THE UNIVERSITY *of* EDINBURGH

Effects of prosody on natural language processing

Elizabeth Nielsen

Doctor of Philosophy

Institute for Language, Cognition and Computation
School of Informatics
University of Edinburgh

2024

Abstract

Prosody — or the systematic variation in the energy, pitch, timing, and voice quality of speech — plays an important role in speech communication. For example, pitch is the primary way an English speaker can distinguish between certain kinds of questions and statements (e.g., *That's today?* vs. *That's today.*). Despite the fact that prosody can convey a range of linguistic features, it is uncommon for NLP systems that deal with speech inputs to give consideration to prosodic features. Many systems such as dialog agents start with an automatic speech recognition (ASR) step, which converts the audio signal into text, after which all prosodic information is discarded. Previous research has established that prosody can be helpful — it has been shown to aid in tasks such as syntactic parsing (Tran et al., 2018) — but the amount of benefit shown for many tasks is modest enough that including prosodic inputs still remains a niche approach in NLP.

The goal of this thesis is to revisit the question of how prosodic features can benefit a range of NLP tasks. First, Chapter 3 considers the question of what modeling choices are best for incorporating prosodic inputs to NLP tasks. These experiments show that a wide input context is helpful in detecting prosodic information, but even so, text features alone are able to predict a relatively large portion of prosodic activity. Second, Chapter 4 showcases an example where prosody has no observed effect. Even though there is good linguistic justification for expecting that prosody should help in better conveying information status in speech translation, this effect is not seen because the biases of the speech translation model itself make any effect unmeasureable, underscoring the importance of task and model selection. Third, Chapter 5 shows that prosody does help with syntactic parsing in the more realistic setting where the input is not pre-segmented into sentences. In fact, prosody helps more with segmenting the speech into sentences than with parsing itself, but both tasks benefit. These experiments show that the realistic task of parsing plus segmentation benefits in more ways from including prosody than does parsing alone. Finally, Chapter 6 considers what happens in the sentence segmentation task when an ASR transcript is used as the lexical input, and acoustic noise is introduced to the audio signal. As more sources of noise are added, prosody becomes progressively more important for the model's performance. This suggests that the information in the prosodic and lexical channels is somewhat redundant, with the prosodic channel acting more as a 'back-up' for the lexical channel than as a channel for novel information.

Together, these results suggest that prosody has the potential to be helpful in many NLP tasks, but that these benefits are more marked in cases that better approximate

real-world language usage, where there are obstacles to clear communication. Because the information in the prosodic and lexical channels overlaps so much, adding prosodic information does not boost performance as much when both channels are clear and unobstructed. However, when obstacles to clear perception (such as lacking sentence boundaries, using an ASR transcript, or acoustic noise) are present, prosody becomes more important. This suggests that in future work, it will be important to move towards modelling assumptions that better approximate the non-idealized conditions of real-world language use in order to fully understand the value of prosody for NLP tasks.

Lay Summary

There's a lot more to the way we speak than just words. We can raise our voice to indicate a question (*He does?*), or use an even more dramatic dip and raise in pitch to show incredulity (*He does???*). We can convey sarcasm (*Oh, I'm **sure***) or clarification (e.g., *No, I mean **those** ones*), all by just varying the pitch and volume of the voice, and by adding pauses and lengthening or shortening words.

These variations in the voice beyond the words are collectively known as *prosody*. Despite the major role that prosody plays in how we speak to each other, most of the time, computer-based applications don't make use of it. For example, speech-based digital assistants tend to transcribe the user's speech into text and then discard the rest of the audio information. Previous researchers have shown that prosody can help in a variety of speech processing tasks, but generally, the amount of benefit from prosody isn't huge, and so its use isn't widespread.

The goal of this thesis is to reconsider the question of whether prosody can be useful for computer models that are tasked with processing human speech. The first experiment establishes some best practices for how to handle prosodic inputs when we introduce them to language processing models. Subsequent experiments test whether prosody is helpful in a handful of tasks. In one case — helping a speech translation model capture some subtleties of word ordering in Russian — it isn't helpful. However, the other experiments, which concern splitting a speech input into sentences and then analyzing its grammatical structure, show modest benefits from including prosody. Additionally, we find that prosody proves more helpful when we add more background noise to the audio.

These experiments support the conclusion that in practice, prosody isn't usually the carrier of novel information, but instead acts as more of a 'back-up' channel to the words themselves. That's likely why, in an idealized setting, we see relatively small benefits from adding prosody to a task, since the information it carries is mostly already present in the text. However, as we move towards settings that are noisier and make the words harder to understand, prosody is more helpful. This suggests that as we ask our speech processing models to handle more realistic and noisy speech inputs, they may see more and more benefit from being able to learn from prosody.

Acknowledgements

I have many people to thank who have helped me throughout this PhD. Top of the list are Sharon Goldwater and Mark Steedman, who have been extraordinary supervisors. I am so grateful to have had four years of talking through research ideas with them, getting their input on designing experiments, and benefitting some of the most insightful feedback on writing that I've ever received. I'm especially grateful to them for being kind and empathetic people, in addition to skilled mentors. I truly won the supervisor lottery.

I'm grateful to other mentors I've had throughout this process. Thank you as well to Catherine Lai, for all of the thoughtful and insightful conversations on prosody. Thank you to Mari Ostendorf and Trang Tran, for discussing their work in this area with me and helping me formulate my research ideas. I'm also very grateful to my examiners, Gina-Anne Levow and Simon King, for all their effort in reading and giving me feedback, and for an interesting and enjoyable viva. I'm especially thankful to Gina for coming all the way to Scotland for an in-person viva.

I've also been very lucky to have mentors who helped me throughout my internships at Google. Thank you to Brian Roark, Christo Kirov, Jiaming Luo, George Foster, and Colin Cherry. I'm grateful I landed in internship teams with such kind, insightful people.

I'm also grateful to my fellow students (and postdocs) at Edinburgh, for encouragement over cups of tea in the Forum, for talking over and improving my ideas, for the occasional run or hike around Edinburgh, and generally for banding together, both virtually and outdoors (in some really questionable Edinburgh weather), during the novel experience of living overseas during a pandemic. I'm especially grateful to Nick McKenna, Ida Szubert, Sameer Bansal, Yevgen Matusevych, Coleman Haley, Yumnah Mohamied, Sarenne Wallbridge, Nickolay Bogoychev, and Ramon Sanabria.

I'm indebted to the people outside the university who supported me in Edinburgh, most of all Brontë and the rest of the Waddoups family, who took me in and became a surrogate family. Thank you as well to John and Linda Harmer, and all the other members of the Edinburgh Ward who have been such an important community for me.

I'm also fortunate to have had friends outside of Edinburgh who kept in contact with me from afar, especially Jacquelyn, Catie, and Marissa. Surviving a pandemic would not have been possible without regular talks on the phone while I wandered around Edinburgh. Thank you for dealing with the time difference and the persistently windy audio on my end.

I'm also lucky to have been supported by a wonderful family. Thank you to Sarah and Jake, for letting me hang out with their children when I desperately needed a break from working. Thank you to David for being a non-judgmental sounding board for all kinds of NLP, engineering, and life questions. Thank you to Peter for always having a kind word to say. Thank you to my parents, Richard and Kaylene, for more things than I can enumerate: for letting me call them at bizarre hours to talk over research and anything else I happened to have on my mind; for visiting me in Edinburgh and letting me show them all my favorite moss; and most of all, for always being excited for me to pursue my educational goals, wherever they took me.

Finally, thank you to Anthony, for everything.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Elizabeth Nielsen*)

To Anthony.

Table of Contents

Chapter 1

Introduction

Spoken language is characterized by all kinds of prosodic variation — that is, variation in the pitch and energy of the voice, the timing of words and syllables, or the quality of the voice.[1] For example, the following sentence could be pronounced with a marked rise in energy and pitch on the bolded word:

(1) The concert is **tomorrow**.

This pitch and energy variation could be caused by non-linguistic factors. For example, if someone nearby turned on a lawnmower during the final word of Example (1), the speaker might raise the energy (colloquially, the volume) of their voice to be heard over the noise. This is variation in an acoustic correlate of prosody — energy — but it has no linguistic meaning. This thesis is largely concerned with *linguistic prosody* — variations in acoustic correlates of prosody that signal some kind of linguistic information. For example, if the speaker was answering the question, *Is the concert today?*, the raised energy and pitch in Example (1) would carry important information about the contrast between the two possible concert times. This is an example of linguistic prosody.

Prosody can carry an array of different types of linguistic information, and functions differently from language to language. In English, prosody can serve numerous functions, including marking contrast, as in the above example. Another use of prosody in English is distinguishing questions (*We do?*) from statements (*We do.*) with different pitch and energy patterns. There is abundant non-computational linguistic research indicating that that English prosody carries various other kinds of linguistic information (see e.g., Cutler et al. (1997)). Some types of linguistic information dis-

[1] For more details on what is meant by voice quality, see Section 2.1.1.

cussed in this thesis include signals for sentence boundaries, speech disfluencies, and information status (i.e., the givenness or newness of discourse elements).

However, previous attempts to make use of this linguistic information in computational models have yielded equivocal results. This is illustrated by research into prosody and syntactic parsing. Gregory et al. (2004) found that incorporating prosodic annotations into a PCFG parser actually worsened performance. Later researchers found ways to show benefits from prosodic annotations in non-neural parsers (e.g., Huang and Harper (2010); Kahn et al. (2004)), but the overall performance of these parsers has since been outdone by current state-of-the-art neural parsers. With higher performing neural parsers, such as the models of Tran et al. (2018) and Tran et al. (2019), the effect sizes from including prosody are quite small, and seem to derive largely from the ways in which prosody helps with disfluency handling.

So, despite the ample evidence that prosody carries many types of linguistic information, previous work has not shown clear benefits to using prosodic inputs in the NLP applications that have been tried so far. One possible reason for this limited benefit is the encoding of the prosodic information: Many pre-neural studies had to incorporate prosodic information in the form of word-level acoustic correlates of prosody like pitch trajectory over a word (e.g., Gregory et al. (2004)), which involves a loss of granular acoustic information. In other cases, the task itself may simply be one for which prosody is of limited benefit. For example, studies of including prosody in syntactic parsing tend to show small effects (e.g., Tran et al. (2019)), or occasionally, no benefit Gregory et al. (2004). This is likely because only some syntactic boundaries are signaled by prosodic cues, as discussed in Chapter 5. In fact, in Tran et al. (2018) and Tran et al. (2019), most of the benefit seems to boil down to helping with handling speech disfluencies, which is a task prosody actually helps with a great deal compared to parsing (Zayats and Ostendorf, 2019). In general, it seems that prosody is more helpful in tasks that better approximate the noisier setting of real speech usage (e.g., by including speech disfluencies), as opposed to assuming well-edited inputs.

The primary goal of this thesis is to reconsider the question of how the information carried in prosody can be used in NLP models. The approaches taken differ from previous models in one of three ways:

Choice of inputs. In this thesis, prosodic inputs are generally less processed. For example, two important acoustic correlates of prosody — fundamental frequency (or pitch[2]) and energy — are extracted directly from the audio recording. Neural models

[2]Note that in technical usage, pitch is more precisely the *perception* of the fundamental frequency

are important to this choice, since they are capable of processing acoustic signals that are closer to their original continuous form, where more traditional models might struggle to use features that aren't word-level. Using these types of less-processed acoustic features in neural models is not original to this thesis: This is inspired by work such as Tran et al. (2018, 2019) and Stehwien et al. (2018), though this thesis differs from these works in architectural choices and/or task design. These less-processed acoustic signals have many advantages over human-generated prosodic annotations: First, and most importantly, this choice allows us to use data that doesn't have prosodic annotations. This makes it possible to use much larger corpora than would otherwise be possible, which opens up many research avenues that would otherwise be stymied by a lack of data. Another reason to prefer acoustic features is that prosodic annotations generally come in the form of ToBI annotations, which incorporate a theoretical account of prosodic structure. Since the goal of this work is to see the effect of the information from the prosodic signal, using ToBI annotations as input complicates this picture by imposing structure on the input.

Choice of task. This thesis focuses on NLP tasks for which there is more robust linguistic evidence that prosody could help, including sentence segmentation (see Chapters 5 and 6) and tasks that rely on detecting information structure (see Chapter 4). With the exception of the work covered in Chapter 3, the models used are generally not directly detecting either prosodic events or the linguistic features that the acoustic correlates of prosody are posited to convey (e.g., information structure or speech disfluencies). Instead, these are models for more general-purpose NLP tasks where there is theoretical reason to think that prosody would have a measurable effect (e.g., a speech translation model in Chapter 4, or a syntactic parser in Chapter 5). Outcomes are measured by how the overall performance of the model on the target task changes in response to prosodic inputs, as well as other changes in the output due to the prosodic input. For example, in the speech translation case, word order variations in the output that may indicate sensitivity to prosodic variation in the input, and in the parsing case, the way in which prosody helps the parser handle speech disfluencies.

Some of the applications tried show a sizeable benefit from using prosodic inputs, particularly sentence segmentation. Other tasks show less benefit, including improving word order in speech translation.

Adding noise. Many of the experiments in this thesis suggest that one reason

(F0), rather than F0 itself. However, in this thesis, the word *pitch* is used to refer to F0, rather than perceived F0, in keeping with its colloquial usage.

that prosody provides only limited benefit is that the information in prosody is often redundant with lexical information. In order to test this hypothesis, I experiment with cases where the information from the lexical and prosodic channels is limited, for example by using ASR transcripts instead of human generated transcripts in Chapter 6, and by adding acoustic noise to the audio input to the task. The motivating hypothesis for these experiments is that in these noisier cases, prosodic information may be more accessible than lexical information, leading to a larger effect from prosody. The results in Chapter 6 generally support this hypothesis.

1.1 Thesis outline

This thesis is organized as follows:

Chapter 2 is devoted to background on related research. This includes more theoretical work from the linguistics world on prosody, as well as applied computational linguistics and NLP research on how prosody can be used in various applications.

Chapter 3 includes the content of the paper "The role of context in neural pitch accent detection in English," published in EMNLP 2020 with Mark Steedman and Sharon Goldwater. This chapter describes experiments aimed at isolating the effect of how prosodic inputs are handled. In order to abstract away from questions of what linguistic information is carried by prosody, this experiment focuses instead on the task of detecting prosodic events themselves. Specifically, a neural model is trained on the task of pitch accent detection. Overall, the results of this experiment show that prosodic inputs are more valuable when they include more of the input context. There is also find a great deal of redundancy between the lexical and prosodic channels: A model with access to text inputs only can still predict the prosodic phenomenon of pitch accent location with relatively high accuracy. This chapter lays the groundwork for future chapters in demonstrating how to effectively incorporate prosodic inputs into NLP models. It also establishes the overlap in information between text and prosody that is investigated in greater detail in Chapter 6.

The experiments in **Chapter 4** measure whether prosodic indicators of information status can be used in speech translation (ST). The hypothesis is that in an English to Russian speech-to-text translation system, the prosody of the source will affect the model behavior. In particular, English tends to mark new information prosodically, while Russian tends to mark this same information with word order. This experiment looks for evidence that variations in the source English prosody cause corresponding

variation in the target Russian word order. However, the results of this experiment show that the ST model tested tends to so thoroughly replicate the source word order that no effect can be measured for the influence of English prosody. These results suggest a compelling linguistic argument for why prosody isn't as helpful as expected in practice: The phenomenon must be robustly represented in the training data, and it must be possible to detect the effect in the model output. This chapter does not include content from any published papers.

Chapter 5 includes the content of the paper "Parsing dialog turns with prosodic features in English," which was published in INTERSPEECH 2023 with Mark Steedman and Sharon Goldwater. This chapter investigates how prosody can help in a common NLU task: syntactic parsing. Previous work has shown that prosody can help with parsing single sentences, though its effect is quite small (Tran et al., 2018, 2019). One reason for this small effect is that prosody is a relatively weak signal for syntax (see Section 2.2.2 in the following chapter). For that reason, the task used here is parsing speech that has not been divided into sentences.[3] The ends of sentences are consistently the sites for multiple types of prosodic cues (longer pauses, final word lengthening, and pitch events (Shriberg, 2001)), and so it seems likely that prosody could be a crucial signal for this task in particular, in a way that it isn't for general syntactic parsing.

Another reason for choosing this task is that it represents a step towards more realistic language processing: In a deployed setting, a model such as a dialog agent would not have access to sentence boundary information, nor do humans have this kind of information in conversation. This allows us to see if previous experiments obscured some of the benefits provided by prosody by their choice of simplifying assumptions.

The study outlined in this chapter shows that prosody helps with parsing, though the effect size is not substantially larger than the one found by Tran et al. (2018). However, when looking at the model's performance on sentence segmentation alone, there is a much larger performance gain due to prosody. This points to an interesting observation: The ability of a model to do sentence segmentation and its ability to parse are not necessarily strongly correlated. These results do show that, at least for the sample of spontaneous dialog represented in our corpus, the pattern we predicted holds true: Namely, the prosodic signal carries substantial, non-redundant information about sentence boundaries. Other types of syntactic information, including the boundaries of

[3]Note that throughout this thesis, the term 'sentence' is used for simplicity's sake. In speech, not all utterances are syntactically complete sentences. The units referred to here as 'sentences' are equivalent to Meteer and Taylor (1995)'s 'slash-units.' This distinction is explained in greater detail in Chapter 5.

constituents below the level of a sentence, are not present in the prosodic signal in a way that consistently helps beyond the text signal.

In **Chapter 6**, the sentence segmentation task is moved one step closer to a realistic setting by replacing the human-generated gold standard text with ASR-generated transcripts. Using ASR transcripts makes the information in the text signal harder to access without entirely removing it, and so functions as a kind of natural ablation of the text signal. In addition, adding Gaussian noise to the source audio at varying levels is a way of simultaneously ablating both the text and the prosody. Since evaluating parsing of ASR text is relatively complex and not directly comparable with parsing scores for gold transcripts, this chapter focuses only on sentence segmentation.

