcript)	27.8%	26.2%	15.8%	74.8%
	%Δ(asr, transcript) + prosody	17.0%	25.3%	3.3%	59.5%

Finally, we show corresponding DA recognition results on the test set in Tables 6.11 and 6.12. With the caveat of large discrepancies in ASR systems and experiment setups, our approach of integrating prosody also improved over a recent work by (Dang et al., 2020).

6.4 Summary of Findings

In this chapter, we assessed the performance of our developed models on imperfect transcripts, i.e. transcripts from a typical ASR system. In our parsing experiments, we tested a SOTA parser that incorporates prosodic information. Our simple re-ranking framework using standard parse tree features and ASR scores achieved 12–14% improvements in F1 scores over 1-best parses relative to the oracle N-best gain. In all settings, parsing using prosodic features outperforms parsing with only transcripts. When parsing improvement is observed, words involved in the hypothesis selection change are mostly function words (around 80%).

For DA recognition, we showed that prosody also helps improve error rates over models

Table 6.11: DA recognition results on dev set. All metrics are macro averages.

Model	SLER	DAER	NSER	ASER
transcript	0.293	0.247	0.072	0.040
transcript+prosody	0.288	0.243	0.071	0.040
%Δ(+prosody, transcript)	1.8%	1.4%	2.0%	0.7%
asr+fbank (Dang et al., 2020)	-	-	0.148	-
asr	0.459	0.484	0.123	0.141
asr+prosody	0.391	0.416	0.124	0.148
%Δ(+prosody, asr)	14.8%	13.9%	-0.9%	-4.3%
%Δ(asr, transcript)	56.6%	96.0%	70.8%	253.6%
%Δ(asr, transcript) + prosody	35.8%	71.1%	75.9%	271.4%

Table 6.12: DA recognition results on dev set. All metrics are micro averages.

Model	SLER	DAER	NSER	ASER
transcript	0.312	0.247	0.073	0.050
transcript+prosody	0.310	0.242	0.081	0.049
%Δ(+pros, trans)	0.8%	1.9%	-11.0%	3.3%
asr+fbank (Dang et al., 2020)	-	0.351	-	-
asr	0.424	0.330	0.070	0.101
asr+prosody	0.387	0.303	0.083	0.100
%Δ(+prosody, asr)	8.6%	8.1%	-17.7%	1.8%
%Δ(asr, transcript)	35.7%	33.5%	-3.7%	101.8%
%Δ(asr, transcript) + prosody	25.0%	25.1%	2.1%	104.9%

trained on only transcripts, where the relative gains are significantly higher in the ASR setting than in the perfect transcript setting. In assessing performance on ASR transcripts, we also introduced new metrics, SLER and ASER, that are more informative and less sensitive to ASR errors. Finally, we found that DA segmentation is more severely affected than DA classification when ASR transcripts are used, motivating further research in joint DA recognition instead of focusing only on classification.

Chapter 7
CONCLUSION

In this final chapter, we summarize our findings and contributions in Section 7.1, and suggest directions for future research in Section 7.2.

7.1 Summary of Contributions

In this thesis, we have made the following contributions.

We present a computational model of prosody that automatically learns acoustic representations useful for spoken language understanding. This model learns to summarize frame-based speech features such as fundamental frequency and energy via a CNN, and is trained jointly with a downstream task. Our model therefore can automatically learn task-specific speech signal representations without the need for expensive human annotations. In experiments with human-human conversational speech, we demonstrate the impact on two tasks: constituency parsing and dialog act recognition (segmentation and classification).

Our first sets of results provide new examples showing that contextualized embeddings are indeed powerful tools useful in a range of NLP tasks. Despite being trained on written text, these embeddings provided significant gains over non-contextualized ones in all our experiments. Given these strong baselines, we show that our use of prosody can still benefit parsing and DA recognition, for both hand transcripts and ASR transcripts. Additionally, we show analyses of cases where prosody most benefits these two tasks, contributing to a better understanding of how acoustic-prosodic information can be integrated into NLP systems.

For constituency parsing, we show that prosody most benefits longer and more disfluent sentences, helping disambiguate and avoid attachment errors, and detect disfluencies. We show empirically that spontaneous speech and read speech differ in both the lexical style and prosodic style, where a parser trained on spontaneous speech suffers less performance degradation when evaluated on read speech, as opposed to vice versa. This result suggests that spontaneous speech in general is useful for training AI systems, both in terms of word choice and prosody. Our finding further motivates the importance of studying natural, spontaneous speech when developing language technology.

We also assessed our parsers on imperfect, i.e. ASR transcripts. Using a simple re-ranking system, we show that prosody still helps parsing, yielding improvements over 1-best parses relative to the oracle N-best gain. In all settings, parsing using prosodic features outperforms parsing with only transcript information. In relation to WER, the better parser/re-ranker appears to favor grammatically correct sentences.

In DA recognition on independent turns, we show that using prosody improves joint segmentation and classification, with more gains achieved mainly thanks to segmentation and correction of opinion DAs. Overall, our experiments suggest that prosody and dialog history seem to be complementary, as prosody benefits segmentation while turn history benefits classification. However, our current framework does not give benefit from combining these two sources of context on the test data.

In assessing our DA recognition system on ASR transcripts, similar to parsing results, we show that prosody is still beneficial, where the relative reduction in error rates is significantly better in the ASR setting than in the hand transcript setting. We also introduced new metrics, segment label error rate (SLER) and aligned segment error rate (ASER), which are more informative and less sensitive to ASR errors. Additionally, we found that segmentation is more severely affected than DA classification when ASR transcripts are used, motivating further research in joint DA recognition instead of focusing only on classification.

Both parsing and dialog act recognition are important components of automatic spoken language processing systems. Our findings in this thesis may lead to better understanding of prosody in human-human communication, which then can be applied to human-computer interaction systems. Consequently, our contributions have the potential to improve language systems, by facilitating accessibility via more natural human-computer interactions, especially in education, health care, elder care, and numerous other AI-assisted domains.

7.2 Future Directions

For future work, some promising directions we think are worth exploring include: new architectures for both the encoder and decoder, new methods of integrating prosody-sensitive language processing in ASR, and assessment of impact of prosody on other SLU tasks.