These experiments show that when the linguistic signal from text is degraded, the information carried by the prosodic signal plays a more important role. When segmenting ASR transcripts into sentences, performance goes down both with and without prosody. However, the performance of the model with prosody drops much less, relative to the text-only baseline, as more noise is added to the input signal. This supports the hypothesis that prosody is more important in noisier conditions. This chapter does not include any work that has been published yet.

1.1.1 Motivation for choice of tasks

The tasks in this thesis were selected for a variety of reasons. First, the pitch accent detection task in Chapter 3 sheds light directly on the question of how much one kind of prosodic event (pitch accents) can be predicted from the lexical signal alone. Also, starting with directly detecting prosodic events allows us to bracket the question of what linguistic information is signaled by prosody, and focus on the interactions of the lexical and prosodic signals.

Second, the experiments with information status in speech translation in Chapter 4 are an attempt to devise an evaluation that is independent of having human-labeled output categories. While the phenomenon being studied — information status — is complex and difficult to annotate, the proposed evaluation — measuring effects in the target word order — should theoretically work without annotated data. However, this method didn't yield informative results because of the biases of the model used.

Third, the combined parsing and sentence segmentation task in Chapter 5 is proposed because omitting human-annotated sentence boundaries from model input makes the task more realistic. Additionally, sentence boundaries are more likely to be sig-

naled prosodically than other syntactic boundaries, suggesting that prosody should be especially helpful here.

Finally, the task of sentence segmentation with noise in Chapter 6 is designed to look at the question of how prosody helps when the lexical signal is obscured by noise. The motivation for this task includes work showing that prosody is often more robust to noise than lexical information (van Zyl and Hanekom, 2011), suggesting that it may function partly as a back-up channel in cases where the lexical signal is less accessible.

1.2 Contributions

The primary contributions of this thesis are as follows:

- Demonstrating that it is possible to improve neural models that directly incorporate acoustic correlates of prosody by including wider context windows for the acoustic data (Chapter 3).

- Demonstrating that one important part of the prosodic signal — pitch accent location — can be predicted relatively well from the text alone (Chapter 3).

- Showing that some tasks that have good linguistic evidence suggesting that prosody should be helpful don't show much benefit from incorporating prosody, simply because of dataset and/or modeling constraints. This is shown to be the case in the work on improving word order in speech translation in Chapter 4.

- Demonstrating how prosody affects the combined tasks of sentence segmentation and parsing, which is a more realistic task than parsing already-segmented sentences. Prosody is shown to be very helpful in sentence segmentation, but still provides only limited benefit for parsing (Chapter 5).

- Showing the effect of noise on how prosody interacts with model performance. This is done by testing how ASR-generated output affects sentence segmentation, and how the benefit from prosody increases as noise is added to the audio. There is a correspondence between the amount of noise present in the inputs and the amount of benefit that prosody provides to a downstream task, namely sentence segmentation (Chapter 6).

1.3 A note on language

This thesis focuses primarily on English prosody and NLP, with some use of Russian in Chapter 4. This limitation comes from the fact that most of the experiments in this thesis require speech corpora that have been human-annotated for syntactic categories (e.g., Switchboard-NXT; Calhoun et al. (2010)) or prosodic categories (e.g., BURNC; Ostendorf et al. (1995)), and unfortunately, large corpora meeting these criteria don't seem to exist in other languages. Additionally, a large portion of the linguistic description and theory concerning prosody is focused on English. Unfortunately, relatively few languages even have a well-developed prosodic analysis or annotation system available (Jun, 2005).

Chapter 2

Background

This chapter describes previous research on prosody. Section 2.1 gives a basic definition of prosody and explains important concepts for talking about prosody. Section 2.2 is devoted to research into what linguistic information is present in the prosodic signal, and finally Section 2.3 covers previous experiments with incorporating prosody into NLP tasks. As with most of the further chapters, this section focuses on English language phenomena and data. This is because the annotated resources needed are available in English, and also because much of the linguistics research on prosody focuses on English.

2.1 Defining prosody

For the purposes of this thesis, prosody is linguistically motivated, systematic variation in the pitch, energy (or intensity), timing, or voice quality of the speech signal. In this section, I break down each part of this definition.

First, prosody is **linguistically motivated** and **systematic**. There are lots of reasons that someone might vary (for example) the pitch or energy of their voice. Recall the example in Chapter 1, where a speaker raised the energy of their voice to be heard over the noise of a lawnmower. This is certainly variation in one of the acoustic correlates of prosody, but its motivation is not linguistic, so for present purposes, this doesn't fall under the relevant definition of prosody.

The example in (1) shows how this same acoustic feature — energy — could be used in a linguistically motivated way. Imagine that the speaker is talking to their friend in an art museum, and their friend has mistaken which painting the speaker is referring to. Again, the speaker increases the energy (and also, likely, the pitch

and duration) of the bolded word, but in this case, it is to convey something within the linguistic domain: what is being referred to, and how this entity relates to other entities in the conversation.

(1) I didn't mean **that** painting. I meant the one below it.

This variation is also **systematic**: In lab experiments, speakers of English can consistently interpret comparable prosodic patterns in the same way, and can uncover rules that associate linguistic features to prosodic patterns (see e.g., Price et al. (1991); Wightman et al. (1991)).

2.1.1 Phonetics

The definition of prosody given above can be broken down into the various acoustic correlates of prosody within the speech signal. First, **pitch** is the perception of the fundamental frequency (F0) of the speech signal. As noted in Chapter 1, in this thesis, *pitch* is also used to refer to the F0 itself, though in technical usage it refers properly to the perception of F0. Second, the **energy** of the speech signal corresponds roughly to the concept of volume in common usage. **Timing** is a broad term that refers to the relative length of words and pauses, as well as the rhythmic properties of speech. Finally, **voice quality** refers to the way in which the speaker's vocal folds vibrate when producing speech. The most important distinction in voice quality for English prosody is the difference between *modal* and *creaky* voice. *Modal voice* is the default voice quality in English, in which the vocal folds vibrate at a regular rate, producing regular periodic sound waves. Holding the vocal folds under greater tension produces *creaky voice*, where the acoustic waves are aperiodic. This is sometimes popularly called vocal fry.

2.1.2 Phonology

These phonetic properties of speech are used in systematic ways to produce meaningful prosodic patterns. For example, in Example (1), the speaker may raise both the pitch and energy of their voice on the contrastive word *that* to convey their meaning. In order to capture the connections between patterns of acoustic correlates of prosody and linguistic meanings, linguists often use categories that abstract away from the acoustic features themselves. A few of these relevant terms are presented here that are helpful

in describing the prosodic patterns investigated in later chapters. These terms come from the Tones and Break Indicies (ToBI) system of prosodic annotation, which is currently a widespread way of categorizing prosodic phenomena (Pierrehumbert, 1980). Readers familiar with ToBI will note that this section doesn't include all the prosodic structures described in ToBI, but is limited to the categories relevant to this thesis.

First, speech can be divided into *prosodic phrases*. The largest type of prosodic phrase, called an intonational phrase in ToBI, is at most the size of a sentence, with smaller prosodic phrases posited to exist as constituents of this top-level prosodic phrase (Jun, 2005). These prosodic phrases can be marked by a few different features in English. First, prosodic phrases can be marked by pauses at their ends. Larger prosodic phrases are generally followed by longer pauses — that is, the pause at the end of an intonational phrase (say, one corresponding to a single sentence) is longer than the pauses inside that phrase.

Second, the ends of phrases can be marked by *boundary tones*. In general, a *tone* is a distinct rise or fall in the pitch of the voice. A tone can also be accompanied by changes in the energy of the voice, lengthening of the words that carry the tone, or changes in voice quality (e.g., creaky voice). Boundary tones fall at the end of prosodic phrases and help to mark them along with pauses.

In addition to phrases, *prominences* are an important prosodic structure in the ToBI system. A prominence is a segment of speech (generally a word or syllable) that has been marked as being more important in some way than other portions of the same prosodic unit. In English, each phrase generally has one or more prominences marked by *pitch accents*. Pitch accents use the same basic acoustic machinery as boundary tones (pitch excursions, changes in energy, and occasionally changes in voice quality), but apply these features in different combinations. Pitch accents can signal various linguistic categories, as discussed in Section 2.2 below.

2.2 Theory: Linguistic information in the prosodic signal

This thesis focuses on making use of the linguistic information in the prosodic signal. It is motivated by linguistic research showing that various kinds of information can be conveyed prosodically. Though many connections between prosody and linguistic structure exist, this section focuses solely on those that are relevant to later

chapters. Specifically, it covers how prosody has been posited to convey information status (Section 2.2.1), syntax (Section 2.2.2), disfluencies (Section 2.2.3), and the rate of information being communicated (Section 2.2.4). Finally, Section 2.2.5 is dedicated to the ways in which the information carried by the prosodic signal is robust to noise.

2.2.1 Prosody and information status

Information status is the relationship of entities to the discourse context. There are many different theoretical accounts of information status, and consequently many different sets of terminology for describing information status (e.g., Allerton (1978); Prince (1981, 1992); Steedman (2000)). For the purposes of this thesis, the most important distinction in information status is between *new* information and *old* (or *given*) information, as used by Calhoun et al. (2010).[1] Any entity that has been mentioned previously in a discourse is old information, while any entity not previously mentioned is new information. In Example (2) below, the first mention of *Sandy Thompson* is an example of new information, and the second is an instance of old information:

(2) I'm waiting for it to be noon so I can call *Sandy Thompson*.
Why are you trying to get in touch with *Sandy Thompson*? (Prince, 1992)

Information status is prosodically marked in English (Steedman, 2000). Just as there are many accounts of information status, there are also many accounts of how information status governs prosody in English. However, all these accounts hold that new information is more likely to be marked by a pitch accent. Not every pitch accented word necessarily carries new information, since pitch accent is governed by a number of considerations, including information status, syntactic structure, and the metrical properties of a phrase. The final content word of a sentence is especially likely to carry a pitch accent in the default case, when no other rules apply that would move it elsewhere. But in cases where pitch accents don't occur in the default location, they can often indicate new information.

As discussed in Chapter 4, information status can be marked non-prosodically, especially in languages other than English. The most relevant example here is Russian, which has relatively unconstrained word order, and so uses sentence-final position to signal new information (Rodionova, 2001).

[1] Calhoun et al. (2010) also distinguish a third category, *mediated* information, which is information that can be assumed given the discourse context, but the basic old/new distinction is the most important one here.

2.2.2 Prosody and syntax

Syntax concerns how words are combined into phrases and utterances. There are many differing theoretical accounts of syntax, but due to the limited availability of resources that are annotated with other syntactic formalisms, this thesis focuses on a traditional constituency approach to syntax, as used by resources such as the Penn Treebank (Marcus et al., 1993), or Switchboard-NXT (Calhoun et al., 2010).

Since both syntax and prosody involve grouping word words into phrases, there has been lots of research into the relationship between prosodic phrases and syntactic phrases. One location where prosodic and syntactic boundaries consistently line up is at the ends of sentences, where the ends of sentences and the ends of intonational phrases almost always coincide (Cutler et al., 1997; Kahn et al., 2005). However, prosodic phrases at levels below the sentence or intonational phrase do not always line up with traditionally defined syntactic phrases. Example (3) from Steedman (2000) illustrates this conflict. Phrase boundaries can fall between any of these words, with potential boundaries marked by the subscript numbers.

(3) $_1$ Marcel $_2$ proved $_3$ completeness $_4$

A traditional account of syntax would require a phrase boundary at 2, dividing the sentence into an NP (*Marcel*) and a VP (*proved completeness*). However, the prosodic phrasing is more variable. Sometimes the prosodic boundary *would* fall at *2*, such as when this sentence answers the question *Who proved completeness?*. But if this sentence is an answer to the question *What did Marcel prove?*, it's likely that there would be a prosodic boundary at 3. In this case, the prosodic phrase would conflict with the traditional syntactic phrases. One way to resolve this conflict is to posit that syntactic boundaries can also vary in the same way that prosodic boundaries do, as in Steedman (2000) and the broader Combinatory Categorial Grammar approach. However, all currently existing speech corpora that have syntactic annotations use a traditional constituency or dependency syntax approach, so for the purposes of this thesis, there is not a lot of experimentation that can be done with non-traditional approaches like CCG.

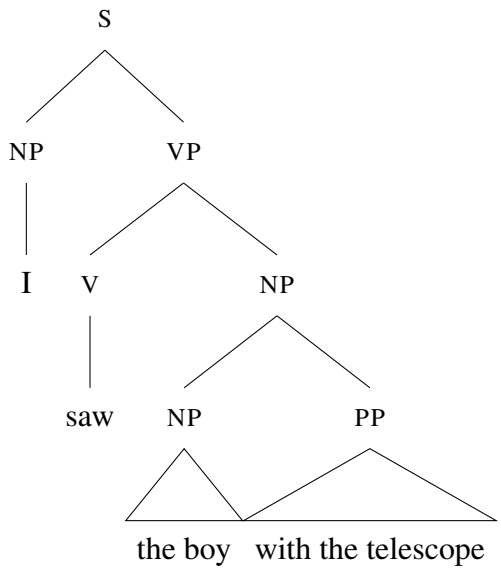
One particular syntactic phenomenon where prosody has been suggested as an important information source is ambiguous phrase attachment. This can be seen in sentences such as (4):

(4) I saw the boy with the telescope. (Ghaly and Mandel, 2017)

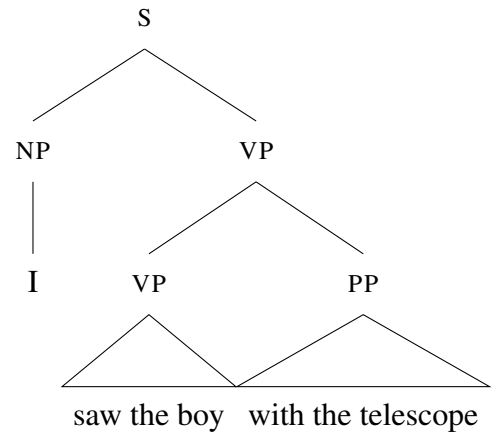
This sentence can be interpreted at least two ways:

(5) a. I saw the boy who had a telescope.
b. I used a telescope to see the boy.

The meaning in (5-a) corresponds to the syntactic analysis shown in Figure 2.1a where the PP *with the telescope* attaches to and modifies the NP *the boy*. (5-b) is produced by the analysis shown in Figure 2.1b, where the PP attaches to the VP *saw the boy*.



(a) Meaning: 'I saw the boy who had a telescope.'



(b) Meaning: 'I used a telescope to see the boy.'

Figure 2.1: Two possible parses for the sentence *I saw the boy with the telescope*, each with different meanings.

Lehiste (1973) demonstrated that when speakers are prompted to produce these kinds of ambiguous sentences, they use different prosodic cues for each of the different meanings. Furthermore, Price et al. (1991) and Wightman et al. (1991) showed that when playing this kind of ambiguous sentence back, these distinctions can be perceived both by human speakers and computational models.

However, in their review of the intervening literature, Cutler et al. (1997) found that these results were "far from robust and determinate" (169). They suggest that kind of ambiguity shown in Example (4) was likely to be relatively rare in spontaneous speech, and so the role of prosody was harder to detect outside of somewhat artificial settings.

On the whole, the one kind of syntactic structure that seems to be robustly signaled by prosody is the sentence boundary. Other prosodic boundaries can be informative given the right experimental conditions, but do not consistently coincide with syntactic structures in spontaneous speech.

2.2.3 Prosody and disfluencies

Spontaneous speech is a complex type of data, partly because of the prevalence of incidents of misspeaking, often called *disfluencies*. Any NLP model that starts with spontaneous speech needs to be able to handle these disfluencies.

Disfluencies can be broken down into several parts, as shown in (6) below:

(6) any health cover– any health insurance Shriberg (2001)

The first part of this utterance (*any health cover–*) is called the reparandum, and the second part (*any health insurance*) is called the repair. The point at which the speaker stops after *cover–* is called the interruption point.

The reparandum and the interruption point are both prosodically marked. For example, the reparandum is usually either slower or faster than normal speech, and the interruption point is characterized by a long pause (Shriberg, 2001).

Disfluencies play an important but indirect role in this thesis and other related research: When measuring the benefit of adding prosody to an NLP task, some of the effect may be attributable to how much prosody helps with handling disfluencies. For example, Tran et al. (2018) find that their parser gets most benefit from prosody when parsing disfluent sentences.

2.2.4 Prosody and information density

One other connection between the prosodic and linguistic signals comes by way of the Uniform Information Density (UID) hypothesis. While previous sections of this chapter have focused on linguistic approaches to prosody, the UID is phrased in terms that come from the realm of information theory and psycholinguistics. Briefly, the UID posits that the amount of information transmitted by a speaker tends to be relatively constant (Levy and Jaeger, 2006). In information-theoretic terms, each word contributes a certain amount of information, which is often measured in terms of *surprisal*, which reflects how probable a given word is given the preceding context. If the UID holds, then it would predict that if a word is less expected (i.e., high surprisal), it will take more time in the signal, keeping the overall rate of information transmission stable. The work of Aylett and Turk (2004) shows that one way this is accomplished is that high-surprisal words are pronounced with greater duration.

This connection between the lexical and the prosodic signals suggests that the two channels are somewhat redundant. If a lexical property like surprisal can be detected in the prosodic signal, the prosodic signal may be acting less as a conduit for novel information, and more as a back-up channel for information present in the lexical signal. One expected consequence of this would be that there would be minimal benefit from including prosody in most NLP tasks, but that benefit might grow as the lexical channel becomes less reliable. This idea motivates some of the research in this thesis, especially Chapter 6.

2.2.5 Prosody and noise

Research into the acoustics of speech have suggested one more important property of prosody: its robustness to noise. The work of van Zyl and Hanekom (2011) showed that even when a speech signal is overlaid with so much noise that words are hard to perceive, listeners are still able to recover linguistic information from the prosodic signal. These findings help to motivate the research in Chapter 6, since they suggest that prosody may be particularly effective as a 'back-up' channel for information from the lexical stream.