On the encoder side, transformers have recently become more common, and sometimes even a more popular alternative to RNNs. However, it is not straightforward to train transformers, as they can be hard to tune effectively with smaller batch sizes. As we found in our DA recognition experiments, an under-tuned transformer architecture yielded poor results. Faster and more trainable transformer encoders might therefore provide some gain. Recent promising models include the Reformer (Kitaev et al., 2020), Performer (Choromanski et al., 2020), Longformer (Beltagy et al., 2020), Linformer (Wang et al., 2020), and Linear Transformers (Katharopoulos et al., 2020), among others. CNN modeling enhancements can also benefit from doing cross attention between the text and speech modalities, as this was shown to be important in earlier work (Shriberg and Stolcke, 2004). Additionally, better human-computer interaction might require incremental speech processing, which would necessitate more significant encoder architecture changes.

New architectures for the decoder can also be explored. So far, we have only considered RNN decoders and FF decoders as used in tagging tasks. A full transformer decoder can also be explored, though the issue of efficiency and trainability should also be taken into account, as with transformer encoders. As our DA recognition results showed, dialog history and prosody are potentially complementary, with prosody helping segmentation more and history helping classification more, although the combination did not help both. It may be useful to introduce a factored attention mechanism to better use dialog and prosody contexts in a more flexible manner, so that better performance in one aspect (segmentation) does not hurt that in another (classification). Modeling of speakers as context would also be a promising direction, as demonstrated in Cheng et al. (2019). This is additionally interesting from the scientific perspective, as modeling speaker interactions, both through words and prosodic cues, can help us better understand aspects of human conversation dynamics, such as entrainment.

How to leverage SOTA ASR systems is another promising direction, as integration with ASR systems should be considered for practical impact. While end-to-end joint learning of ASR and SLU, as in Dang et al. (2020), is a popular direction, an advantage of independent learning (or at least pretraining) of ASR is that transcribing speech is much less costly than annotating language structures, so training resources for SLU are more costly than for ASR. With a pipeline approach, there are questions of how to account for ASR uncertainly and how to align acoustic cues to words. Not all ASR systems provide n-best hypotheses, so it will be important to explore different ways to represent and use this uncertainty information (e.g. hypothesis probability scores, full lattices, etc.).

Another important and practical issue to consider is imperfect time alignments. So far, we relied on human transcripts or ASR systems with generally reliable time alignments. This is not necessarily the case, as SOTA ASR systems do not always provide time alignments, or the alignments might be poor. This issue also gives rise to the question of (de)coupling acoustic features and word features. For example, which fusion or attention mechanisms are appropriate — local attention vs. global attention on the whole sequence, word-level lexical-prosodic feature fusion vs. sequence level fusion.

Finally, we should assess the impact of integrating prosody on other SLU tasks, or more general versions with even less ideal transcripts, e.g. not having hand-annotated sentence or turn boundaries. The natural next task for constituency parsing would be joint parsing and sentence segmentation, and turn detection for both parsing and DA recognition. As ASR improves, human-computer communication can become more natural and therefore it is more promising for prosody to be useful. Examples of such systems include dialog state tracking, spoken chat intent detection, personal tutoring bots, and many more.

BIBLIOGRAPHY

Li Aijun. Chinese Prosody and Prosodic Labeling of Spontaneous Speech. In *Proc. Speech Prosody*, 2002.

Jeremy Ang, Yang Liu, and Elizabeth Shriberg. Automatic Dialog Act Segmentation and Classification in Multiparty Meetings. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages I/1061–I/1064–Vol.–1, 2005. doi: 10.1109/ICASSP.2005.1415300.

Harish Arsikere, Arunasish Sen, A. Prathosh, and Vivek Tyagi. Novel Acoustic Features for Automatic Dialog-Act Tagging. *Proc. Acoustics, Speech and Signal Processing (ICASSP)*, pages 6105–6109, 2016.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proc. Learning Representations (ICLR)*, 2015.

Mary Beckman and Jan Edwards. Intonational Categories and the Articulatory Control of Duration. In *Speech Perception, Production and Linguistic Structure*, pages 359–375. IOS Press, Tokyo : Ohmsha, 1992.

Iz Beltagy, Matthew Peters, and Arman Cohan. Longformer: The Long-Document Transformer. *arXiv:2004.05150*, 2020.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A Neural Probabilistic Language Model. *Journal of Machine Learning Research (JMLR)*, 3:1137–1155, March 2003. ISSN 1532-4435. URL `http://dl.acm.org/citation.cfm?id=944919.944966`.

Taylor Berg-Kirkpatrick, David Burkett, and Dan Klein. An Empirical Investigation of Statistical Significance in NLP. In *Proc. Spoken Language Processing (ICSLP)*, Jeju Island, South Korea, July 2012.

David Blei, Andrew Ng, and Michael Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research (JMLR)*, 3:993–1022, 2003. ISSN 1532-4435. doi: http://dx.doi.org/10.1162/jmlr.2003.3.4-5.993. URL `http://portal.acm.org/citation.cfm?id=944937`.

Sara Bögels, Herbert Schriefers, Wietske Vonk, and Dorothee Chwilla. Prosodic Breaks in Sentence Processing Investigated by Event-Related Potentials. *Language and Linguistics Compass*, 5(7):424–440, 2011. doi: 10.1111/j.1749-818X.2011.00291.x. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1749-818X.2011.00291.x`.

Mara Breen, Laura Dilley, Marti Bolivar, John Kraemer, and Edward Gibson. Inter-Transcriber Reliability for Two Systems of Prosodic Annotation: ToBI (Tones and Break Indices) and RaP (Rhythm and Pitch). In *Corpus Linguistics and Linguistic Theory*, volume 8, 01 2006. doi: 10.1515/cllt-2012-0011.

Jason Brenier, Daniel Cer, and Daniel Jurafsky. The Detection of Emphatic Words Using Acoustic and Lexical Features. In *Proc. Interspeech*, pages 3297–3300, 01 2005.

Ivan Bulyko and Mari Ostendorf. Efficient Integrated Response Generation from Multiple Targets Using Weighted Finite State Transducers. *Computer Speech & Language*, 16(3):533–550, 2002. ISSN 0885-2308. doi: https://doi.org/10.1016/S0885-2308(02)00023-2. URL `http://www.sciencedirect.com/science/article/pii/S0885230802000232`. Spoken Language Generation.

Chris Burges, Krysta Svore, Qiang Wu, and Jianfeng Gao. Ranking, Boosting, and Model Adaptation. Technical Report MSR-TR-2008-109, MSR, 2008. URL `https://www.microsoft.com/en-us/research/publication/ranking-boosting-and-model-adaptation/`.

Christopher Burges. From Ranknet to LambdaRank to LambdaMart: An Overview. Technical report, MSR, 2010.

Dani Byrd, Jelena Krivokapić, and Sungbok Lee. How Far, How Long: On the Temporal Scope of Prosodic Boundary Effects. *The Journal of the Acoustical Society of America*, 120(3):1589–1599, 2006.