Moving to the world of speech synthesis, Govender and King (2023) use the technique of adding noise to a text-to-speech (TTS) audio signal that is presented to a human for evaluation. Though this work doesn't explicitly look into the influence of prosody, the findings of van Zyl and Hanekom (2011) suggest that the added noise is

more likely to obscure lexical information over prosodic information. This means that this work may potentially introduce a useful methodology for isolating the effect of acoustic correlates of prosody on perceived TTS quality, as well as other downstream tasks.

2.3 Application: Previous research on prosody in NLP

Given all the types of information that have been found in the prosodic signal, NLP researchers have tried various ways of taking advantage of this information using computational models. This section outlines various lines of research that involve incorporating prosody into computational models of language. Section 2.3.1 discusses the different representations of prosody that have been used as model inputs. Further sections concern the effect of these prosodic inputs on various tasks, including prosodic event prediction (Section 2.3.2), speech translation (Section 2.3.3), information status detection (Section 2.3.4), syntactic parsing (Section 2.3.5), and sentence segmentation (Section 2.3.6).

In general, despite the importance of prosody for human language processing, the effects of including prosodic inputs in NLP models are not always large. The experiments in later chapters of this thesis consider if there are tasks or experimental frameworks that might show the advantages of prosody more clearly. However, it is worth noting here that one reason that prosody may help NLP models less than it helps humans could simply be the differences between machine and human language processing. For example, humans have hard limits on the number of concepts that can be held in working memory (Cowan, 1996), which places an important constraint on real-time language processing. By contrast, a machine may be able to process much more information simultaneously than a human can, making this constraint less relevant. Additionally, for some models, the language input isn't being received as a constant stream that must be processed on-line. If the benefits of prosody to humans come from allowing humans to process language more efficiently, with less of a tax on working memory, then prosody simply may not be helpful for the processing problem that an NLP model faces. The work in this thesis doesn't rule out the possibility that this dynamic may play a role in reducing the benefits of prosody for NLP models. However, the experiments in Chapters 5 and 6 suggest that the picture is more complicated: In certain experimental conditions, NLP models do benefit from prosodic inputs, suggesting that prosody is relevant even to models that don't face human-like constraints such

as working memory.

2.3.1 Prosodic inputs to models

One major consideration in incorporating prosody into any model is how to represent the prosodic input. Where the input unit for traditional NLP models is generally a discrete unit (e.g., words or characters), acoustic correlates of prosody are generally the properties of a continuous acoustic signal. With the possible exception of timing features, which can be encoded as word duration or similar discrete measures, these continuous acoustic correlates of prosody are difficult to mesh with discrete lexical features. Of course, the continuous acoustic signal is often digitally processed as a series of short frames (e.g., 25 ms frames every 10 ms), but even though this is a type of discretization, it doesn't map easily to the discrete lexical features.

Previous research has generally gone in one of two directions: (1) converting acoustic correlates of prosody into word-level features in some way, or (2) leaving acoustic correlates of prosody in a pseudo-continuous form as features at the level of acoustic frames). One example of discretizing continuous features is Gregory et al. (2004), who use several different methods, such as taking the slope of the pitch track over the course of each word. Similarly, Levow (2005) uses the pitch of each syllable, sampled at five equal intervals. Other researchers map acoustic correlates of prosody to ToBI-like prosodic categories. For example, Wightman et al. (1991) map the timing of pauses and phones to ToBI-like break indices, and use these as input to their model, which is a syntactic parser.

Approach (2), using more continuous acoustic correlates of prosody, has become easier in the age of neural networks, which can handle this kind of data more easily. The research of Tran et al. (2018, 2019), Stehwien and Vu (2017), and Stehwien et al. (2018) are good examples of handling the acoustic correlates of prosody at the level of the acoustic frame, using convolutional neural networks. The work of Tran et al. (2018) and Tran et al. (2019) inspired the work in Chapter 5; Stehwien and Vu (2017) and Stehwien et al. (2018) inspired the work in Chapter 3.

These works (and the research presented in Chapters 3 – 6) still rely on having access to an explicit mapping between the acoustic frames and the corresponding words in order to line up the two types of representations. More recent work, such as wav2tobi (which is based on wav2vec (Baevski et al., 2020)), doesn't rely on this explicit alignment between words and acoustic correlates of prosody, and uses even

less-processed inputs in the form of the original audio, augmented with a pitch track (Zhai and Hasegawa-Johnson, 2023).

In this thesis, only the frame-level form of acoustic correlates of prosody is used (approach (2)), though still with human- or machine-generated alignments between the text and acoustic signals. The hypothesis motivating this choice is that approach (2) is superior to approach (1) because it includes more information from the original signal. The one possible exception is versions of approach (1) that use theoretically motivated prosodic annotations as an input, such as Wightman et al. (1991), which may arguably help abstract away from unhelpful or distracting parts of the acoustic signal, assuming the theoretical account of prosody used is a helpful one. However, the question of which theoretical accounts of prosody are best is beyond the scope of this thesis, so none of the experiments included directly compare approach (1) and (2), but rather stick to approach (2).

2.3.2 Prosodic event labeling

The first NLP task considered in this thesis is prosodic event labeling, as described in Chapter 3. In this task, the model maps from speech and text inputs to abstract prosodic categories, such as those used in the ToBI system. In the case of Chapter 3, the prosodic category of interest is the presence or absence of a pitch accent (see Section 2.1.2). The presence of a pitch accent is a ToBI category which can be fairly reliably identified by human annotators, so it makes a good prediction target. Some distinctions made by ToBI annotations, such as the difference between types of pitch accents, are often disagreed upon by annotators, and so make less reliable targets for prediction (Gut and Bayerl, 2004).

While this task isn't a common general-purpose NLP task, it does provide a good starting point as it removes the question of how prosody maps to linguistic categories. Experiments like the ones in Chapter 3 can focus on how to optimally access the prosodic signal, without the extra variable of how well that signal maps to relevant linguistic categories. The insights of this experiment inform the modeling approach taken in the further chapters of this thesis (Chapters 4 – 6).

Previous research on prosodic event detection has included efforts to detect all ToBI categories, including models such as AuToBI (Rosenberg, 2010) and wav2tobi (Zhai and Hasegawa-Johnson, 2023). Other work is more similar to ours in focusing just on predicting the location of pitch accents (Stehwien and Vu, 2017; Stehwien et al., 2018)

The focus of Chapter 3 is how to optimize pitch accent detection, with the primary finding that prosodic information is best processed with ample input context. This study additionally shows that some prosodic categories are predictable in large part from the text signal alone — an indication of the overlap between the two informational channels.

2.3.3 Speech translation

There are a few examples of research into the effects of prosody on speech translation. Rangarajan Sridhar et al. (2013) enrich a statistical speech translation system by using prosodic cues to predict the dialog act type of the source, and providing this information to the model. They find that this predicted dialog act feature improves translation quality according to a few different metrics.

Zhou et al. (2024) look at the question of how prosody affects Korean-to-English ST. They compare pipeline models (where prosodic information is inaccessible after an initial ASR step) to end-to-end models (where prosodic information is available throughout the translation model). They create a challenge test set to probe the model's ability to distinguish wh-questions from yes-no questions and statements in cases where the only differentiating signals are prosodic. The end-to-end model performs this task better than the cascade model, indicating that source prosody can affect target translation quality.

2.3.4 Information status detection

Since prosody is a signal to information status, it seems likely that it could be used alongside text inputs to directly detect properties like the givenness or newness of an entity. While there are no published studies on using computational models to detect precisely the given/new distinction, several models have been published on detecting the related (but distinct) phenomenon of *contrast*. Nenkova and Jurafsky (2007) define contrast as a feature of entities that are picked out of a limited set of entities in the discourse context. For example, in Example (7), *salmon* and *chocolate mousse* are marked as contrastive, since they are picked out of the set of possible dinner options, which is introduced by the question:

(7) Q: What did you have for dinner?
A: **Salmon**, and a **chocolate mousse** for dessert. (Nenkova and Jurafsky,

2007)

Like the given/new distinction, the contrastive/non-contrastive distinction is marked by the presence and type of pitch accents.

Nenkova and Jurafsky (2007) found that for the constrast detection task on spontaneous English speech, it is difficult to beat a baseline of just labeling adjectives and nouns as contrastive. Adding acoustic correlates of prosody didn't boost performance much beyond this baseline, unless non-pitch accented words were excluded from the task. Other somewhat similar studies on detecting contrast have been conducted as well (e.g., Heldner et al. 1999; Brenier et al. 2005), but in these studies, the quantity being detected is defined somewhat idiosyncratically, often with prosodic events as a core part of the phenomenon to be predicted (as in the pitch accent prediction task, see §2.3.2). On the whole, it seems that while prosody is clearly used in English to signal information status, the ability to detect this phenomenon directly with computational models hasn't been established yet.

2.3.5 Syntactic parsing

The work in Chapter 5 involves a syntactic parser that simultaneously parses and segments the input into sentences (see Section 2.3.6 on sentence segmentation). There has been a considerable amount of studies on incorporating prosodic information into syntactic parsing, which have yielded mixed results. For example, Gregory et al. (2004) incorporate prosodic cues as words in the input, and find that these cues are, if anything, slightly unhelpful. Later work by Kahn et al. (2004, 2005) and Hale et al. (2006) incorporates prosody in other ways into statistical parsers or parse rerankers, and shows gains from prosody, indicating that the way a model handles prosody may play a role in how helpful it is.

As parsers have improved in performance, the gains from incorporating prosody have generally shrunk (Jamshid Lou et al., 2019). Work by Tran et al. (2018) and Tran et al. (2019) has shown that incorporating prosody into state-of-the-art neural parsers does lead to performance gains, but these gains are relatively small. The work in Chapter 5 is based directly on the parsing model of Tran et al. (2019), but jointly performs both sentence segmentation and parsing.

2.3.6 Sentence segmentation

The models in Chapters 5 and 6 are trained to perform sentence segmentation, either alone or in combination with parsing. Because prosodic cues to sentence boundaries are relatively strong (see Section 2.2.2), sentence segmentation seems like a prime candidate for augmentation by prosody.

In previous work, the sentence segmentation task has taken different forms, depending on the sub-fields it appears in. For example, in many speech applications, the input must be chunked in some way, but the resulting chunks need not be syntactically coherent units, so prosody is not especially relevant (e.g., Jain et al. (2020)). In other applications, the focus is on adding punctuation to speech transcripts, a task with obvious connections to sentence segmentation, but with other output categories as well (e.g., question marks vs. exclamation marks vs. periods). Some of these puncutuation models have included prosody (Tilk and Alumäe, 2016), but text-only models remain competetive (e.g., Che et al. (2016); Alam et al. (2020)), likely because they are tested on non-spontaneous speech, which lacks many of the features of speech (e.g., disfluencies) that make it more difficult for machines to process.

In Chapters 5 and 6, the segmentation task is framed as identifying all the human-identified sentence breaks in the transcript. This task framing is motivated more by determining what linguistic information is present in prosody, rather than by creating output that is suitable for one particular application, though this output should be widely usable in tasks that need segmented input.

The most direct predecessors to this work are studies such as Kahn and Ostendorf (2012), which frame the segmentation task in a similar way and use spontaneous speech that has disfluencies. They find that prosody helps significantly with sentence segmentation. Kahn and Ostendorf (2012) also combine sentence segmentation with parse reranking on ASR transcripts, which is similar to the combination of parsing and sentence segmentation in Chapter 5, though their parsing-and-reranking approach is substantially different from the parsing approach used here. They find that using prosody in reranking is helpful in improving parsing, sentence segmentation, and the WER from the ASR system.

This thesis makes several novel contributions to this line of research on sentence segmentation. First, the approach in Chapter 6 combines the task of sentence segmentation with handling ASR output with noise added. Second, Chapter 5 combines sentence segmentation and parsing in a state-of-the-art parser. As discussed in Sec-

tion 2.3.5, some of the contributions prosody has made to parsing are arguably due to lower-performing parsers that can't fully exploit the information present in the lexical signal. Using a high-performing parser allows for the actual contributions of prosody to be more clear.

Chapter 3

Pitch accent detection

3.1 Introduction

This chapter is focused on the task of pitch accent detection. Pitch accents are prosodic prominences that are signaled by a deviation from the speaker's usual pitch, duration, energy, voice quality, or some combination of these features. While the task of detecting which words carry pitch accents isn't a standard NLP task with obvious applications, it is a good way to focus on how best to handle input features for the acoustic correlates of prosody. Modeling approaches that improve performance on pitch accent detection may also improve performance on any downstream task that benefits from better access to prosodic information. Excluding this downstream task for now removes the extra variable of how prosodic information maps to linguistic categories, which will be reintroduced in later chapters. Furthermore, using both text and prosodic inputs for pitch accent detection offers a more direct way to investigate a motivating question of this thesis: how much of the prosodic signal (or at least the important category of pitch accents) is predictable from the text signal alone.

Additionally, the task of pitch accent detection has been investigated by several researchers, allowing for comparison between modelling approaches and their resulting performance. Previous pitch accent detection models include rule-based models (Brenier et al., 2005), traditional machine learning models (Wightman and Ostendorf, 1994; Levow, 2005; Gregory and Altun, 2004), and neural models (Fernandez et al., 2017; Stehwien and Vu, 2017; Stehwien et al., 2018). Most recently, Stehwien and Vu (2017) and Stehwien et al. (2018) showed that neural methods can perform comparably to traditional methods using a relatively small amount of speech context — just a single word on either side of the target word. The primary goal of the present work

is to consider whether providing even more context to a pitch accent detection model like those in Stehwien and Vu (2017) and Stehwien et al. (2018) can improve accuracy. Since pitch accents are deviations from a speaker's average pitch, energy, and duration, a wider input context ought to allow the model to better determine the speaker's baseline for these features and therefore improve its ability to detect deviations. This result would fit with the findings of several pre-neural studies, including Levow (2005, 2008); Rosenberg and Hirschberg (2009). In addition, a recurrent model (rather than the CNN used by Stehwien and Vu (2017); Stehwien et al. (2018)) should also improve performance, since it is better adapted to processing long-distance dependencies.

The model presented here is a neural pitch accent detection model that takes in features corresponding to acoustic correlates of prosody, text features, or both. This model has access to more context than Stehwien et al. (2018), and does in fact show improved performance on the corpus they used for evaluation, the Boston University Radio News Corpus (henceforth BURNC; Ostendorf et al. (1995)). In addition to these experiments done on BURNC, this model is also tested on the Switchboard-NXT corpus (henceforth SWBD-NXT; Calhoun et al. (2010)), a corpus of spontaneous telephone conversations. Including this second corpus allows for discussion of the role of genre on pitch accent production and detection.

Another important finding of this chapter is that the contributions of text input actually are fairly simple: A baseline of simply labeling all content words with pitch accents matches the performance of the text-only model. This shows that the useful information from the text signal is mostly reducible to the content vs. function word distinction. In Section 3.4, I argue that this content-word baseline is the correct baseline for this task. Additionally, when this content-word baseline is taken into consideration, the contributions of the prosodic inputs to the model become clearer: The prosody-only model is able to outperform this baseline by detecting some of the cases where a speaker deviates from the labels produced by the content-word baseline. This chapter concludes with an analysis of which acoustic correlates of prosody yield the most benefit.

The main contributions of this chapter are summarized as follows:

- Showing that text-only inputs are able to perform relatively well at the task of pitch accent detection, confirming that at least some parts of the prosodic signal are quite predictable from text alone.

- Showing that context-enhancing innovations in the model improve accuracy and

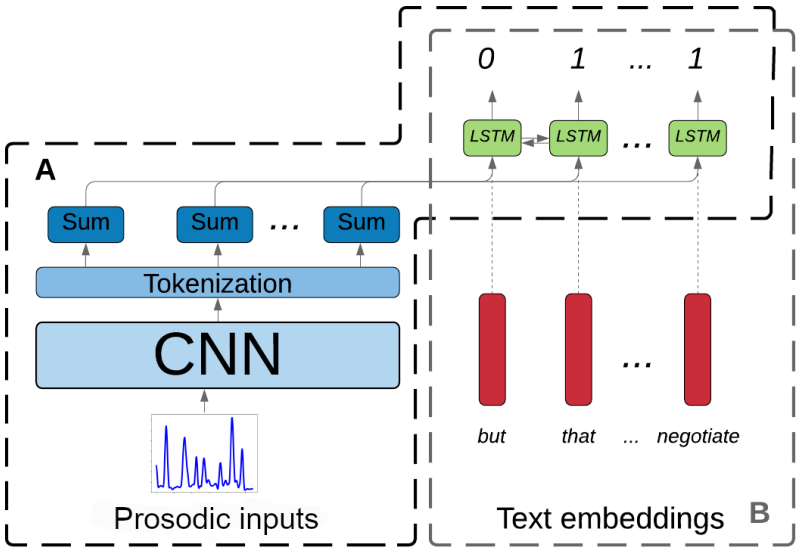


Figure 3.1: The combined prosody+text model. Box A outlines the prosody-only model components, while box B outlines the text-only model.

F1 on this task on two corpora of American English speech (BURNC Ostendorf et al. (1995) and SWBD-NXT Calhoun et al. (2010)), compared to previous published models.

- Demonstrating that the contributions of the text signal are largely reducible to the content vs. function word distinction, and arguing for the content-word baseline as the most appropriate baseline for this task.

- Showing the effect of genre on pitch accent detection, specifically the greater difficulty of pitch accent detection for spontaneous speech.

- Showing the effect of different acoustic correlates of prosody on pitch accent detection.

3.2 Model

The pitch accent detection model takes in text features, features corresponding to acoustic correlates of prosody, or both. It outputs a label for each word, indicating whether that word carries a pitch accent of any kind. The variants of the model with prosody-only, text-only, and both forms of input are shown in Figure 3.1 and described below.

3.2.1 Prosody-only model

Like Stehwien et al. (2018)'s model, the prosody encoder begins with several CNN layers that take a series of frames $f_1, f_2, ..., f_n$ as input, where each frame f_i is a vector of six features for the acoustic correlates of prosody. These features encode pitch, energy, and voicing information. See Section 3.3.4 for more discussion of these features. These frames are encoded by the CNN, which reduces the overall number of frames by passing a kernel over the input with a stride of size 2, resulting in frames $f'_1, f'_2, ..., f'_k$.