Sasha Calhoun, Jean Carletta, Jason Brenier, Neil Mayo, Dan Jurafsky, Mark Steedman, and David Beaver. The NXT-Format Switchboard Corpus: A Rich Resource for Investigating the Syntax, Semantics, Pragmatics and Prosody of Dialogue. *Language Resources and Evaluation*, 44(4):387–419, 2010.

Houwei Cao, Štefan Beňuš, Ruben Gur, Ragini Verma, and Ani Nenkova. Prosodic Cues for Emotion: Analysis with Discrete Characterization of Intonation. In *Proc. International Conference on Speech Prosody*, pages 130–134, 2014. doi: 10.21437/SpeechProsody.2014-14. URL `http://dx.doi.org/10.21437/SpeechProsody.2014-14`.

Wallace Chafe. Givenness, Contrastiveness, Definiteness, Subjects, Topics, and Point of View. In *Subject and Topic*, pages 25–55. Academic Press, New York, 1976.

Nattanun Chanchaochai, Christopher Cieri, Japhet Debrah, Hongwei Ding, Yue Jiang, Sishi Liao, Mark Liberman, Jonathan Wright, Jiahong Yuan, Juhong Zhan, and Yuqing Zhan. GlobalTIMIT: Acoustic-Phonetic Datasets for the World's Languages. In *Proc. Interspeech*, pages 192–196, 2018.

Eugene Charniak and Mark Johnson. Edit Detection and Parsing for Transcribed Speech. In *Proc. North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 118–126. Association for Computational Linguistics, 2001.

Ken Chen, Mark Hasegawa-Johnson, Aaron Cohen, Sarah Borys, Sung-Suk Kim, Jennifer Cole, and Jeung-Yoon Choi. Prosody Dependent Speech Recognition on Radio News Corpus of American English. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):232–245, 2006. doi: 10.1109/TSA.2005.853208.

Xi Chen, Sarah Ita, Michelle Levine, Marko Mandic, and Julia Hirschberg. Acoustic-Prosodic and Lexical Cues to Deception and Trust: Deciphering How People Detect Lies. *Transactions of the Association for Computational Linguistics*, 8:199–214, 2020. doi: 10.1162/tacl_a_00311.

Hao Cheng, Hao Fang, and Mari Ostendorf. A Dynamic Speaker Model for Conversational Interactions. In *Proc. North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 2772–2785. Association for Computational Linguistics, 2019. URL `https://www.aclweb.org/anthology/N19-1284`.

Kyunghyun Cho, Bart Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179. URL `https://www.aclweb.org/anthology/D14-1179`.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, David Belanger, Lucy Colwell, and Adrian Weller. Masked Language Modeling for Proteins via Linearly Scalable Long-Context Transformers, 2020.

Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, KyungHyun Cho, and Yoshua Bengio. Attention-Based Models for Speech Recognition. *CoRR*, abs/1506.07503, 2015.

Christopher Cieri, David Graff, Owen Kimball, Dave Miller, and Kevin Walker. Fisher English Training Speech Part 1 Transcripts, LDC2004T19. Web Download, 2004.

Delphine Dahan. Prosody and Language Comprehension. *Wiley Interdisciplinary Reviews: Cognitive Science*, 6(5):441–452, 2015. doi: 10.1002/wcs.1355. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/wcs.1355`.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. *ArXiv*, abs/1901.02860, 2019.

Viet-Trung Dang, Tianyu Zhao, Sei Ueno, Hirofumi Inaguma, and Tatsuya Kawahara. End-to-End Speech-to-Dialog-Act Recognition. In *Proc. Interspeech*, pages 3910–3914, 2020.

Neeraj Deshmukh, Andi Gleeson, Joseph Picone, Aravind Ganapathiraju, and Jonathan Hamaker. Resegmentation of SWITCHBOARD. In *Proc. Spoken Language Processing (ICSLP)*, 1998.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proc. North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 4171–4186. Association for Computational Linguistics, 2019.

Laura Dilley. *The Phonetics and Phonology of Tonal Systems*. PhD thesis, Massachusetts Institute of Technology, 2005.

Markus Dreyer and Izhak Shafran. Exploiting Prosody for PCFGs with Latent Annotations. In *Proc. Interspeech*, pages 450–453, 2007.

Susan Dumais. Latent Semantic Analysis. *Annual Review of Information Science and Technology*, 38(1):188–230, 2004. doi: https://doi.org/10.1002/aris.1440380105. URL `https://asistdl.onlinelibrary.wiley.com/doi/abs/10.1002/aris.1440380105`.

Bradley Efron and Robert Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall/CRC, 1993.

Florian Eyben, Martin Wöllmer, and Björn Schuller. OpenSMILE: The Munich Versatile and Fast Open-Source Audio Feature Extractor. In *Proc. ACM Multimedia*, MM '10, page 1459–1462, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781605589336. doi: 10.1145/1873951.1874246. URL https://doi.org/10.1145/1873951.1874246.

Gunnar Fant and Anita Kruckenberg. On the Quantal Nature of Speech Timing. In *Proc. Spoken Language Processing (ICSLP)*, volume 4, pages 2044–2047–vol.4, Oct 1996. doi: 10.1109/ICSLP.1996.607202.

Cécile Fourgeron and Patricia Keating. Articulatory Strengthening at Edges of Prosodic Domains. *The Journal of the Acoustical Society of America*, 101(6):3728–3740, 1997.

Kathleen Fraser, Frank Rudzicz, and Elizabeth Rochon. Using Text and Acoustic Features to Diagnose Progressive Aphasia and Its Subtypes. In *Proc. Interspeech*, 2013.

Kathleen Fraser, Jed Meltzer, and Frank Rudzicz. Linguistic Features Identify Alzheimer's Disease in Narrative Speech. *Journal of Alzheimer's disease : JAD*, 49(2):407–422, 2016. ISSN 1387-2877. doi: 10.3233/jad-150520. URL https://doi.org/10.3233/JAD-150520.

David Gaddy, Mitchell Stern, and Dan Klein. What's Going On in Neural Constituency Parsers? An Analysis. In *Proc. North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Volume 1 (Long Papers)*, pages 999–1010. Association for Computational Linguistics, 2018.

John Garofolo, David Graff, Doug Paul, and David Pallett. *CSR-I (WSJ0) Complete LDC93S6A*. Linguistic Data Consortium, 1993.

John Godfrey and Edward Holliman. *Switchboard-1 Release 2*. Linguistic Data Consortium, 1993.

Carlos Gómez-Rodríguez and David Vilares. Constituent Parsing as Sequence Labeling. In *Proc. Empirical Methods in Natural Language Processing (EMNLP)*, pages 1314–1324, 2018.