Since the motivating hypothesis of this work is that more context will help the model, this model labels the whole sequence at once, rather than outputting the label for a single word at a time, as Stehwien et al. (2018) do. This introduces the difficulty of aligning a timeseries of features extracted from the acoustic signal — which have been passed through several layers of CNN — to the representations corresponding to each word of input. Fortunately, the corpora being used (BURNC and SWBD-NXT) both have human-generated word token timestamps, which are used to divide the post-CNN acoustic features at the places that correspond to word boundaries. The final difficulty is that each resulting subdivision of the frames $[f'_i, f'_{i+1}, ..., f'_j]$ contains different numbers of frames, since words are of various lengths. To obtain word-token-level prosodic representations of identical size, the model sums across all frames for a given word token: $t_j = sum(f'_i, f'_{i+1}, ..., f'_k)$.[1] Each word-token-level prosodic embedding $t_1, ..., t_m$ is then passed into a bidirectional LSTM, and finally a feed forward layer that outputs a label for each word.

While the full model takes an entire utterance as input and outputs all labels at once, this chapter also presents the results of experiments with using only three or one word(s) as input. The three-word scenario is designed to be most similar to Stehwien et al. (2018)'s model, for comparison purposes. In these cases, the model only outputs the label for the central input word.

3.2.2 Text-only model

The text-only model is a simple bidirectional LSTM. An embedding for each word is passed to the BiLSTM and a prediction is made at each timestep. Following Stehwien et al. (2018), the model uses pretrained 300d GloVe word embeddings Pennington et al.

[1] This is not dissimilar to the approach taken by Tran et al. (2018), but they choose to maxpool across the frames for each word token, rather than sum. Experiments with this model showed that summing across the frames for each word token performed better than pooling for this task. See Section 3.3.6 for further discussion.

(2014), although using pre-trained embeddings does not improve performance much over randomly initialized embeddings.

3.2.3 Prosody+text model

The prosody-only and text-only models both include a bidirectional LSTM, so in the combined model, the prosody embedding for each word token generated by the CNN encoder is simply concatenated with the pretrained text embedding for that word before being passed to the LSTM.

3.2.4 Model ablations

In Section 3.4, the performance of the model described above is compared to the performance with various ablations. These ablations can affect either (1) the type of input features (§3.2.4.1), (2) the model architecture and the amount of input context (§3.2.4.2, or (3) the size of the model's vocabulary (§3.2.4.3

3.2.4.1 Ablation of acoustic correlates of prosody

Ablating groups of input features corresponding to acoustic correlates of prosody helps determine how important each type of feature is to overall model performance. Rather than ablate one feature at a time, the features are first grouped into those related to pitch (smoothed F0), energy (RMS energy, loudness), and voicing (harmonics-to-noise ratio, zero-crossing rate, voicing probability). One or more feature groups are then ablated at both the training and testing stages. The results of these ablation experiments are reported in Section 3.4.1.2

3.2.4.2 Model architecture and context ablation

Compared to Stehwien et al. (2018), the main architectural addition to the model presented here is a bidirecitonal LSTM that is added after the initial CNN. The motivation for including the LSTM is that prosodic events such as a rise or fall in pitch are defined against the backdrop of the speaker's average fundamental frequency, and so the model will need to be able to compare pitch features over relatively long distances, something that LSTMs excel at. In order to determine if the LSTM does play a beneficial role, the model is also run as simply the CNN component, plus a feed-forward layer. The results of this ablation are reported in Table 3.4.

The other major difference between the model presented here and Stehwien et al. (2018)'s model is that this model recieves an entire utterance as input, and outputs labels for the entire sequence of words. In contrast, Stehwien et al. (2018)'s model receives three words as input, and only outputs the label for the central word. Again, this change is motivated by the idea that when using prosodic inputs, there is a benefit to having more context.

This extra context is ablated by comparing the full-utterance model to a three-word model like Stehwien et al. (2018)'s. For the three-word model, since only one label is to be output, the prosodic input is also not split into words, as it is in the full-utterance model. Instead, following Stehwien et al. (2018), an additional dimension is appended to the prosodic feature vectors, taking the form of a binary indicator of whether or not that prosodic feature vector belongs to the central word, which is the one being labeled.

Of course, since the features corresponding to acoustic correlates of prosody are not tokenized, the strategy of concatenating word-level acoustic features and text embeddings is impossible. Instead, for the prosodic + text model, both the prosodic representations (post-CNN) and the text embeddings are flattened into one dimensional embeddings, and concatenated. This approach follows that of Stehwien et al. (2018), with some assumptions made about their exact implementation.

3.2.4.3 Vocabulary ablation

The final type of experiment in this category is the ablation of the text-only model's vocabulary size. These experiments are aimed at determining what parts of the text signal are actually important for the the text-only model's performance. With a vocabulary of size N, all words except for the top N most frequent words in the training set will be represented as unknown word tokens (i.e., UNK for the model. The word embedding for the UNK input token is randomly initialized and it is trained along with the rest of the model. Vocabulary sizes from 1000 word types all the way down to just 5 word types are tried.

3.2.5 Baselines

In addition to a majority class baseline, a content-word baseline is included, where all content words (which are approximated by considering all non-stopwords to be content words as identified by NLTK) are labelled as carrying a pitch accent. As discussed in section 3.4, there are strong indications that the text-only model is relying mostly on

the content word vs. function word distinction. Therefore, this baseline is particularly interesting in helping to show how much of the text-only model's performance may reducible to its ability to distinguish these categories. Furthermore, since the duration of a word is highly correlated with whether or not it is a content word, a duration-only baseline is included, in order to see how much of the *prosody-only* model's performance may be attributable to the content vs. function distinction. In the duration-only baseline, the acoustic input features to the prosody-only model are all replaced with the value 1, so that the model can only tell how many frames each word contains.

3.3 Data and experimental setup

The models described in Section 3.2 are trained and evaluated on two corpora: the Boston University Radio News Corpus (BURNC; Ostendorf et al. (1995)) and Switchboard-NXT (SWBD-NXT; Calhoun et al. (2010)). All performance numbers reported for a corpus reflect that the model was trained and evaluated on that corpus. Where BURNC consists of read speech from trained radio announcers, SWBD-NXT is made up of spontaneous telephone conversations. Differences in genre can lead to significant differences in how prosody is used, which motivates the use of both these corpora. Margolis et al. (2010) note some of these prosodic differences, including (but not limited to) the fact that conversational speech is generally faster and more disfluent, while read speech tends to have a higher rate of pitch accents overall.

3.3.1 BURNC

BURNC is a speech corpus of General American English, primarily gathered from recordings of the WBUR FM radio news station in Boston, with additional recordings from the radio announcers in other contexts. The corpus is partially annotated with prosodic information using the ToBI labeling conventions (Pierrehumbert, 1980). These include pitch accent locations, pitch accent types, boundary types, boundary tones, and phrasal tones, though only pitch accent locations are relevant for this study. The transcription of the speech in BURNC includes automatically detected breaths, which are used here to segment the corpus into utterances.

The annotated subsection of the corpus that is used to train and evaluate the model includes five speakers, three female, and two male, all of them trained radio journalists reading pre-written news segments. The data used amount to approximately 2.75

hours of speech, consisting of 1721 utterances. These come from a total of 398 news segments. There are approximately 28.5k words, 15.5k of which carry pitch accents. Though this is a limited amount of data, all of the (small number of corpora) with human-generated prosodic annotations are quite small, since this kind of annotation requires expert annotators and is time-intensive. Using BURNC specifically enables us to compare with previous studies that use this resource, including (Stehwien and Vu, 2017) and (Stehwien et al., 2018).

3.3.2 SWBD-NXT

SWBD-NXT is a speech corpus of General American English, based on the Switchboard corpus (Godfrey et al., 1992). The SWBD-NXT project added a variety of rich annotations to a subset of the Switchboard data, ranging from syntactic trees to information status annotations. For this chapter, the relevant annotations are ToBI-style prosodic annotations, which include pitch accent locations (along with other ToBI categories which aren't used here). SWBD-NXT is already divided by annotators into utterances, which are the input units for the model.

The portion of SWBD-NXT that is annotated with pitch accents consists of 25 conversations, which are divided into 5,221 utterances, amounting to approximately 43.1k total words. Of these, 15.4k words are marked as having pitch accents.

3.3.3 Corpus differences

These two corpora have very different genres — read speech in the case of BURNC, and casual telephone conversation in the case of SWBD-NXT. This leads to several differences that are relevant to the experiments here.

First and most notably, only about 36% of words are pitch accented in SWBD-NXT, whereas about 54% of the words in BURNC are pitch accented. This accords with previous observations that read speech contains more pitch accents overall; for example, Yuan et al. (2005) found that 52-54% of words were pitch accented in read speech, compared to 46-49% in spontaneous speech. This is one of a number of observed differences in prosody between read and spontaneous speech, including differences where prosodic fall (Ayers, 1994) and differences in how sentence boundaries are marked (Margolis et al., 2010).

Second, despite the fact that the SWBD-NXT corpus is larger (43.1k words compared to BURNC's 28.5k), it has a smaller vocabulary (roughly 2500 word types, com-

pared to BURNC's 3000).

3.3.4 Extracting features for acoustic correlates of prosody

Following Stehwien and Vu (2017) and Stehwien et al. (2018), the OpenSMILE toolkit (Eyben et al., 2013) is used to extract six features corresponding to acoustic correlates of prosody from the audio, which fall into three broad categories: pitch features (smoothed F0), energy features (RMS energy, loudness[2]), and voicing features (zero-crossing rate, voicing probability, and harmonics-to-noise ratio). The exact procedure used for feature extraction is taken from the Interspeech 2013 shared task on paralinguistics (Schuller et al., 2013), for which OpenSMILE configuration files were released that could be used to produce these (and other) features. Using Schuller et al. (2013)'s procedure, these acoustic correlates of prosody are extracted from frames of varying sizes, with frames offset by 10 ms.

One concern for the feature extraction process is that telephone recordings — like those that comprise SWBD-NXT — can lead to unreliable pitch estimates (Wang and Seneff, 2000). To address this, the utterances from four SWBD-NXT conversations are manually surveyed. A spectrogram with the pitch feature for each of these utterances was visually checked against the corresponding audio recording, to make sure that the pitch feature roughly matched the perceived pitch contour of the utterance. While there is too much data to manually verify all of it, this inspection suggested that the pitch feature is capturing something like fundamental frequency. However, the results obtained for SWBD-NXT do suggest that the quality of the acoustic features extracted from SWBD-NXT using the above procedure may be an issue (see Section 3.4).

The three categories of features — pitch, energy, and voicing — represent three of the four primary acoustic correlates of prosody discussed in Chapter 2. The fourth — duration — isn't modeled explicitly here, but the number of acoustic frames per word is a direct indication of duration. As noted in discussion in Section 3.3.6, this may be one reason that summing across the prosodic frames for each word token is better than maxpooling — it better preserves information about number of input frames, and therefore, duration. One possible future addition to these three features categories used here would be an explicit feature for the duration of a given word, normalized by the average duration of that word in the corpus, as used by Tran et al. (2018). While the absolute duration of a word is informative (and is in fact the focus of one ablation test

[2]Specifically the "PCM loudness" feature in the OpenSmile library, defined in the documentation as "loudness as the normalised intensity raised to a power of 0.3."

below in Section 3.4), the length of a word *relative* to its average production is a better indicator of if it is, for example, being pronounced longer because it carries a pitch accent.

It is important to note that the prosody-only model has no access to phone-level or spectral information that might allow it to make predictions based on word identity. While audio-derived features such as MFCCs or filterbank features might carry lexical information that a sufficiently advanced model could exploit, this model sees only the features noted above, which correspond to the acoustic correlates of prosody. Therefore, the model gets no lexical information through the prosodic channel, except for indirect information about word length from the number of acoustic frames in the input.

3.3.5 Evaluation procedure

The model is trained and evaluated on one of the two corpora, BURNC or SWBD-NXT. One challenge of this task is that the available labeled corpora are relatively small, which makes training and then reliably evaluating models difficult. To address this problem, except where noted otherwise, all experiments are evaluated with tenfold cross-validation, with the average performance reported.

The exact process used for tenfold cross-validation is as follows. If the corpus had a total of 100 utterances, the first step would be to shuffle the utterances, and designate utterances 1-10 as the test set. From the remaining 90 utterances, 10 would be randomly designated as development and 80 as training, which would create the first train/development/test split. Next, utterances 11-20 would be designated as the test set, and the development and training sets would be selected from the remaining 90 utterances. This process would be repeated until 10 distinct train/development/test splits had been created, each with a unique test set. To cross-validate a model, it is trained and evaluated on all 10 of these data splits. The development portions are used to determine where to stop training for each split. The model trains for 25 epochs, and the epoch with the highest development set accuracy of these 25 epochs is used to report development and test set accuracy.

Both accuracy and F1 score are reported for all models. Accuracy is included since this is the metric used by Stehwien and Vu (2017) and Stehwien et al. (2018) for the BURNC corpus, where the distribution of pitch accents makes this a balanced binary classification task. However, SWBD-NXT has a much lower proportion of pitch

accented words, and therefore accuracy is a less informative metric. To address this deficiency of the accuracy metric, F1 score is also included. F1 score is the harmonic mean of precision and recall, defined as follows:

$$Precision = \frac{true\ positives}{true\ positives + false\ positives}$$

$$Recall = \frac{true\ positives}{true\ positives + false\ negatives}$$

$$F1 = 2\frac{precision \cdot recall}{precision + recall}$$

For some comparisons between models, the statistical significance of the difference in performance is reported. This significance is calculated using pairwise bootstrap resampling (Efron and Tibshirani, 1994), with 10k resamples. A value of $p < 0.05$ is reported as significant. Note that because this procedure uses a pairwise resampling between the results of two models, the results must be comparable. For this reason, it's not possible to use this procedure to calculate the difference in significance between models that output labels for sequences and those that output labels for each word.

3.3.6 Model hyperparameters

Hyperparameter	Possible values	Selected values
CNN layers	2, 3, 4	3
LSTM layers	2, 3	2
Dropout	0, 0.2, 0.5, 0.7	0.5
Weight decay	0, 10e-5, 10e-4	10-e5
Filter width	9, 11, 13, 15, 17, 19, 21, 23	11
Post-tokenization	sum, max	sum

Table 3.1: Possible and selected values for each hyperparameter considered in the search. The 'post-tokenization' hyperparameter corresponds to the method used to collapse the word-token-level prosodic representations — max pooling or summing across all frames.

The model is trained for a total of 25 epochs, using a batch size of 64. For the BURNC corpus, each epoch consists of approximately 1400 training examples, and the prosody-only and prosody+text models take a total of about 200 seconds to train on average, with each epoch taking around 8 seconds to train on a single Titan X-equivalent GPU. The text model takes about 75 seconds to train, with an average of 3 seconds per epoch. Evaluation on the entire development set of BURNC (about 200

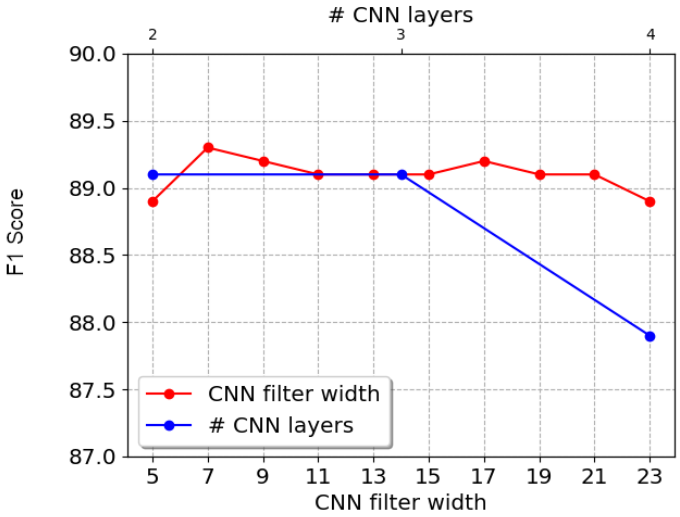


Figure 3.2: The performance of the prosody-only model on BURNC given different CNN hyperparameters, tested on a development set using tenfold cross-validation. When varying CNN filter width, the number of CNN layers was held at 3; when varying the number of CNN layers, the filter width was held at 11 frames.

instances) takes an average of 2 seconds to run for the prosody-only and prosody+text models, and 1 second for the text model.

A hyperparameter search was performed on BURNC for the combined model, and the resulting hyperparameters were used for all input configurations (text, prosody, prosody+text). The possible values of each hyperparameter are as shown in Table 3.1, with each hyperparameter configuration being chosen at random from these values. The selected value for the hyperparameter is shown in the right column. Ninety-six distinct hyperparameter configurations were run, picking the configuration with the highest accuracy on the development set. The average performance on the development set over the search space was 83.5% accuracy, with a variance of 0.005 and a standard error of 0.007.

Other hyperparameters were selected manually without searching: the first CNN layer has 128 kernels, with 256 kernels in all subsequent CNN layers, and a stride length of 2 throughout all CNN layers. The LSTM layers each have a hidden size of 128. PyTorch's Adam optimizer is used for training, with a learning rate of 0.001 (Paszke et al., 2019). For the text inputs, a vocabulary size is set at approximately 80% of the total types present in each corpus. For BURNC, this results in a vocabulary size of 3000 types, with a vocabulary size of 2500 for SWBD-NXT.