Michelle Gregory, Mark Johnson, and Eugene Charniak. Sentence-Internal Prosody Does Not Help Parsing the Way Punctuation Does. In *Proc. North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 81–88. Association for Computational Linguistics, 2004.

Martine Grice, Stefan Baumann, and Ralf Benzmüller. German Intonation in Autosegmental-Metrical Phonology. In *Prosodic Typology: The Phonology of Intonation and Phrasing*, chapter 3. Oxford University Press, Oxford, 2005.

Franćois Grosjean, Lysiane Grosjean, and Harlan Lane. The Patterns of Silence: Performance Structures in Sentence Production. *Cognitive Psychology*, 1979.

Barbara Grosz. *The Representation and Use of Focus in Dialogue Understanding.* PhD thesis, University of California, Berkeley, 1977.

Barbara Grosz and Julia Hirschberg. Some Intonational Characteristics of Discourse Structure. In *ICSLP*, pages 429–432, 1992.

Michael Halliday. *Intonation and Grammar in British English.* The Hague: Mouton, 1967a.

Michael Halliday. Notes on Transitivity and Theme in English: Part 2. *Journal of Linguistics*, 3(2):199–244, 1967b. ISSN 00222267, 14697742. URL `http://www.jstor.org/stable/4174965`.

Mark Hasegawa-Johnson, Ken Chen, Jennifer Cole, Sarah Borys, Sung-Suk Kim, Aaron Cohen, Tong Zhang, Jeung-Yoon Choi, Heejin Kim, Taejin Yoon, and Sandra Chavarria. Simultaneous Recognition of Words and Prosody in the Boston University Radio Speech Corpus. *Speech Communication*, 46(3):418–439, 2005. ISSN 0167-6393. doi: https://

doi.org/10.1016/j.specom.2005.01.009. URL http://www.sciencedirect.com/science/article/pii/S0167639305001007. Quantitative Prosody Modelling for Natural Speech Description and Generation.

Xuanli He, Quan Tran, William Havard, Laurent Besacier, Ingrid Zukerman, and Gholamreza Haffari. Exploring Textual and Speech information in Dialogue Act Classification with Speaker Domain Adaptation. In *Proceedings of the Australasian Language Technology Association Workshop 2018*, pages 61–65, Dunedin, New Zealand, December 2018. URL https://www.aclweb.org/anthology/U18-1007.

Julia Hirschberg and Christine Nakatani. Acoustic Indicators of Topic segmentation. In *Proc. International Conference on Spoken Language Processing (ICSLP)*, 1998.

Julia Hirschberg, Diane Litman, and Marc Swerts. Prosodic and Other Cues to Speech Recognition Failures. *Speech Communication*, 43:155–175, 06 2004. doi: 10.1016/j.specom.2004.01.006.

Daniel Hirst. *La Représentation Linguistique des Systèmes Prosodiques : Une Approche Cognitive*. PhD thesis, Université de Provence, 1987. URL http://www.theses.fr/1987AIX10055.

Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997.

Yan Huang and Julia Hirschberg. Pragmatics and Prosody, November 2015. URL http://www.oxfordhandbooks.com/view/10.1093/oxfordhb/9780199697960.001.0001/oxfordhb-9780199697960-e-28.

Paria Jamshid and Mark Johnson. Improving Disfluency Detection by Self-Training a Self-Attentive Model. In *Proc. Association for Computational Linguistics (ACL)*, pages 3754–3763, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.346. URL https://www.aclweb.org/anthology/2020.acl-main.346.

Paria Jamshid, Yufei Wang, and Mark Johnson. Neural Constituency Parsing of Speech Transcripts. In *Proc. Human Language Technologies (HLT) Volume 1 (Long and Short Papers)*, pages 2756–2765, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1282. URL `https://www.aclweb.org/anthology/N19-1282`.

Mark Johnson and Eugene Charniak. A TAG-based Noisy Channel Model of Speech Repairs. In *Proc. Association for Computational Linguistics (ACL)*, pages 33–40. Association for Computational Linguistics, 2004.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel Weld, Luke Zettlemoyer, and Omer Levy. SpanBERT: Improving pre-training by representing and predicting spans. *arXiv preprint arXiv:1907.10529*, 2019.

Sun-Ah Jun. K-ToBI (Korean ToBI) Labeling Conventions: Version 3. *Speech Sciences*, 7: 143–169, 01 2000.

Daniel Jurafsky, Elizabeth Shriberg, and Debra Biasca. Switchboard SWBD-DAMSL Shallow-Discourse-Function Annotation Coders Manual, Draft 13. Technical Report 97-02, University of Colorado, Boulder Institute of Cognitive Science, Boulder, CO, 1997.

Daniel Jurafsky, Elizabeth Shriberg, Barbara Fox, and Traci Curl. Lexical, Prosodic, and Syntactic Cues for Dialog Acts. In *Discourse Relations and Discourse Markers*, 1998. URL `https://www.aclweb.org/anthology/W98-0319`.

Jeremy Kahn and Mari Ostendorf. Joint Reranking of Parsing and Word Recognition with Automatic Segmentation. *Computer Speech & Language*, pages 1–19, 2012.

Jeremy Kahn, Matthew Lease, Eugene Charniak, Mark Johnson, and Mari Ostendorf. Effective Use of Prosody in Parsing Conversational Speech. In *Proc. Human Language Technologies and Empirical Methods in Natural Language Processing (HLT-EMNLP)*, pages 233–240, 2005.

Nal Kalchbrenner and Phil Blunsom. Recurrent Convolutional Neural Networks for Discourse Compositionality. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, pages 119–126, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W13-3214`.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention. In *Proc. International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pages 5156–5165, Virtual, 13–18 Jul 2020. PMLR.

Hamed Khanpour, Nishitha Guntakandla, and Rodney Nielsen. Dialogue Act Classification in Domain-Independent Conversations Using a Deep Recurrent Neural Network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2012–2021, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee. URL `https://www.aclweb.org/anthology/C16-1189`.

Ji-Hwan Kim and Philip Woodland. A Combined Punctuation Generation and Speech Recognition System and Its Performance Enhancement Using Prosody. *Speech Communication*, 41(4):563–577, 2003. ISSN 0167-6393. doi: https://doi.org/10.1016/S0167-6393(03)00049-9. URL `http://www.sciencedirect.com/science/article/pii/S0167639303000499`.

Diederik Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980, 2014.

Nikita Kitaev and Dan Klein. Constituency Parsing with a Self-Attentive Encoder. In *Proc. Association for Computational Linguistics (ACL) (Volume 1: Long Papers)*, pages 2676–2686. Association for Computational Linguistics, 2018.

Nikita Kitaev, Steven Cao, and Dan Klein. Multilingual Constituency Parsing with Self-Attention and Pre-Training. In *Proc. Association for Computational Linguistics (ACL)*, pages 3499–3505. Association for Computational Linguistics, July 2019. doi: 10.18653/v1/P19-1340. URL https://www.aclweb.org/anthology/P19-1340.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The Efficient Transformer. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rkgNKkHtvB.

Greg Kochanski, Esther Grabe, James Coleman, and Burton Rosner. Loudness Predicts Prominence: Fundamental Frequency Lends Little. *The Journal of the Acoustical Society of America*, 118(2):1038–1054, 2005. doi: 10.1121/1.1923349. URL https://doi.org/10.1121/1.1923349.

Jáchym Kolář, Elizabeth Shriberg, and Yang Liu. Using Prosody for Automatic Sentence Segmentation of Multi-party Meetings. In *Text, Speech and Dialogue*, pages 629–636, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

Jacqueline Kory. Storytelling with Robots: Effects of Robot Language Level on Children's Language Learning. Master's thesis, Massachusetts Institute of Technology, 2014.

Jonathan Kummerfeld, David Hall, James Curran, and Dan Klein. Parser Showdown at the Wall Street Corral: An Empirical Investigation of Error Types in Parser Output. In *Proc. Empirical Methods in Natural Language Processing*, 2012.

D. Robert Ladd. Declination "reset" and the Hierarchical Organization of Utterances. *The Journal of the Acoustical Society of America*, 84(2):530–544, 1988. doi: 10.1121/1.396830. URL https://doi.org/10.1121/1.396830.

D. Robert Ladd and Rachel Morton. The Perception of Intonational Emphasis: Continuous or Categorical? *Journal of Phonetics*, 25(3):313–342, 1997. ISSN 0095-4470. doi: https://doi.org/10.1006/jpho.1997.0046. URL http://www.sciencedirect.com/science/article/pii/S0095447097900462.

Ji Lee and Franck Dernoncourt. Sequential Short-Text Classification with Recurrent and Convolutional Neural Networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 515–520, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1062. URL `https://www.aclweb.org/anthology/N16-1062`.

Ilse Lehiste. Phonetic Disambiguation of Syntactic Ambiguity. *The Journal of the Acoustical Society of America*, 53(1):380–380, 1973. doi: 10.1121/1.1982702. URL `https://doi.org/10.1121/1.1982702`.

Sarah Levitan, Angel Maredia, and Julia Hirschberg. Acoustic-Prosodic Indicators of Deception and Trust in Interview Dialogues. In *Proc. Interspeech 2018*, pages 416–420, 2018. doi: 10.21437/Interspeech.2018-2443. URL `http://dx.doi.org/10.21437/Interspeech.2018-2443`.

Gina-Anne Levow. Context in Multi-lingual Tone and Pitch Accent Recognition. In *Proc. European Conference on Speech Communication and Technology*, pages 1809–1812, 01 2005.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proc. Association for Computational Linguistics (ACL)*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL `https://www.aclweb.org/anthology/2020.acl-main.703`.

Yang Liu, Andreas Stolcke, Elizabeth Shriberg, and Mary Harper. Comparing and Combining Generative and Posterior Probability Models: Some Advances in Sentence Boundary Detection in Speech. In *Proc. Conference on Empirical Methods in Natural Language*

Processing (EMNLP), pages 64–71, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W04-3209`.

Yang Liu, Kun Han, Zhao Tan, and Yun Lei. Using Context Information for Dialog Act Classification in DNN Framework. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2170–2178, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1231. URL `https://www.aclweb.org/anthology/D17-1231`.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv*, abs/1907.11692, 2019.

Margaret Lochrin, Joanne Arciuli, and Mridula Sharma. Assessing the Relationship Between Prosody and Reading Outcomes in Children Using the PEPS-C. *Scientific Studies of Reading*, 19(1):72–85, 2015. doi: 10.1080/10888438.2014.976341. URL `https://doi.org/10.1080/10888438.2014.976341`.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *Proc. International Conference on Learning Representations (ICLR)*, 2019. URL `https://openreview.net/forum?id=Bkg6RiCqY7`.

Iker Luengo, Eva Navas, Inmaculada Hernáez, and Jon Sánchez. Automatic Emotion Recognition Using Prosodic Parameters. In *Proc. Interspeech*, 2005.

Ranniery Maia, Tomoki Toda, Heiga Zen, Yoshihiko Nankaku, and Keiichi Tokuda. An Excitation Model for HMM-Based Speech Synthesis Based on Residual Modeling. *ISCA Workshop on Speech Synthesis*, 2007.

Mitchell Marcus, Beatrice Santorini, Mary Marcinkiewicz, and Ann Taylor. *Treebank-3 LDC99T42*. Linguistic Data Consortium, 1999.

Alex Marin and Mari Ostendorf. Domain Adaptation for Parsing in Automatic Speech Recognition. In *Proc. Acoustics, Speech, and Signal Processing (ICASSP)*, pages 6379–6383, 2014. doi: 10.1109/ICASSP.2014.6854832.

David Martinez, Lukáš Burget, Luciana Ferrer, and Nicolas Scheffer. IVector-based Prosodic System for Language Identification. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 03 2012. doi: 10.1109/ICASSP.2012.6289008.

David Martinez, Eduardo Lleida, Alfonso Ortega, and Antonio Miguel. Prosodic Features and Formant Modeling for an ivector-based language recognition system. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6847–6851, 2013. doi: 10.1109/ICASSP.2013.6638988.

Julie Medero and Mari Ostendorf. Atypical Prosodic Structure as an Indicator of Reading Level and Text Difficulty. In *Proc. Association for Computational Linguistics and Human Language Technology (ACL-HLT)*, pages 715–720. Association for Computational Linguistics, 2013. URL `http://aclweb.org/anthology/N13-1085`.

Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. Recurrent Neural Network Based Language Model. In *Proc. Interspeech*, 2010.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeff Dean. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems*, volume 26, pages 3111–3119. Curran Associates, Inc., 2013. URL `https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf`.

Nicolas Obin, Christophe Veaux, and Pierre Lanchantin. Making Sense of Variations: Introducing Alternatives in Speech Synthesis. In *Proc. Speech Prosody*, 2012.

Nicolas Obin, Julie Beliao, Christophe Veaux, and Anne Lacheret. SLAM: Automatic Stylization and Labelling of Speech Melody. In *Proc. Speech Prosody*, page 246, France, May 2014. URL `https://hal.archives-ouvertes.fr/hal-00997238`.