Many of the hyperparameter experiments focus on changes to the CNN that should allow it to process a wider swath of the input at once: adjusting filter width, and ad-

		Speech		Text		Speech + Text	
		Acc	**F1**	**Acc**	**F1**	**Acc**	**F1**
BURNC	**CNN+LSTM model**	89.13	90.21	84.44	86.49	**89.73**	**90.73**
	S18, as reported	87.1	—	78.5	—	87.5	—
	S18 replication	87.37	88.72	83.20	85.31	85.39	87.05
	Majority baseline	—	—	—	—	55.25	71.18
	Content-word baseline	—	—	81.04	83.75	—	—
	Duration-only baseline	80.50	83.60	—	—	—	—
SWBD	**CNN+LSTM model**	73.49	62.83	75.50	63.56	**76.63**	**67.64**
	S18 replication	73.50	64.30	72.94	60.98	73.40	62.81
	Majority baseline	—	—	—	—	64.15	0
	Content-word baseline	—	—	68.97	61.48	—	—
	Duration-only baseline	73.06	62.29	—	—	—	—

Table 3.2: Test set F1 score of the CNN+LSTM model, compared to a replication of Stehwien et al. (2018)'s model (abbreviated S18 above). The differences between the two are that Stehwien et al. (2018)'s model (1) only includes a CNN, rather than a CNN+LSTM, and (2) only takes in three words at a time, rather than a whole utterance. Also included for comparison are three baselines: a majority baseline; a baseline where all content words are labelled as accented; and a duration-only baseline, where the model only has information about the number of frames per word, and then the model is trained and evaluated as normal.

justing the number of CNN layers. Neither change showed a sizeable positive effect, and both were harmful when taken to the extreme. As can be seen in Figure 3.2, given a constant depth of 3 CNN layers, the very narrowest kernels underperformed, but widening the kernel did not consistently produce better performance, and eventually degraded performance. Likewise, adding CNN layers — which increases the number of frames of the input data being viewed by the final CNN layer — was actively harmful to performance beyond depths of 3 layers.

3.4 Results and discussion

Table 3.2 shows the general performance of the model on the test split of both BURNC and SWBD, as measured by F1 score for the positive category. For comparison, three baselines are included: a majority category baseline, a content-word baseline (where all content words are marked as carrying pitch accents), and a duration-only baseline, where the only information the model receives is the number of acoustic frames per word. Additionally, the reported results of Stehwien et al. (2018)'s model are included for BURNC. Since Stehwien et al. (2018) do not use SWBD-NXT, a replication of this

model is also included in this table, with results reported for both corpora.

As mentioned in Section 3.3.5 above, accuracy is included in Table 3.2 primarily to allow for comparison with the results Stehwien et al. (2018) obtained on BURNC.

The highest performing model for both corpora is the CNN+LSTM model with both prosodic and text inputs. Removing either prosody or text reduces performance, though for BURNC, the performance loss from removing text is not statistically significant. In general, both the prosody-only and text-only models perform at a high percentage of the prosody+text model's performance. The high performance of the text-only model in particular suggests that the location of pitch accents is mostly predictable from the text signal alone, indicating a degree of redundancy between the lexical and prosodic channels.

Performance also drops when the LSTM and the extra input context are removed, as in the replication of Stehwien et al. (2018)'s model (though as discussed in Section 3.3.5, statistical significance isn't computable for the comparison between the full CNN+LSTM model and the replication of Stehwien et al. (2018)'s model). This drop suggests that the combination of the longer input context and the LSTM is helpful.

Turning to the question of model architecture, the underperformance of the CNN-only model cannot be attributed to overfitting, since the CNN+LSTM model is more overfit to the training set than the CNN-only model: The CNN+LSTM model's accuracy on BURNC's train set is approximately 93% accuracy, compared to the CNN-only model's 91%. Furthermore, the replication of Stehwien et al. (2018)'s model has more parameters than the CNN+LSTM model (13M vs. 12M parameters)[3], so the improvements of the latter are not due to model size.

For both corpora, the content-word baseline is relatively high-performing, compared to the majority-class baseline — the content-word baseline's F1 score is only approximately 3% lower than the corresponding text-only CNN+LSTM model's performance. This suggests that the content-function distinction is a large part of the information that the text-only model is able to extract from the text signal in order to perform this task. Given how well the simple content-word baseline performs, I argue that future work on this task should include this baseline, in order to clarify what added benefit is gained from a more complex model. Interestingly, the prosody-

[3]This is counterintuitive at first glance, since the CNN+LSTM (obviously) has an entire LSTM's worth of more parameters. The extra parameters in the replication of Stehwien et al. (2018) are found in the final feed-forward layers of model, which takes in acoustic frames as input, rather than word tokens, as in the CNN+LSTM model. Far more acoustic frames than word tokens are required to represent the same sizes of input.

only model tends to outperform the text-only model in precisely those places where the speaker's realization deviates from the content-word baseline. For example, the BURNC prosody-only model can correctly detect some pitch accents that fall on function words (as in the word *that* in example (1)) and unaccented content words (as in the word *Mary* in example (2)), while the text-only model cannot.

(1)

Input:	*but*	*that*	*would*	*require*	*the*	*union*
Gold:	0	1	0	1	0	1
Content-word:	0	0	0	1	0	1
Text:	0	0	0	1	0	1
Prosody:	0	1	0	1	0	1

(2)

Input:	*she*	*agrees*	*with*	*Mary*	*Conroy*
Gold:	0	1	0	0	1
Content-word:	0	1	0	1	1
Text:	0	1	0	1	1
Prosody:	0	1	0	0	1

Only considering those words where the speaker's production deviates from the content-word baseline, the BURNC prosody-only model achieves 66.7% accuracy, vs. only 38.2 percent for the text-only model.

Comparing the two corpora, both the baselines and the model results for SWBD-NXT are substantially lower than for the BURNC model, suggesting that this is a more difficult dataset for this task. There are several possible reasons for this. Most obviously, there are simply fewer pitch accents overall in SWBD-NXT, with only about 36% of words carrying a pitch accent in SWBD-NXT, compared to about 54% in BURNC. Also, the content-word distinction is simply less predictive for SWBD-NXT: While almost 80% of content words are pitch accented in BURNC, only 55% of content words are pitch accented in SWBD-NXT. This seems to clearly be a source of SWBD-NXT's elevated difficulty.

Another difference between the two corpora is that the prosodic inputs seem to be a relatively better signal for the BURNC models, while text inputs are relatively stronger for the SWBD-NXT models. The BURNC prosody-only models all outperform the corresponding text model by a wide margin, while the SWBD-NXT models have a more mixed record: The CNN+LSTM model performs better with text alone compared to with prosody alone, but the CNN-only model (i.e., the replication of Stehwien et al. (2018)) has better performance from prosody alone than from text *or* prosody+text.

As seen above, this is likely not because SWBD-NXT's text signal is more in-

	CNN+LSTM model (prosody + text)	**Stehwien & Vu 2017** (prosody only)
f1a	**89.43**	85.6
f2b	**88.14**	82.9
f3a	**89.65**	83.5
m1b	**85.05**	81.4
m2b	84.42	**84.8**

Table 3.3: Speaker-independent results of the prosody+text model, identified by speaker IDs in BURNC. Scores are compared to the prosody-only model of Stehwien and Vu (2017).

formative. The SWBD-NXT text model's performance is still largely accounted for by the content-word baseline, and content words are if anything *less* predictive for SWBD-NXT. Differences in the speech signal are likely to blame. One obvious difference is that SWBD-NXT's audio recordings are simply of relatively lower quality than BURNC's, making the prosodic signal less reliable: The sample rate of the BURNC audio is 16 kHz, compared to 8 kHz in SWBD-NXT. Additionally, the SWBD-NXT audio is recorded from telephone conversations, which effectively is both high- and low-pass filtered. The high performance of the duration-only baseline on SWBD-NXT relative to the prosody-only model also supports the idea that the input features for the acoustic correlates of prosody may be of lower quality: Adding actual prosodic information to the acoustic frames only slightly boosts performance for this corpus, though it makes a large difference for BURNC.

One other interesting evaluation case for these models is speaker-independent evaluation. That is, evaluating on a single speaker that is held out for testing and all the other speakers being used for training and development. Since the SWBD-NXT has many more speakers, and less audio per speaker, this experiment is only conducted on BURNC results are shown in Table 3.3. There are no published results of a prosody+text model evaluated in this test condition to compare to. However, compared to the results of the prosody-only model of Stehwien and Vu (2017), this model outperforms theirs on all speakers except the speaker identified as *m2b*. The reasons for this underperformance are unclear.

	Context	Architecture	Prosody-only	Text-only
BURNC	**Content word**		—	85.47
	Full utt.	**CNN+LSTM**	90.21	86.49
		CNN only	89.19	86.62
	Three tok.	**CNN+LSTM**	89.84	86.02
		CNN only	88.72	85.31
SWBD	**Content word**		—	60.59
	Full utt.	**CNN+LSTM**	62.99	64.25
		CNN only	60.17	62.40
	Three tok.	**CNN+LSTM**	66.37	62.94
		CNN only	64.62	61.53

Table 3.4: Development set F1 score of prosody-only and text-only model variants using different input contexts and architectures. Greater input context helps most of the time, and models that include an LSTM layer work better than just CNN-only models.

	Context	Architecture	Prosody-only	Text-only
BURNC	**Content word**		—	83.41
	Full utt.	**CNN+LSTM**	89.13	84.44
		CNN only	87.88	84.37
	Three tok.	**CNN+LSTM**	88.67	83.78
		CNN only	87.37	83.20

Table 3.5: Development set F1 score of prosody-only and text-only model variants using different input contexts, for BURNC.

3.4.1 Ablations

3.4.1.1 Ablating the model architecture and input context

While the results in Table 3.2 show that the model proposed here is the highest performing of those tested for both corpora, this model differs in two ways from Stehwien et al. (2018)'s: (1) It includes an LSTM step after the CNN, and (2) it takes whole utterances as inputs, instead of just three words at a time. Both of these changes increase the amount of acoustic context available to the model, but the relative effect of each of these changes isn't clear from just comparing these two models. In order to clarify the effect of these changes, Table 3.4 reports results for the prosody-only and text-only models with and without the LSTM,[4] and with and without the full utterance context. The accuracy scores for these same models are reported in Table 3.5 for BURNC. All these results are reported on the development set, with tenfold cross-validation.

The results in Table 3.4 support the hypotheses that for prosodic inputs, both a full utterance of input context and a long-distance-adept model like a LSTM are helpful. The performance gap between the CNN+LSTM vs. CNN only model is significant for both corpora, with both a full utterance of context, and a three-word window.

The one exception to the pattern of more context being beneficial is the prosody-only SWBD-NXT model, where both model architectures perform better with only the three-word context. It's unclear why there is this improvement in performance with a drop in the amount of context, especially since the inclusion of the LSTM — a change that should help the model to take advantage of longer-distance context — *does* help the SWBD-NXT prosody-only models.

The BURNC development set experiments with the text-only model find a relatively small effect from context and architecture (see Table 3.4), compared to the prosody-only model. This suggests that the information that this text model relies on isn't affected by access to context, which fits with the finding that a large part of the text model's performance is attributable to its ability to distinguish content and function words — a task that isn't context-dependent. However, the SWBD-NXT text-only models do show effects of both context and architecture that are closer to the effect sizes for the prosody-only model. It isn't immediately clear why context would help

[4]It's worth noting that the CNN-only model with full utterance context actually has substantially fewer trainable parameters than all the other models. Compared to the CNN-only model, the CNN+LSTM model has the added parameters of the LSTM. However, the three-word model also has more parameters in the final feed-forward steps, since the inputs to this layer are audio frames, rather than word tokens. This leaves only the full-utterance, CNN-only model with fewer parameters ($\tilde{2}$M, compared to $\tilde{1}$2-13M).

more in this case, since the SWBD-NXT text-only models also tend to perform close to the content-word baseline. It may simply be that since the prosodic signal is less helpful in SWBD-NXT, the amount to be gained from more prosodic context is roughly the same as the amount to be gained from more text context.

3.4.1.2 Ablating prosodic inputs

The duration-only baseline shown in Table 3.2 shows that the prosody-only model is able to perform quite well given only information about word length, without access to explicit prosodic inputs for pitch, energy, and voicing, but that for BURNC at least, these prosodic inputs are still used in achieving the prosody-only model's performance. To determine the importance of each of type of prosodic inputs, ablation tests are carried out as described in Section 3.2.4.1. The prosodic inputs are grouped into those related to pitch, energy, and voicing, and then one or more feature groups are ablated. The results of these experiments on the development set can be seen in Figure 3.3.

The effect of the prosodic inputs is quite different for the SWBD-NXT model compared to the BURNC model. While both models suffer the least from ablating energy, all other feature combinations have different effects. For BURNC, pitch seems to play the biggest role of these inputs, with its ablation leading to the lowest performance. Voicing appears to be the weakest feature for BURNC, actually harming model performance in one case: energy and voicing features combined underperform energy features alone.

On the whole, the differences from ablations are much smaller and much noisier for SWBD-NXT. The 'all' features condition is not the highest performing, nor is the 'none' condition the lowest, suggesting that the differences seen here are more likely noise rather than meaningful variation.

3.4.1.3 Ablating vocabulary

As noted above, the content-word baseline is extremely strong for both models, especially when compared to the text-only model's performance. This conclusion is further supported by an additional analysis where the vocabulary size of the text-only models is progressively reduced from 1000 down to 5. As shown in Figure 3.4, performance is basically steady until the vocabulary size drops below 100 words (with the rest of the word types labelled as 'UNK'). The vocabulary is organized in terms of frequency, so that when the model is running with a vocabulary of only 100 words, these are the

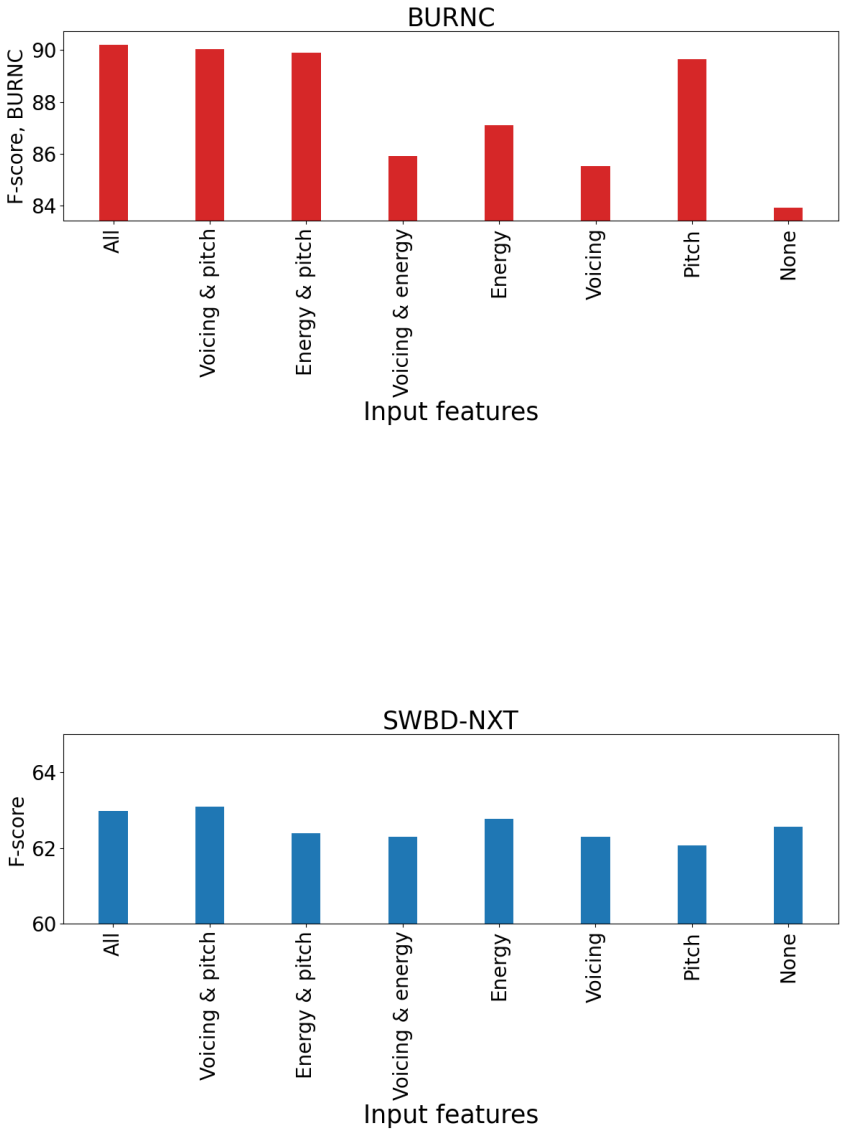


Figure 3.3: Ablation of prosodic inputs in the BURNC and SWBD-NXT prosody-only models, as evaluated on the development set via tenfold cross-validation. Labels on the x-axis indicate which features were available to the model. The model with 'all' features has access to voicing, pitch, and energy; the model with 'none' is the same as the duration-only baseline in Table 3.2. Note the differing scales for each model's F1 score.