Sylvester Orimaye, Jojo Wong, and Karen Golden. Learning Predictive Linguistic Features for Alzheimer's Disease and Related Dementias Using Verbal Utterances. In *Proc. Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality (CLPsych)*. Association for Computational Linguistics, 2015.

Daniel Ortega and Ngoc Vu. Neural-based Context Representation Learning for Dialog Act Classification. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 247–252, Saarbrücken, Germany, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-5530. URL `https://www.aclweb.org/anthology/W17-5530`.

Daniel Ortega and Ngoc Vu. Lexico-Acoustic Neural-Based Models for Dialog Act Classification. *Proc. Acoustics, Speech and Signal Processing (ICASSP)*, pages 6194–6198, 2018.

Mari Ostendorf, Patti Price, and Stefanie Shattuck-Hufnagel. Boston University Radio Speech Corpus, LDC96S36. Web Download, 1996.

Shimei Pan. *Prosody Modeling in Concept-to-Speech Generation*. PhD thesis, Columbia University, 2002.

Shimei Pan and Kathleen McKeown. Integrating Language Generation with Speech Synthesis in a Concept to Speech System. *Proc. ACL/EACL Concept to Speech Workshop*, 1997.

John Pate and Sharon Goldwater. Unsupervised Dependency Parsing with Acoustic Cues. *Trans. of the Association for Computational Linguistics (ACL)*, 1:63–74, 2013.

Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global Vectors for Word Representation. In *Proc. Empirical Methods in Natural Language Processing (EMNLP)*, pages 1523–1543, 2014.

Varghese Peter, Genevieve McArthur, and Stephen Crain. Using Event-Related Potentials to Measure Phrase Boundary Perception in English. *BMC Neuroscience*, 15(1):129, Nov 2014. ISSN 1471-2202. doi: 10.1186/s12868-014-0129-z. URL https://doi.org/10.1186/s12868-014-0129-z.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep Contextualized Word Representations. In *Proc. North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 2227–2237. Association for Computational Linguistics, 2018.

Janet Pierrehumbert. *The Phonology and Phonetics of English Intonation*. PhD thesis, Massachusetts Institute of Technology, September 1980.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlíček, Yanmin Qian, Petr Schwarz, Jan Silovský, Georg Stemmer, and Karel Veselý. The Kaldi Speech Recognition Toolkit. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2011.

Daniel Povey, Vijayaditya Peddinti, Daniel Galvez, Pegah Ghahremani, Vimal Manohar, Xingyu Na, Yiming Wang, and Sanjeev Khudanpur. Purely Sequence-Trained Neural Networks for ASR Based on Lattice-Free MMI. In *Proc. Interspeech*, 2016.

Patti Price, Mari Ostendorf, Stefanie Shattuck-Hufnagel, and Cynthia Fong. The Use of Prosody in Syntactic Disambiguation. In *Proc. Workshop on Speech and Natural Language*, pages 2956–2970, 1991.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter Liu. Exploring the Limits of Transfer Learning with a

Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL `http://jmlr.org/papers/v21/20-074.html`.

Vipul Raheja and Joel Tetreault. Dialogue Act Classification with Context-Aware Self-Attention. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3727–3733, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1373. URL `https://www.aclweb.org/anthology/N19-1373`.

Nils Reimers and Iryna Gurevych. Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging. In *Proc. Spoken Language Processing (ICSLP)*, pages 338–348, 2017.

Eugénio Ribeiro, Ricardo Ribeiro, and David Matos. Deep Dialog Act Recognition using Multiple Token, Segment, and Context Information Representations. *Journal of Artificial Intelligence Research (JAIR)*, 66:861–899, 2019.

Brian Roark, Mary Harper, Eugene Charniak, Bonnie Dorr, Mark Johnson, Jeremy Kahn, Yang Liu, Mari Ostendorf, John Hale, Anna Krasnyanskaya, Matthew Lease, Izhak Shafran, Matthew Snover, Robin Stewart, and Lisa Yung. SParseval: Evaluation Metrics for Parsing Speech. In *Proc. Language Resources and Evaluation (LREC)*, pages 333–338, 2006.

Brian Roark, Margaret Mitchell, John-Paul Hosom, Kristy Hollingshead, and Jeffrey Kaye. Spoken Language Derived Measures for Detecting Mild Cognitive Impairment. *IEEE Transactions on Audio, Speech, and Language Processing (TASLP)*, 19(7):2081–2090, 2011. doi: 10.1109/TASL.2011.2112351.

Ina Roesiger, Sabrina Stehwien, Arndt Riester, and Ngoc Vu. Improving Coreference Resolution with Automatically Predicted Prosodic Information. In *Proc. Workshop on Speech-Centric Natural Language Processing*, pages 78–83, 2017.

Andrew Rosenberg. AutoBI - A Tool for Automatic ToBI Annotation. In *Proc. Interspeech*, 2010.

Andrew Rosenberg and Julia Hirschberg. Detecting Pitch Accents at the Word, Syllable and Vowel Level. In *Proceedings of Human Language Technologies: The Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 81–84, Boulder, Colorado, June 2009. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/N09-2021`.

Yikang Shen, Zhouhan Lin, Athul Jacob, Alessandro Sordoni, Aaron Courville, and Yoshua Bengio. Straight to the Tree: Constituency Parsing with Neural Syntactic Distance. In *Proc. Association for Computational Linguistics (ACL) (Volume 1: Long Papers)*, pages 1171–1180. Association for Computational Linguistics, 2018.

Elizabeth Shriberg and Andreas Stolcke. Direct Modeling of Prosody: An Overview of Applications in Automatic Speech Processing. Technical report, SRI International and International Computer Science Institute (ICSI), 2004.

Elizabeth Shriberg, Andreas Stolcke, Daniel Jurafsky, Noah Coccaro, Marie Meteer, Rebecca Bates, Paul Taylor, Klaus Ries, Rachel Martin, and Carol Ess-Dykema. Can Prosody Aid the Automatic Classification of Dialog Acts in Conversational Speech? *Language and Speech*, 41(3-4):443–492, 1998.

Kim Silverman, Mary Beckman, John Pitrelli, Mari Ostendorf, Colin Wightman, Patti Price, Janet Pierrehumbert, and Julia Hirschberg. ToBI: A Standard for Labeling English Prosody. In *Proc. Spoken Language Processing (ICSLP)*, pages 867–870, 1992.