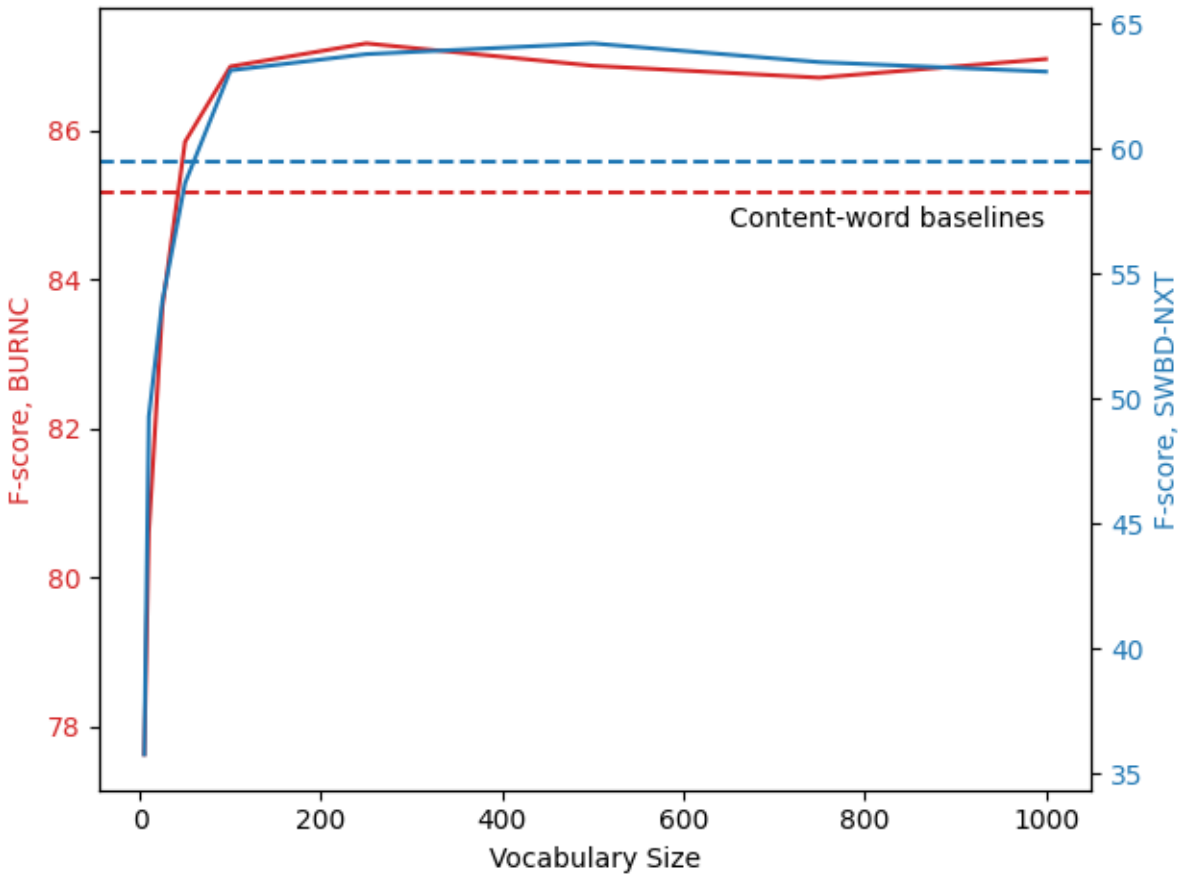


Figure 3.4: Performance of the text-only models with ablated vocabulary size, trained on BURNC (shown in red) and SWBD-NXT (shown in blue). The horizontal lines indicate the F1 score of the content-word baseline for each model. Note the differing scales for each model's F1 score, shown in the corresponding color.

100 most frequent words in the corpus. This means that with low vocabulary sizes, the only non-UNK words being presented to the model are stopwords. The stopword/non-stopword distinction is very correlated with the function/content word distinction, with most stopwords being function words. The fact that the model can perform so well with access only to stopwords suggests that either word frequency or the strongly correlated content/function word distinction are the main source of information for the text-only model.

3.5 Conclusion

This work demonstrates some important principles for detecting pitch accent from text and prosody. First, it shows that pitch accent location can be detected with relatively high accuracy from text alone, indicating overlap between the information in the text and prosodic channels. Second, a prosody-only model generally benefits from changes to the model that enhance access to context. This is true both for having longer segments of input context, and for having an LSTM that helps with processing longer utterances. Third, this work introduces two related baselines: a content-word baseline and a duration-only baseline. The high performance of both of these baselines shows that both the text and the prosody-only model derive at least some of their performance from being able to distinguish function words from content words. In fact, a large part of the text-only model's performance can be explained by the content-word baseline. This work also shows that a prosody-only model can successfully detect pitch accent in cases where a text-only model cannot, particularly in cases where the speaker's production differs from the content-word baseline. These results indicate that the prosody-only model uses information available in the prosodic inputs to surpass the content-word baseline, and that even when the text-only model has access to actual word identities it cannot perform much better. Finally, this work shows several effects of genre, most notably the greater difficulty of this task for spontaneous speech like SWBD-NXT, where pitch accents are fewer in number. Additionally, prosodic inputs are generally less helpful for SWBD-NXT, though this may be an artefact of poorer audio quality. These insights into genre are especially relevant in Chapters 5 and 6, where the primary resource used is SWBD-NXT.

Chapter 4

Word order in speech translation

4.1 Introduction

The previous chapter established that it is possible to detect prosodic events from a combination of text and prosodic inputs. In this and all following chapters, the task is not detecting the prosodic *events* themselves, but rather detecting the things that those prosodic events signal.

One important distinction that may be prosodically marked in English is the difference between *new* and *old* information. For example, in (1), the word *trespassers* is new in the discourse context, and so it is marked by a pitch accent (shown by bolding in Example (1)):

(1) Who will be prosecuted?
Trespassers will be prosecuted.
(Selkirk, 1995)

An entity is new the first time it's introduced into the discourse context, and subsequent mentions of it are examples of old information. This new/old distinction is a fundamental part of most accounts of information status (though these accounts can include many other categories as well).

While the new/old distinction is often prosodically marked in English, this same kind of information status can be conveyed by very different structures in different languages. In this chapter, Russian is used as an example of a language that expresses information status by different means, namely constituent order. In Russian, noun phrases that contain new information tend to come last in a sentence (Rodionova, 2001). For example, in ((2)) below, when the new information is that it is Masha

who loves Sasha, *Masha* appears last in the sentence. When the new information is that it is Sasha who Masha loves, *Sasha* is last.

(2) Context: 'Who did Masha love?'

Mash-a l'ub'ila Sash-u
Masha-NOM loved Sasha-ACC

'Masha loved Sasha.'

(3) Context: 'Who loved Sasha?'

Sash-u l'ub'ila Mash-a
Sasha-ACC loved Masha-NOM

'Masha loved Sasha.'
(Rodionova, 2001, 4)

In languages such as English, where major constituent order is fixed by syntactic rules, this kind of meaningful word-order variation isn't possible without recourse to changing the argument structure of the sentence, e.g., through passivization.

In theory, in a parallel English-Russian corpus with English speech available, there should be a correspondence between entities that are prosodically marked as new in English and entities that are marked as new by their sentence-final position in the Russian text. This means that an English-to-Russian speech-to-text translation system trained on this kind of corpus of parallel English speech and Russian text data should be able to learn this correspondence: Pitch-accented words in English are more likely to correspond to sentence-final words in Russian.

In this chapter, I describe an experiment aimed at determining whether an English-to-Russian speech translation (ST) system can learn the information status-motivated correspondence between English prosody and Russian word order. More specifically, can a speech translation system learn that a certain pitch accent pattern in English is best translated by a specific word order in Russian?

In order to answer this question, two English-Russian speech-to-text translation systems are trained, one with full access to prosodic information, one with limited prosodic information. The evaluation metric of interest is the comparative ability of these two systems to produce word orders that are closer to the gold target order. The hypothesis is that the model with more accessible prosodic information will have better

access to information status and therefore will produce translations with word orders that are on the whole closer to the gold target translations.

These experiments show that both the prosodically enriched and the baseline model produce output that very closely tracks with the source word order and rarely deviates from it for any reason, related to information status or otherwise. Because the effect of the source word order is so strong, including prosody in the output does not significantly impact the generated target word order.

4.2 Background

This section outlines the linguistic research on the connection between information status and both English prosody and Russian word order, which underpins this research.

Information status is broadly concerned with "the salience and organization of information in an utterance in relation to a discourse" (Calhoun, 2010, 1). This work is specifically concerned with the distinction between *new* and *given* information as made by Nissim et al. (2004), where new entities are those that haven't yet been mentioned in the discourse context, and given entities have been.

According to many linguist accounts of English prosody (e.g., Selkirk 1995, Bolinger 1972, Kruijff-Korbayová and Steedman 2003, and Pierrehumbert 1980), new items are marked by pitch accents in English. Other linguists (e.g., Calhoun 2010) propose a more complicated relationship between pitch accents and information status, but for this project, we assume the simpler account: New information is generally accompanied by a pitch accent, as in example (1) above.

As shown in (2), Russian can make use of other resources to mark newness. Because Russian word order isn't fixed by syntactic rules, newness is often signaled instead by putting the new information last in a sentence (Rodionova, 2001).

4.3 Method and model

In order to test the hypothesis that a speech translation model can learn a correspondence between English prosody and Russian word order, two English-to-Russian speech-to-text translation systems are trained and evaluated. The goal is to compare a baseline system with limited prosodic information to one that has access to more complex prosodic information. This is accomplished by giving both models standard mel filterbank features as input, while augmenting the prosodically enriched system with

additional features that directly reflect pitch, energy, and voicing. More detail on these features corresponding to acoustic correlates of prosody is given in Section 4.4 below. The baseline model still has some access to prosodic information — for example mel filterbank features convey pitch information and signals to duration. However, the prosodically enriched model has more direct and explicit access to pitch, energy, and voicing features. Finally, both systems are evaluated using BLEU score and permutation distance, as described in Section 3.3.5.

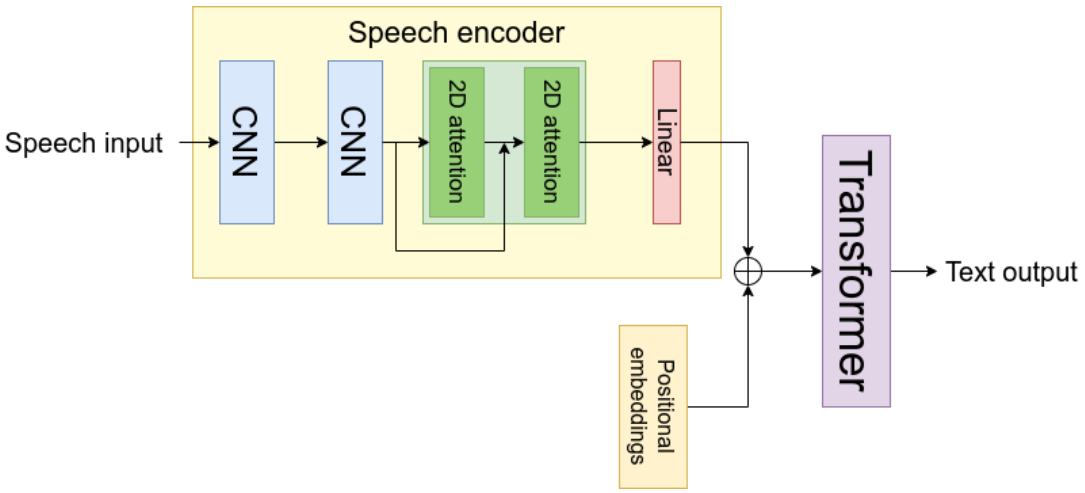


Figure 4.1: The architecture of the speech translation system.

Both models are architecturally identical, using the code from Gangi et al. (2019), shown in Figure 4.1. This model is based on a standard encoder-decoder transformer, with six encoder and six decoder layers. Gangi et al. (2019) add additional layers to the input side of the transformer for processing acoustic features: a series of two CNN layers, followed by two self-attention layers before the transformer. Following Gangi et al. (2019), both models are pretrained on an ASR task, using the English transcripts of the English-to-French section of the MUST-C corpus (which is the same corpus used for training the model; see Section 4.4 for more details). The pretraining lasts for 100 epochs, and the pretrained encoder weights are transferred to the final model. Finally, the model is trained for 100 epochs on the English-to-Russian section of MUST-C. The innovation added to this model is the incorporation of prosodic inputs, which are discussed in section 4.4.2 below. These prosodic inputs are simply concatenated to each frame of filterbank features.

All other hyperparameters are shown in Table 4.1.

Category	Hyperparameter	Value
Prosody encoder	CNN kernel size	3x3
	Num. CNN filters	16
	Dropout	0.1
Transformer	Encoder layers	6
	Decoder layers	6
	Attention heads	4
	Layer size	256
	Hidden size	768
	Attention dropout	0.1
	ReLU dropout	0.1
	Other dropout	0.3
Training	Training epochs	100
	Optimization method	Adam
	Learning rate	0.0002

Table 4.1: Hyperparameters used for the speech translation model, as specified by Gangi et al. (2019).

4.4 Data

The MUST-C corpus (Di Gangi et al., 2019) is used for model training and evaluation. This is a corpus of TED talks, with approximately 500 hours of English source audio transcribed and translated into eight Indo-European languages (German, Spanish, French, Italian, Dutch, Portuguese, Romanian, and Russian). The two sections used for this chapter are the English-Russian section of the corpus (489 hours), used for training and evaluating the models, and the English speech and transcripts from the English-French section (492 hours), used for ASR pretraining.

The translations in this corpus are produced by volunteer subtitlers for the TED organization.[1] Since they are generated as video subtitles, translators have access to the English audio and the speaker's prosody. However, the subtitler instructions suggest that subtitlers have access to an English transcription as well, which may reduce the amount of influence of the original audio and its prosody on the translation, depending on individual volunteers' workflows.

Each model is tuned and evaluated on the development set of the MUST-C corpus, and then evaluated on the test set. In addition, a small subset of the development set (*dev-small* henceforth) is selected, which only contains examples that show the

[1]The instructions for subtitlers, current to the creation of MUST-C, can be found here: `https://web.archive.org/web/20200318053836/https://www.ted.com/participate/translate/subtitling-resources#h2--tutorials`.

phenomenon of interest. In this case, that means examples where the order of content words in the English source is different from the order of content words in the gold Russian target for reasons of information status.

The following process is used to create dev-small: Using Fast Align (Dyer et al., 2013), word alignments are generated between the English and Russian development set data. The word alignment information is then used to identify the examples where at least one pair of content words[2] is permuted between the English and Russian. Finally, these examples are surveyed by hand to identify the examples that genuinely have content word reorderings between source and target which seem to be motivated by information status. Using this process, 77 of the total 1317 development set examples are selected.

Since dev-small is smaller than the optimal size required by Fast Align, the English-Russian bitext is supplemented with 3 million pairs of sentences from the United Nations Parallel Corpus (Ziemski et al., 2016), which are discarded once the alignments are generated. Fast Align is also used to generate Russian-Russian alignments for the evaluation metrics (see Section 3.3.5). When making Russian-Russian alignments, the data is supplemented with approximately 15 million Russian-Russian paraphrase pairs sampled from the ParaPhraserPlus corpus (Gudkov et al., 2020) by selecting pairs from sets of paraphrases. More sentence pairs are used in the Russian-Russian case because there is lower lexical diversity in this corpus, since many pairs are generated from the same set of paraphrases.

The *permutation distance* metric described in section 3.3.5 below is also used to verify that dev-small contains more examples with word order permutations between source and target than the rest of the dev set does. As can be seen in Table 4.2, the target side of dev-small examples has more word order differences with the source side than other dev set examples.

	Permutation distance
Dev small set	0.51
Other dev set	0.28

Table 4.2: Permutation distance between source and target for the dev-small set as compared to all other development set examples. This shows that dev-small on the whole has many more word order differences between source and target than other development set examples do.

[2]The content-function word distinction is approximated by simply considering all nouns, adjectives, adverbs, and verbs to be content words.

4.4.1 Representation of target phenomenon

Before training the speech translation models, it is important to verify that the target phenomenon is adequately represented in the training data to be learnable by the model. In other words, there must be enough bitext examples where there is a new entity in the English that is both marked by a pitch accent and not sentence-final, and a corresponding word in the Russian that *is* sentence-final. Example (4) shows three examples that show this kind of correspondence. The new, pitch accented information is shown in boldface type:

(4) a. EN: I'm in **Germany** in this [photo].

RU: Na etom foto ya v **Germanii**.
in this photo I in **Germany**

b. EN: Over time, people do what you **pay** them to do.

RU: So vremenem, lyudi delayut to, za chto vy im **platite**.
with time people do that for which you them **pay**

c. EN: [...] they want that to **live**, more than anything else.

RU: [...] bol'she vsego na svete im khochetsya, chtoby ikh
[...] more than-all on earth they want so-that their

tvorenie prodolzhalo **zhit'**.
creation continued **to.live**

d. EN: [...] a woman brought me this little bell, and I want to **end** on this note.

RU: [...] odna zhenschina prenesla mne malen'kii kolokol'chik,
[...] one woman brought me small bell

i vot na etoi note ya khochu **zakonchit'**
and thus on this note I want **to.end**

The training set contains approximately 265k training instances and automating this verification process is non-trivial. To make the search more tractable, the first step taken is to narrow the search to the 27k instances in the train set that contain dative pronouns, since sentences with datives are more likely to have at least three verbal arguments to permute. Finally approximately 350 of these instances are surveyed by hand to check if (1) they show word order differences between the source and target;

(2) the target word order differences can't be attributed to basic syntactic requirements of Russian; and (3) the utterance-final phrase in the Russian target corresponds to a pitch-accented phrase in the English source. Of the instances checked, 20 fit all three criteria. These 20 examples come from a total of 14 different speakers. There is no explicit metadata recording which translators worked on which TED talks, but since the translators are assigned one talk at a time, it is likely that the data from these 14 speakers also represents close to 14 translators. If these examples are representative of the subset of the training data that contains dative pronouns, there should be at least 1.5k representative instances in the training data; if these examples are representative of the whole training set, there might be up to 15k instances.

4.4.2 Input features

The baseline model takes 40-dimensional mel filterbank features extracted from the audio signal with a window size of 25 ms and step size of 10 ms, following Gangi et al. (2019). For the prosodically enriched model, six features for acoustic correlates of prosody are extracted at each timestep. These features are extracted from the audio using OpenSmile (Eyben et al., 2013), following Stehwien and Vu (2017). These features are: smoothed F0, RMS energy, loudness,[3] voicing probability, and harmonics-to-noise ratio. In order to input these features to the prosodically enriched model, they are concatenated with the 40 filterbank features at each timestep, so that each timestep has a 46-dimensional vector of input features. For more discussion of these prosodic inputs, see Chapter 3, where they are used in the pitch accent detection model.

4.5 Evaluation

Two separate metrics are used to evaluate the model output. BLEU score is used to judge overall translation quality, since it facilitates comparison with other work. However, the hypothesis motivating this work is about the word order of the model output, not its overall quality. Since BLEU measures n-gram overlap with the gold target, it does capture information about word order, but also reflects irrelevant things such as lexical differences. In order to abstract away from lexical choice, we also measure the *permutation distance* between the model output and the gold target, following Birch

[3] Specifically the "PCM loudness" feature in the OpenSmile library, defined in the documentation as "loudness as the normalised intensity raised to a power of 0.3."