RJ Skerry-Ryan, Eric Battenberg, Ying Xiao, Yuxuan Wang, Daisy Stanton, Joel Shor, Ron Weiss, Rob Clark, and Rif Saurous. Towards End-to-End Prosody Transfer for Expressive Speech Synthesis with Tacotron. In *Proc. Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, pages 4693–4702, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL `http://proceedings.mlr.press/v80/skerry-ryan18a.html`.

Jesse Snedeker and Elizabeth Casserly. Is It All Relative? Effects of Prosodic Boundaries on the Comprehension and Production of Attachment Ambiguities. In *Language and Cognitive Processes*, pages 1234–1264, 2010.

Sabrina Stehwien and Ngoc Vu. Prosodic Event Recognition Using Convolutional Neural Networks with Context Information. In *Proc. Interspeech*, pages 2326–2330, 2017. doi: 10.21437/Interspeech.2017-1159. URL `http://dx.doi.org/10.21437/Interspeech.2017-1159`.

Sabrina Stehwien, Ngoc Vu, and Antje Schweitzer. Effects of Word Embeddings on Neural Network-based Pitch Accent Detection. In *Proc. International Conference on Speech Prosody*, pages 719–723, 2018. doi: 10.21437/SpeechProsody.2018-146. URL `http://dx.doi.org/10.21437/SpeechProsody.2018-146`.

Mitchell Stern, Jacob Andreas, and Dan Klein. A Minimal Span-Based Neural Constituency Parser. In *Proc. Association for Computational Linguistics (ACL) (Volume 1: Long Papers)*, pages 818–827. Association for Computational Linguistics, 2017.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van, and Marie Meteer. Dialogue Act Modeling for Automatic Tagging and Recognition of Conversational Speech. *Computational Linguistics (COLING)*, 26(3):339–374, 2000. URL `https://www.aclweb.org/anthology/J00-3003`.

Paul Taylor. The Tilt Intonation Model. In *The 5th International Conference on Spoken Language Processing, Incorporating The 7th Australian International Speech Science and*

Technology Conference, Sydney Convention Centre, Sydney, Australia, 30th November - 4th December 1998, 1998. URL `http://www.isca-speech.org/archive/icslp_1998/i98_0827.html`.

Keiichi Tokuda, Yoshihiko Nankaku, Tomoki Toda, Heiga Zen, Junichi Yamagishi, and Keiichiro Oura. Speech Synthesis Based on Hidden Markov Models. *Proc. IEEE*, May 2013. ISSN 0018-9219. doi: 10.1109/JPROC.2013.2251852.

Trang Tran, Shubham Toshniwal, Mohit Bansal, Kevin Gimpel, Karen Livescu, and Mari Ostendorf. Parsing Speech: A Neural Approach to Integrating Lexical and Acoustic-Prosodic Information. In *Proc. North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Volume 1 (Long Papers)*, pages 69–81. Association for Computational Linguistics, June 2018. doi: 10.18653/v1/N18-1007. URL `https://www.aclweb.org/anthology/N18-1007`.

Trang Tran, Jiahong Yuan, Yang Liu, and Mari Ostendorf. On the Role of Style in Parsing Speech with Neural Models. In *Proc. Interspeech*, pages 4190–4194, 2019. doi: 10.21437/Interspeech.2019-3122. URL `http://dx.doi.org/10.21437/Interspeech.2019-3122`.

Gokhan Tür, Dilek Hakkani-Tür, Andreas Stolcke, and Elizabeth Shriberg. Integrating Prosodic and Lexical Cues for Automatic Topic Segmentation. *Computational Linguistics*, 27:31–57, 2001.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. WaveNet: A Generative Model for Raw Audio, 2016. URL `http://arxiv.org/abs/1609.03499`. cite arxiv:1609.03499.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is All You Need. In *Proc. Neural Information Processing Systems (NeurIPS)*, pages 5998–6008. Curran Associates, Inc., 2017.

Jennifer Venditti. *Discourse Structure and Attentional Salience Effects on Japanese Intonation*. PhD thesis, Ohio State University, 2000.

Klara Vicsi and Gyorgy Szaszak. Using Prosody to Improve Automatic Speech Recognition. *Speech Communication*, 52(5):413–426, 2010. ISSN 0167-6393. doi: https://doi.org/10.1016/j.specom.2010.01.003. URL `http://www.sciencedirect.com/science/article/pii/S0167639310000129`.

Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. Grammar as a Foreign Language. In *Proc. Neural Information Processing Systems (NeurIPS) - Volume 2*, pages 2773–2781, 2015.

Michael Wagner and Duane Watson. Experimental and Theoretical Advances in Prosody: A Review. *Language and Cognitive Processes*, 25:905–945, 09 2010. doi: 10.1080/01690961003589492.

Michael Wagner, Mara Breen, Edward Fleming, Stefanie Shattuck-Hufnagel, and Edward Gibson. Prosodic Effects of Discourse Salience and Association with Focus. In *Proc. Speech Prosody*, 2010.

Vincent Wan, Chun-An Chan, Tom Kenter, Rob Clark, and Jakub Vit. CHiVE: Varying Prosody in Speech Synthesis with a Linguistically Driven Dynamic Hierarchical Conditional Variational Network. In *Proc. International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pages 3331–3340, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

Sinong Wang, Belinda Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-Attention with Linear Complexity, 2020.

Yuxuan Wang, RJ Skerry-Ryan, Ying Xiao, Daisy Stanton, Joel Shor, Eric Battenberg, Rob Clark, and Rif Saurous. Uncovering Latent Style Factors for Expressive Speech

Synthesis. In *Workshop Machine Learning for Audio Signal Processing at NeurIPS (ML4Audio@NeurIPS17)*, 2017. URL `arXivpreprintarXiv:1711.00520`.

Nigel Ward, Jason Carlson, Olac Fuentes, Diego Castán, Elizabeth Shriberg, and Andreas Tsiartas. Inferring Stance from Prosody. In *Proc. Interspeech*, pages 1447–1451, 08 2017. doi: 10.21437/Interspeech.2017-159.

Nigel Ward, Jason Carlson, and Olac Fuentes. Inferring Stance in News Broadcasts from Prosodic-feature Configurations. *Computer Speech & Language*, 50:85–104, 2018. ISSN 0885-2308. doi: https://doi.org/10.1016/j.csl.2017.12.007. URL `http://www.sciencedirect.com/science/article/pii/S088523081730150X`.

Duane Watson and Edward Gibson. Intonational Phrasing and Constituency in Language Production and Comprehension*. *Studia Linguistica*, 59(2-3):279–300, 2005. doi: 10.1111/j.1467-9582.2005.00130.x. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9582.2005.00130.x`.