	BLEU (↑)	**Perm. distance (↓)**
Baseline	**10.50**	**0.39**
w/ Prosody	10.06	0.41

Table 4.3: Results on the full development set for the baseline and prosodically enriched models. Both BLEU and permutation distance are calculated between the model output and the gold target. A lower permutation distance indicates better performance. The differences between the two model performances are verified to be significant ($p < 0.05$) using bootstrap resampling, with 1000 resamples of each model's output.

et al. (2010). Permutation distance is a measure of how different the ordering of two sequences is. A lower permutation distance is assigned to model output that more closely matches the gold target word order.

The process used to calculate permutation distance is illustrated in figure (4.2). First, the generated output and the gold target text are aligned using Fast Align (Dyer et al., 2013). These alignments are then used to abstract away from lexical choice: In both the output string and the target string, words that have been aligned to each other are replaced with a symbol. Once all the aligned words have been replaced by symbols, all non-aligned words are dropped. If there are any consecutive repeated symbols, these are collapsed to a single symbol (e.g., ABBCD → ABCD). If there are non-consecutive repeated symbols, all but the first repetition are discarded (e.g., ABACD → ABCD). The reasoning for this choice is that the model output sometimes repeats the same content several times (e.g., *John said hello. John said hello. John said hello.*), and it seems most sensible to compare the first of these repetitions to the target. Finally, the Hamming distance between the resulting sequences is reported (Birch et al., 2010).

4.6 Results and discussion

On the full development set, the prosodically enriched model *underperforms* the baseline model on both permutation distance and BLEU score (see Table 4.3).

However, on the small development set (dev-small), the prosodically enriched model outperforms the baseline model (see Table 4.4). In order to verify the permutation distance calculation is valid, a manual calculation of the metric is conducted on dev-small by aligning the model output with the gold target by hand and calculating permutation distance as before. The manual permutation distances confirm that the prosodically enriched model outperforms the baseline on dev-small.

These results raise two questions: (1) Why does prosodic information lower perfor-

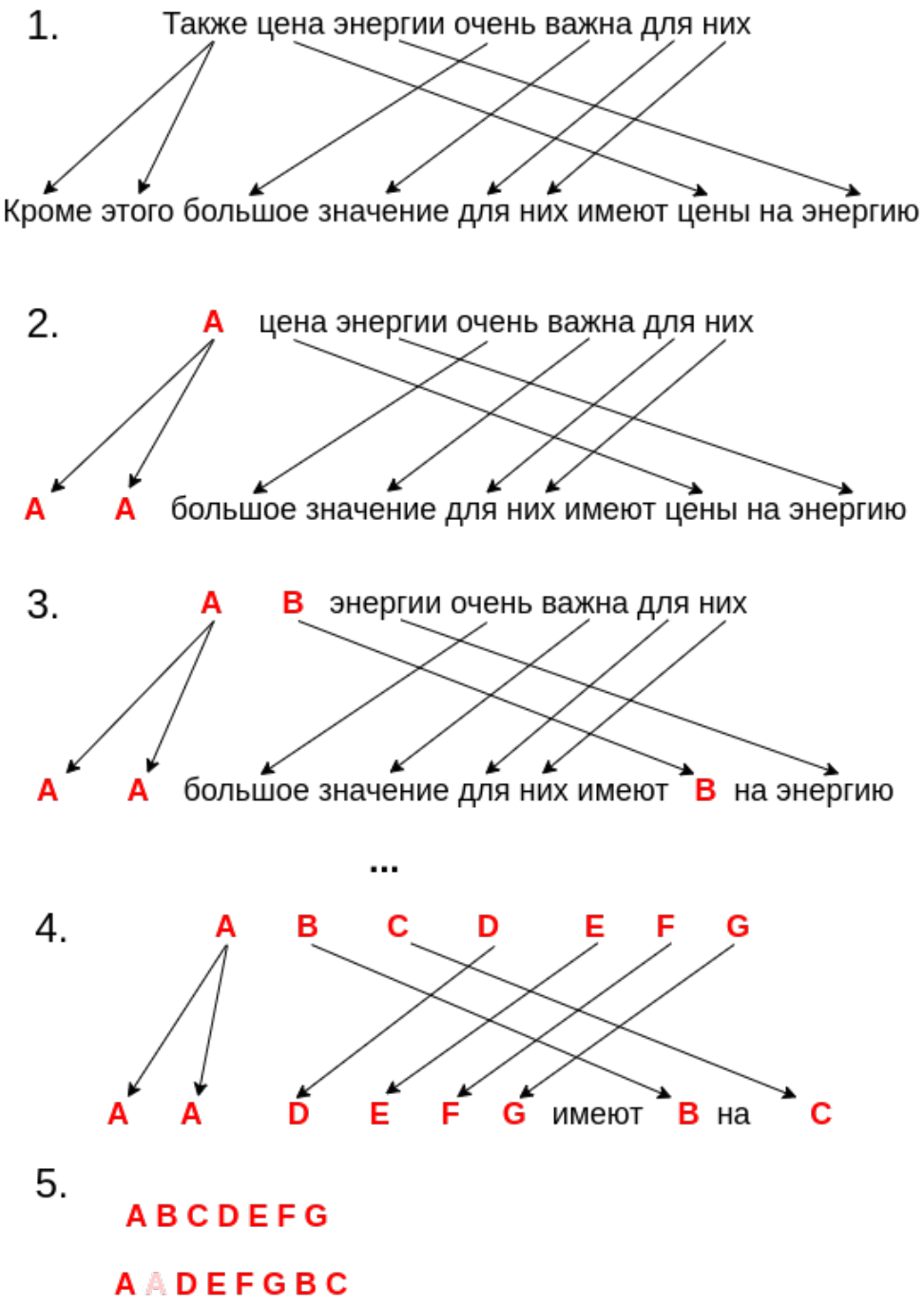


Figure 4.2: How permutation distance between model output and gold target is calculated using word alignments (represented with arrows). (1) Generate word alignments either by hand or using Fast Align; (2-4) Iterate through pairs of aligned words, replacing words with a symbol (e.g., A), which abstracts away from lexical choice. (5) Remove unaligned words and all but the first repetition of repeated words. Use the resulting sequences to compute Hamming distance.

	BLEU (↑)	Auto. perm. dist. (↓)	Manual perm. dist. ↓)
Baseline	8.63	0.52	0.43
w/ Prosody	**8.81**	**0.49**	**0.40**

Table 4.4: Results on the small development set (dev-small) for the baseline and prosodically enriched models. In addition to the automatic permutation distance metric we report on the full development set, we also report the permutation distance calculated with manual alignments. The differences between the two model performances are verified to be significant ($p < 0.05$) using bootstrap resampling, with 1000 resamples of each model's output.

	Perm. dist. btwn source and target (↓)
Baseline	0.07
w/ Prosody	0.08

Table 4.5: Permutation distance between model output and source on a subset of 25 examples of the development set. The low distance between model output and source shows that both models tend to simply reproduce source word order.

mance on the full development set? and (2) Why does prosodic information improve performance on the dev-small set? In both cases, analysis of the model output shows that for most examples where the permutation distance is different for the two models, it is because one or the other model's output is simply missing a word. This changes the permutation distance score, but not because either model is actually permuting more or less. This suggests that there are no actually meaningful word order differences between the two models, and instead this is an artifact of the metric.

The conclusion that the two models' output is not significantly different is supported by the fact that both models seem to closely reproduce the source word order. In fact, when the model outputs and the target instances are hand-aligned and their permutation distance computed for a small subset of the development set consisting of 25 instances,[4], both models are very close to the source word order, as shown in Table 4.5. It is unclear why the two models would have systematic differences in which words they drop, but this does seem to be the most significant source of differences in permutation distance between the two models' output, rather than true permutations.

In fact, it appears that the main reason there isn't much of a word order effect is that no amount of prosodic information prompts a model to deviate significantly from

[4]Hand-alignment is used for this calculation because as can be seen in Table 4.4 the automatic permutation distance calculation is consistently higher than the manual one, likely due to errors in the output of Fast Align. Using manual alignments shows how low the actual distance between model and source is. However, since hand-alignments are costly to perform, this limits the size of the set on which we perform this calculation.

Target		Perm. dist. btwn source and target	
		Full dev	Small dev
	Gold	0.29	0.51
Predicted	Baseline	0.28	0.27
	w/ Prosody	0.30	0.33

Table 4.6: Permutation distance between the English source and the gold or predicted Russian text.

the source word order. Table 4.6 shows the permutation distance between the English source text and the Russian target text. For the dev-small set, the distance between the predicted target Russian text and the source is *shorter* than the distance between the gold target text and the source. These results show that the model output follows the source word order too closely generally, when compared to the gold target text. This difference is not perceptible in the performance metrics of the full development set, but it's more pronounced in the dev-small set, which consists of examples with more differences between source and target.

Furthermore, addressing one of the major shortcomings of the experiment — the problem of using mel filterbank features for the baseline model — is unlikely to change the outcome of this experiment. The issue with mel filterbank features, as discussed in Section 4.3, is that they include some prosodic information, including pitch and duration information, meaning that the baseline model was not entirely without prosodic input. Potentially, replacing mel filterbank features with the 13-dimensional MFCCs could remove some of this pitch information, giving a clearer picture of the effect of prosody as opposed to other features. However, even the most prosodically informed model hews very closely to the source word order, suggesting that prosodic information alone is not enough to produce word order differences, no matter the baseline for comparison.

4.7 Future directions

This study suggests that, at least for the speech translation models used here, word order is not a good candidate for measuring the effect of prosodic information on speech translation performance. Whether or not these word order phenomena are robust enough to theoretically be learned by some other model remains unanswered.

One obvious opportunity for follow-up research is simply to update the speech translation model used. Though Gangi et al. (2019)'s model achieved state-of-the-art

performance on the MUST-C for speech translation models at the time it was first published, the BLEU scores it achieves on the English-Russian pair are quite low (consistently under 15). It could be that a higher-performing speech translation model would have less bias towards reproducing the source word order. There have been substantial developments in the field of speech translation since this model was developed. For example, Zhou et al. (2024) use Whisper (Radford et al., 2022) as their end-to-end speech translation system, which is a multi-task, multilingual speech model. Other recent models such as Communication et al. (2023) also take this multi-task, multilingual approach. It's not clear exactly how these models would compare to Gangi et al. (2019)'s performance specifically on English-to-Russian speech performance, since neither paper reports language-specific performance metrics. However, these models' massively multilingual approach makes them interesting candidates for future work.

However, another possible avenue for follow-up research would simply be to look for effects of English prosody on ST elsewhere, on phenomena other than than word order. For example, prosody can act as a signal to coreference resolution. A speech translation system with access to prosody in the input might be better able to correctly resolve coreference.

Directly evaluating the coreference ability of a speech translation system could be difficult; one possible way to do so would be Winograd-style challenge sets such as those described by Stanovsky et al. (2019). The idea behind this method is that for target languages with grammatical gender — a category which includes most of the languages represented in MUST-C — in some cases, the translation model has to assign grammatical gender to pronouns or nouns that are not gendered in the English source. The translation model's ability to correctly assign gender on a challenge set could serve as a metric for the model's coreference resolution performance. An example of a Winograd-style test instance is shown in the following example from Stanovsky et al. (2019), where the English words *doctor* and *nurse* have no grammatical gender, but the Spanish equivalents *doctor/doctora* and *enfermero/enfermera* do:

(5) English (source): **The doctor** asked **the nurse** to help her in the procedure.
Spanish (target): **La doctora** le pidio a **la enfermera** que le ayudara con el procedimiento.
(Stanovsky et al., 2019)

If the model correctly determines that *the doctor* is the antecedent of *her*, then it should translate *the doctor* as the feminine *la doctora*. However, if it doesn't do this resolu-

tion correctly, it may instead translate *the doctor* as *el doctor*. These Winograd-style schemata are devised by Stanovsky et al. (2019) to determine the effect of gender bias on translation and coreference performance, and so these exact examples might not be ideal for measuring the effect of prosody on coreference resolution, since the focus on cases where the model may have seen gender-biased data (e.g., a bias toward assuming doctors are men and nurses are women), which introduces a confound. However, this general approach of using gender as a way of measuring coreference quality has promise as a method for determining how important prosody is to coreference resolution in speech translation.

However, this experiment would require data beyond the existing data used here. Either naturally occurring Winograd-style examples must be extracted from existing speech corpora (which are unlikely to exist, given the many constraints these examples must satisfy), or recordings must be made of the Winograd examples, with care taken to elicit the target prosody correctly.

One other question for future research is the affect of ASR pretraining on the ability of the models to permute the word order between source and target. As a task, ASR doesn't involve reordering between source and target. While ASR pretraining applies only to the encoder, it may be the case that ASR predisposes the model towards not changing the order of the source.

Finally, the approach of Zhou et al. (2024) to testing the effect of prosody is potentially worth applying in future work. In order to see the effect of prosody, they simply compare end-to-end and pipeline speech translation models. Pipeline models only have prosodic information available at the ASR stage, while the information from prosody can percolate throughout an end-to-end model. Of course, there can be numerous other differences between end-to-end and pipeline models (as is amply demonstrated by experiments in Chapter 5), so this experimental set-up may introduce other sources of variation besides prosody. However, it does do a better job of depriving the prosody-free translation model of prosody at translation time, making it an intriguing approach for future work.

4.8 Conclusion

While there are linguistic reasons to think a speech translation model might be able to learn a correspondence between English prosody and Russian word order, in these experiments, the models' output closely reproduces the source word order regardless

of their input features. This makes it difficult to find any effect from adding prosodic inputs to the model.

This study is a good example of a case where it's reasonable to think that prosody might change a model's performance, but in practice, there is no measurable effect — in this case, due to model bias. In general, this shows that demonstrating an effect from prosody is not always as simple as selecting a task where prosodic information is relevant and including prosody in the model inputs. As shown in subsequent chapters, even in tasks where there is an established effect from including prosody, consideration has to be given to which parts of the task prosody helps with the most (e.g., sentence segmentation in Chapter 5), or the experimental conditions in which prosody is most important (e.g., under noisy conditions in Chapter 6).

Chapter 5

Joint parsing and segmentation with prosody

5.1 Introduction

Parsing is one area of NLP where the idea of incorporating prosody is relatively well-researched. Speech parsing poses unique difficulties — such as the presence of disfluencies — that have made progress more modest in this domain than in general text parsing. Prosody has promise as one way of helping with these increased difficulties. Some previous attempts at incorporating prosody into speech parsers have indeed shown that prosody can modestly improve parsing performance for a variety of different parsing models, ranging from the PCFG parse-rerank models of Kahn et al. (2004, 2005) and Hale et al. (2006), to the transformer-based models of Tran et al. (2018, 2019). As might be expected, in the analysis of these results, some of these researchers have shown that a lot of the benefit afforded by prosody is due to the help it provides with identifying disfluencies, rather than directly helping with finding syntactic boundaries, which are not reliably correlated with prosodic boundaries (Tran et al., 2018). Furthermore, as parsing models have improved, the contributions of prosody have declined somewhat, suggesting that as parsers more effectively exploit the information in the text, the benefits from prosody shrink (Jamshid Lou et al., 2019).

However, this previous research generally only considers parsing single sentences.[1]

[1]For simplicity, the term 'sentence' is used here, although in practice, not all utterances in casual speech (such as SWBD-NXT data) are syntactically complete sentences. The utterances that are demarcated in SWBD-NXT — and which these models are being trained to identify — are based on Meteer and Taylor (1995)'s definition of the 'slash-unit': "A slash-unit is maximally a sentence but can be a smaller unit. [...] Intuitively, slash-units below the sentence level correspond to those parts of the narrative which are not sentential but which the annotator interprets as complete."

This means that one significant potential benefit from prosody — providing cues to sentence boundaries — isn't investigated in most previous research.

The motivating hypothesis of this chapter is that prosodic inputs from the speech signal help with parsing speech that *isn't* segmented into sentences, by improving the parser's ability to find sentence boundaries. To test this hypothesis, entire dialog turns are input to a neural parser without sentence boundaries. These turns resemble the input a dialog agent would receive from a user. Two different modeling approaches are tested: an end-to-end model that jointly segments and parses input, and a pipeline model that first segments and then parses the input. To my knowledge, there hasn't been previous research on combining sentence segmentation and parsing into a single task.[2] Following Tran et al. (2018) and Tran et al. (2019), two experimental conditions are tested for each model to test the effects of prosody: inputting text features only, and inputting both text features and features for the acoustic correlates of prosody which are extracted directly from the audio signal. Additionally, this work follows Tran et al. (2018, 2019) in using the Switchboard corpus of English conversational dialogue (Calhoun et al., 2010).

The experiments in this chapter show that the primary hypothesis holds: When parsing speech that hasn't been segmented into sentences, parsers using both text and prosodic inputs are more accurate than those using text alone. Unsurprisingly, the end-to-end model performs parsing better than the pipeline model because it doesn't suffer from error propagation. The expectation was that gains in parsing quality would come primarily because models with access to prosody would perform sentence segmentation better. This wasn't borne out by these experiments — the best parsing model was not the model that was best at segmentation. While prosody does help all models improve their sentence segmentation, the pipeline model is both better at segmentation than the end-to-end model *and* worse at parsing. Section 5.6 discusses why segmentation and parsing quality do not always correlate in this task. However, even though the best parses and segmentations are not always produced by the same model, all models perform better at both tasks with prosodic information.

The primary contributions of this chapter are:

- Building an end-to-end model that jointly performs sentence segmentation and parsing.

[2] The closest example is Kahn and Ostendorf (2012), which does both segmentation and parsing, but only in pipeline, not joint fashion, and using a parse-rerank model.

- Showing that prosodic inputs are helpful for both sentence segmentation and parsing, whether using an end-to-end or pipeline model.
- Showing that an end-to-end model performs parsing better than a pipeline model, specifically because the end-to-end model is able to model sentence boundaries jointly with other constituent boundaries.
- Demonstrating that a model's parse and segmentation performance are not always correlated. In fact, in some cases, performing well at one task can actively worsen performance on the other.