Colin Wightman and Mari Ostendorf. Automatic Labeling of Prosodic Patterns. *IEEE Transactions on Speech and Audio Processing*, 2(4):469–481, 1994. doi: 10.1109/89.326607.

Colin Wightman, Stefanie Shattuck-Hufnagel, Mari Ostendorf, and Patti Price. Segmental Durations in the Vicinity of Prosodic Phrase Boundaries. *The Journal of the Acoustical Society of America*, 91(3):1707–1717, 1992.

Yi Xu and Ching Xu. Phonetic Realization of Focus in English Declarative Intonation. *Journal of Phonetics*, 33(2):159–197, 2005. ISSN 0095-4470. doi: https://doi.org/10.1016/j.wocn.2004.11.001. URL `http://www.sciencedirect.com/science/article/pii/S0095447005000021`.

Maria Yancheva, Kathleen Fraser, and Frank Rudzicz. Using Linguistic Features Longitudinally to Predict Clinical Scores for Alzheimer's Disease and Related Dementias. In *Proc. Workshop on Speech and Language Processing for Assistive Technologies (SPLAT)*, 2015.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *NeurIPS*, 2019.

Masashi Yoshikawa, Hiroyuki Shindo, and Yuji Matsumoto. Joint Transition-Based Dependency Parsing and Disfluency Detection for Automatic Speech Recognition Texts. In *Proc. Empirical Methods in Natural Language Processing (EMNLP)*, page 1036–1041, 2016.

Jiahong Yuan and Mark Liberman. Speaker Identification on the SCOTUS Corpus. In *Proc. Acoustics*, 2008.

Vicky Zayats, Trang Tran, Richard Wright, Courtney Mansfield, and Mari Ostendorf. Disfluencies and Human Speech Transcription Errors. In *Proc. Interspeech*, pages 3088–3092, 2019. doi: 10.21437/Interspeech.2019-3134. URL `http://dx.doi.org/10.21437/Interspeech.2019-3134`.

Tianyu Zhao and Tatsuya Kawahara. Joint Dialog Act Segmentation and Recognition in Human Conversations Using Attention to Dialog Context. *Computer Speech & Language*, 57:108–127, 2019. ISSN 0885-2308. doi: https://doi.org/10.1016/j.csl.2019.03.001. URL `http://www.sciencedirect.com/science/article/pii/S0885230818304030`.

Appendix A
APPENDIX

A.1 Implementation

Constituency Parsing Most of the data preprocessing code is available at (1), part of our data preprocessing pipeline also uses (2). The implementation for the RNN-seq parsing models is available at (3). For the transformer-based models, the codebase is available at (4), which was adapted from (5).

(1) `https://github.com/trangham283/seq2seq_parser/tree/master/src/data_preps`

(2) `https://github.com/syllog1sm/swbd_tools`

(3) `https://github.com/shtoshni92/speech_parsing`

(4) `https://github.com/trangham283/prosody_nlp/tree/master/code/self_attn_speech_parser`

(5) `https://github.com/nikitakit/self-attentive-parser`

Dialog Act Recognition Both the RNN- and transformer-based models for DA recognition are available at (1), which is adapted from (2). Preprocessing steps were based on (3) and (4).

(1) `https://github.com/trangham283/joint_seg_da`

(2) `https://github.com/ZHAOTING/dialog-processing`

(3) `https://github.com/hao-cheng/dynamic_speaker_model`

(4) `https://github.com/cgpotts/swda`

Automatic Speech Recognition The experiments for parsing and DA recognition on imperfect transcripts can be found at (1). To use the Kaldi ASR system, some guidance can be found at (2).

(1) `https://github.com/trangham283/asr_preps`

(2) `https://github.com/trangham283/kaldi_examples`

A.2 Data Splits

Constituency Parsing Table A.1 shows statistics of our Switchboard dataset used in parsing experiments. The splits are: conversations sw2000 to sw3000 for training (train), sw4500 to sw4936 for validation (dev), and sw4000 to sw4153 for evaluation (test). In addition, previous work has reserved sw4154 to sw4500 for "future use" (dev2), but we added this set to our training set. That is, all of our models are trained on Switchboard conversations sw2000 to sw3000 as well as sw4154 to sw4500.

Table A.1: Data statistics in parsing experiments.

Split	# Conversations	# Sentences	# Tokens
train	541	97,113	729,252
dev	51	5,769	50,445
test	50	5,901	48,625

Dialog Act Recognition The train/dev/test splits for DA recognition tasks are not the same as those in parsing. The splits most commonly used in DA Classification work follow those defined in `https://web.stanford.edu/~jurafsky/ws97/`. Table A.2 shows statistics of this SWDA set for DA recognition experiments.

Table A.2: Data statistics in DA recognition experiments.

Split	# Conversations	# Turns	#Segments	# Tokens
train	1,115	97,367	193,805	1,525,112
dev	21	1,501	3,290	26,819
test	19	2,147	4,096	31,062

A.3 Pause Duration Statistics

Figure A.1 shows the distribution of pause durations in our training data. Our pause buckets described in Section 3.2.1 were based on this distribution of pause lengths.

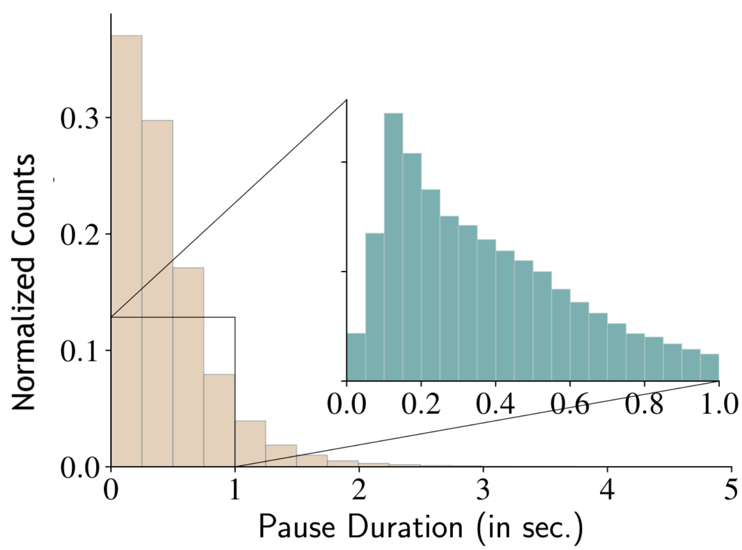


Figure A.1: Histogram of inter-word pause durations in our training set. As expected, most of the pauses are less than 1 second. In some very rare cases, pauses of 5+ seconds occur within a sentence.