5.2 Background: prosody and syntax

Prosodic signals divide speech into prosodic phrases (Pierrehumbert, 1980). The location and type of these prosodic phrases are influenced by a number of linguistic considerations, including information structure (Steedman, 2000), disfluencies (Shriberg, 2001), and to some extent, syntax (Cutler et al., 1997). Some psycholinguistic research shows that in experimental conditions, speakers can use prosody to predict certain kinds of syntactic distinctions. For example, as discussed in Chapter 2, Price et al. (1991) and Wightman et al. (1991) demonstrated that in a laboratory setting, speakers of English could distinguish between two possible prepositional phrase attachments based on prosodic cues alone. However, Cutler et al. (1997) argue that outside of laboratory settings, English speakers often "fail to exploit" this prosodic information even when it is present, so it isn't actually a particularly helpful signal for syntax in practice. The fact that incorporating prosody into syntactic parsers in past experiments has led to limited benefits (see, e.g., Noeth et al. 2000; Gregory et al. 2004; Kahn et al. 2005; Tran et al. 2018) may be due to the fact that in practice, speakers don't actually consistently use prosodic cues for this kind of syntactic disambiguation.

Another reason for prosody's limited benefit may be that units below the sentence level don't always coincide with traditional syntactic constituents (Selkirk, 1995, 1984). In fact, the only prosodic boundaries that consistently coincide with syntactic boundaries are intonational phrase boundaries (see Section 2.1.2 of Chapter 2), which generally fall at the ends of sentences (Wagner and Watson, 2010). These intonational phrase boundaries are more distinctive than other prosodic phrase boundaries, with longer pauses and more distinctive pitch and energy variations, making prosody a reliable signal for sentence boundaries, but less so for lower-level syntactic structure.

The idea of using prosody as a signal for sentence boundary detection is not a new one. Examples of sentence segmentation models that use prosody include Gotoh and Renals (2000); Kolář et al. (2006); Kahn et al. (2004); Kahn and Ostendorf (2012), who all used traditional statistical models (e.g., HMMs, finite state machines, and decision trees), and Xu et al. (2014), who used a neural model. Kahn et al. (2004) and Kahn and Ostendorf (2012)[3] also looked at downstream parsing accuracy on the same corpus we use. They show that not having access to gold sentence boundaries decreases the parse score (measured by the SParseval metric) from 82.3 to 78.5, confirming that parsing without gold sentence boundaries is indeed a more difficult task, as expected.

5.3 Task

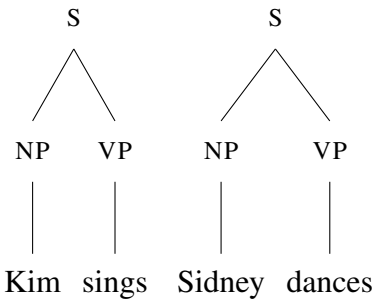
In order to test the hypothesis that prosody is helpful for the combined tasks of sentence segmentation and parsing, we propose a model designed to parse whole dialog turns, rather than just one sentence at a time. The parser follows the general design of Tran et al. (2019), but each turn that it parses must be also divided into one or more constituent sentences. The task is framed in two different ways: an end-to-end model (sentence segmentation and parsing done jointly) and a pipeline model (sentence segmentation done before parsing).

Both models return constituency parses for each turn in the form of Penn Treebank (PTB)-style trees. In order to keep the output in the form of valid PTB trees for the end-to-end-model, a top-level constituent, labelled TURN, is added to all turns, however many sentences they consist of. As discussed in Section 5.7, this innovation allows the end-to-end model to treat sentences in the same way that it treats other syntactic units. Figure 5.1 shows how the two sentences in 5.1a would be fused into a single turn in 5.1b.

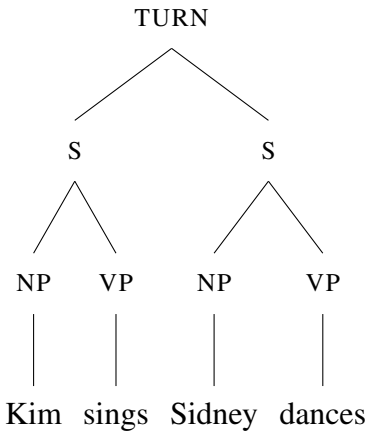
The pipeline approach consists of a segmenter step, followed by a parser. While the end-to-end model is trained on gold parses, the parser step of the pipeline model is trained to parse the sentences as predicted by segmenter predicted on the train set. This allows the model to learn to produce the parses on imperfectly segmented sentences and leads to better parsing scores.

The model can take inputs in the form of either text inputs only, prosodic inputs

[3]Like us, Kahn and Ostendorf (2012) don't use gold sentence boundaries, but direct comparison is impossible because they use ASR output instead of human transcriptions and a different metric for parse performance (SParseval; Roark et al. (2006)).



(a) Separate sentences:
(S (NP Kim) (VP sings))
(S (NP Sidney) (VP dances))



(b) Merged into a single turn:
(TURN (S (NP Kim) (VP sings)) (S (NP Sidney) (VP dances)))

Figure 5.1: For the end-to-end model, turns are represented as shown in Subfigure 5.1b: single sentences are joined into a single turn by a top TURN node.

only, or a combination of the two. The handling of these types of features is covered in Section 5.5.

5.4 Experimental set-up

5.4.1 Data

Using the American English Switchboard NXT corpus (henceforth SWBD-NXT; Calhoun et al. (2010)) for training and evaluation allows for direct comparison of performance with Tran et al. (2018) and Tran et al. (2019). This is a relatively small corpus compared to many datasets used today, but it remains the largest speech corpus with hand-annotated constituency parses. While other corpora with hand-annotated *dependency* parses exist (e.g., UD corpora such as Wong et al. (2017) and Dobrovoljc and Nivre (2016)), these are all significantly smaller than Switchboard NXT. SWBD-NXT comprises 642 telephone dialogues between strangers, totalling 55k dialog turns. For training, development, and testing, the corpus is divided into the split described in Charniak and Johnson (2001), which is a standard split for experiments on SWBD-NXT (e.g., Kahn et al. (2005); Tran et al. (2018)). The training set makes up almost 90 percent of the data (49k turns), and the development and testing sets make up slightly more than 5 percent each (3k turns). These dialogues are transcribed and hand-annotated with Penn Treebank-style constituency parses, and have no punctuation.

Not all turns in the SWBD-NXT contain multiple sentences: Of a total 55k turns, 35.8k consist of a single sentence. The average number of sentences per turn is 1.82. To avoid memory problems from too-long inputs, two problematically long turns are filtered out from the training set (out of 49k turns). No turns need to be removed from the development or test sets. This leaves the maximum turn length at 270 word tokens. Additionally, a small number of turns for which some or all of the audio is missing are removed.

5.4.1.1 Processing SWBD-NXT annotations

The SWBD-NXT corpus consists of recorded speech, corresponding text transcripts, and multiple layers of annotation for features such as prosodic events, syntactic structure, and information status. The annotation layers are arranged hierarchically. At the lowest level are *phonwords*, or phonological words, which are matched to the speech

recording via timestamps. Higher-level annotations, such as the pitch accents layer, don't reference timestamps in the audio directly, but instead reference *phonwords*. This allows the *phonwords* layer to be a single source of truth for how words match up to the speech.

For this task, the information needed from these annotations is (1) timestamps aligning words to the speech, and (2) the parse trees in the *syntax* annotation layer. However, the *syntax* tier doesn't directly reference the *phonwords* tier. Instead, the parse trees reference a *terminals* layer, where terminals are syntactic words, rather than phonological words. There are a few differences between the analyses the SWBD-NXT annotators use for syntactic and phonological words — for example, a contraction such as *don't* is analyzed as two syntactic words (*do* and *n't*), but a single phonological word (*don't*). The *terminals* annotation layer maps each terminal to a *phonword*, but the mapping is not always one-to-one. This complicates the process of mapping syntactic words to the audio.

For comparability with Tran et al. (2018) and Tran et al. (2019), their basic approach to this problem is applied here: When two *terminals* map to the same *phonword*, the *phonword's* timestamps are used for both *terminals*. So, for the example of *don't* $\leftrightarrow$ *do n't*, both *do* and *n't* have the same start and end time stamp. This has the obvious shortcoming of leading to the same prosodic information being input twice, once for each *terminal* word. However, the alternative is to guess at a reasonable timestamp to divide the terminals at, which has similar downsides, while also being more complex and introducing new possible sources of errors.

5.4.2 Features for acoustic correlates of prosody

Acoustic features for pitch, energy, pause duration, and word duration are extracted from the audio, largely following Tran et al. (2018)'s feature extraction procedure, noting any devations below. The alignment of word tokens to audio is annotated in the Switchboard NXT corpus. Pitch and energy features are extracted from the speech signal with Kaldi (Povey et al., 2011), using 25ms frames every 10ms. These include three pitch features: warped Normalized Cross Correlation Function (NCCF); log-pitch with mean subtraction over a 1.5-second window, weighted by Probability of Voicing (POV); and the estimated derivative of the log pitch. For further details on these features, see Ghahremani et al. (2014). It's worth noting that pitch estimation algorithms can be unreliable for telephone recordings of the sort in SWBD-NXT, since

the fundamental frequency may be missing, and the signal may be low relative to the amount of noise present (Wang and Seneff, 2000). To establish that these pitch features are reliable and useful, we conduct ablation tests that show that the pitch features boost model performance even when all other prosodic information sources are omitted 5.6.

For energy features, starting with 40-dimensional mel-frequency filterbank features, the log of the total energy is calculated, normalized by the maximum total energy for the speaker over the course of the dialog. Additionally, this value is calculated for the upper half and lower halves of the 40 mel-frequency bands, resulting in three total energy features.

Pause and word duration features are based on word token timestamp annotations. Each word's pause feature corresponds to the pause that follows it, which we categorize into one of six bins by length in seconds: $p > 1$, $0.2 < p \leq 1$, $0.05 < p \leq 0.2$, $0 < p \leq 0.05$, $p \leq 0$ (see below), and pauses where time-aligned data is missing. Following Tran et al. (2018), the model learns 32-dimensional embeddings for each pause category. Unlike Tran et al. (2018), pauses are calculated based on all words in the transcript, not just the words for a single speaker at a time. This means that at a turn boundary, the pause is calculated as the time between the end of one speaker's turn and the beginning of the other speaker's turn. This introduces negative-valued pauses, where one speaker interrupts the other. These negative-valued pauses are placed in the same bin as pauses with length 0.

Word duration features are normalized, since the quantity of interest is the relative lengthening or shortening of words, rather than their absolute duration. Following the code base for Tran et al. (2018), two different types of normalization are used: normalizing each word's duration by the mean duration of every occurrence of that word type; and normalizing the word's duration by the maximum duration of any word in the input unit.

One other set of features that could be argued to be useful are voice quality features (e.g., harmonics-to-noise ratio, zero-crossing rate), since creaky voice has been observed to mark sentence boundaries in some varieties of American English (Wolk et al., 2011). However, including these voice quality features led to no additional benefit, so they are omitted in the model reported on here.

5.4.3 Training

Unless stated otherwise, each model is trained with five random seeds, and the reported performance is the mean of each metric over all five seeds. To determine the statistical significance of differences in performance, bootstrap resampling is conducted (Efron and Tibshirani, 1994), resampling 10^5 times. With the randomly reseeded models, sampling is done randomly over all the models, as well as sampling over all the examples in the development set.

These models use the hyperparameters specified in Tran et al. (2019)'s code base, as documented in Table 5.1. Each model is trained for 50 epochs on a single Nvidia GTX 1080 GPU, which takes approximately 7 hours per model. The text-only models have approximately 23M trainable parameters each, while the text+prosody models have approximately 20M trainable parameters.

5.4.4 Evaluation

Two metrics are reported for both the pipeline and end-to-end models: parse and sentence segmentation F1 scores. Parse F1 is calculated on the whole turn using a Python implementation of EVALB.[4] The TURN constituents are omitted from this parse score calculation, so that turn-based and sentence-based parse scores are comparable. The sentence segmentation F1 score is calculated on all turn-medial sentence boundaries; turn-final sentence boundaries are not counted. In order to calculate the sentence segmentation F1 score for the end-to-end model, every node that is a direct child of the tree's top TURN node is counted as a sentence. That is, sentences are just one kind of syntactic constituent, differentiated only by their location in the tree. For example, in Figure 5.5a, both the INTJ node and the S nodes are counted as sentences, and the boundaries after the INTJ node and the first S node contribute to the segmentation F1 score.

5.5 Model

Both the end-to-end model and the pipeline parser are based on Tran et al. (2019)'s parser, extending the code base described in their paper.[5] The parser is a neural con-

[4] https://github.com/ekayen/PYEVALB

[5] Original: https://github.com/trangham283/prosody_nlp; the extended code for this project: https://github.com/ekayen/prosody_nlp.

Hyperparameter	Value
Epochs	50
Text embedding dim.	300
Max. seq. length	270
Dropout	0.3
Num. layers	4
Num. heads	8
Model dim.	1536
Key/value dim.	96
Num. CNN filters	32
CNN filter widths	5, 10, 25, 50

Table 5.1: Model hyperparameters, based on Tran et al. (2019).

stituency parser based on Kitaev and Klein (2018)'s text parser, with a transformer-based encoder and a chart-style decoder based on Stern et al. (2017) and Gaddy et al. (2018). The model has been augmented to take both prosodic and text inputs. This section describes the three main model components: the CNN that processes the continuous prosodic inputs before they reach the encoder (§5.5.1), the transformer-based encoder (§5.5.2), and the chart-style decoder (§5.5.3). A simplified diagram of the model is shown in Figure 5.2.

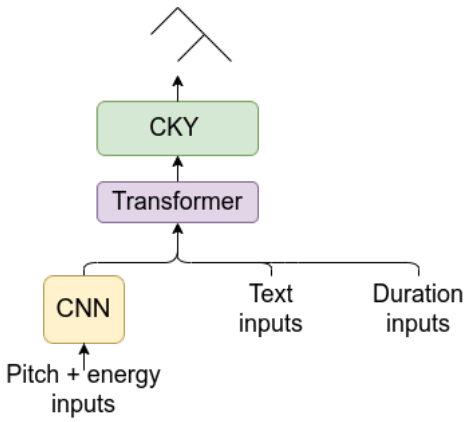


Figure 5.2: A schematic diagram of the parser.

5.5.1 The prosody-processing CNN

Of the four types of prosodic inputs, pause and duration are already discrete at the word level. Pitch and energy, however, are extracted from frames every 10 ms in the original speech signal. If a given word token is shorter than a fixed number of frames (set at 100 frames, following Tran et al. (2019)), some frames of left and right context

are included; frames from longer word tokens are subsampled to reduce their frame length. These two frame-based features have a different dimensionality than the word-level input and they are untenably long for a sequence model or transformer. The CNN solves both these problems by producing a fixed-length representation for each feature at the word token level. This representation can be concatenated with the other word-level features and input to the encoder.

For a speech input with f frames, the features input to the CNN have dimensions $6 \times f$, where 6 is the number of total features for each frame (3 pitch features and 3 energy features). Several filters of different sizes then perform one-dimensional convolution of the input. These different filters allow the CNN to integrate information on various time scales. N of each of these m filters are applied, for a total of $m \cdot N$ filters. The hyperparameters described by Tran et al. (2018) are used here: N = 32 filters of widths w = [5, 10, 25, 50], for a total of $m \cdot N$ = 128 filters. The output of each filter is then max-pooled, which converts the features for a given word to a uniform dimension.

These CNN-processed features are then concatenated with the prosodic inputs that are already word-level (pause and duration) and the text embedding for the word, and then input to the encoder. The CNN is trained along with the encoder-decoder model.

5.5.2 The encoder

The encoder is a standard transformer with eight attention heads, based on the work of Kitaev and Klein (2018). For each word of input x_i, the transformer encoder produces a representation of the forward context, $\overrightarrow{y_i}$, and the backward context $\overleftarrow{y_i}$. A given span between indices i and j is represented by subtracting the forward representations and backward representations and concatenating the results:

$$v_{(i,j)} = [\overrightarrow{y_j} - \overrightarrow{y_i}; \overleftarrow{y_j} - \overleftarrow{y_i}]$$

The next section explains how this span representation $v_{(i,j)}$ is used to generate scores for constituents in a tree.

5.5.3 The decoder

The decoder is a chart-style span-based decoder. Its goal is to output the correct tree T for an input $x_1, ..., x_n$. Each tree's score $S(T)$ is simply the sum of the scores of its constituents, where each constituent is defined by a start index i, an end index j, and a

label l.

$$S_{tree}(T) = \sum_{i,j,label \in T} S_{label}(i,j,l) + S_{span}(i,j)$$

As this formula for tree score shows, each constituent's score is made up of a label score and span score. Conceptually, the span score corresponds to the probability that a constituent exists that exactly covers span (i,j) in the input; the label score reflects the probability that the span (i,j) has a given constituent label (e.g., S, NP). The decoder must have a way of determining the label score and span score for each constituent.

The label scores are generated by passing the span representation $v_{(i,j)}$ through a two-layer feed-forward network like the feed-forward networks Vaswani et al. (2017) use:

$$FFN(x) = W_2(relu(W_1 x + b_1)) + b_2$$

Following Kitaev and Klein (2018), a layer normalization step (*LNorm*) is also included. This feed-forward network produces a vector for each span $S_{label}(i,j)$ whose size is the number of possible labels:

$$S_{label}(i,j) = M_2(relu(LNorm(M_1 v_{(i,j)}) + c_1)) + c_2$$

The lth element of this vector is the score for the label l:

$$S_{label}(i,j,l) = [S_{label}(i,j,)]_l$$

The span score is also needed for the evaluation of each constituent, but calculating the score for all spans (i,j) would be prohibitively inefficient. Instead, Kitaev and Klein (2018), following the approach of Stern et al. (2017) and Gaddy et al. (2018), use a dynamic programming strategy based on the CKY algorithm. The score for a span (i,j) is calculated in terms of the scores of its subspans, which allows span scores to be built up recursively from the stored scores of smaller spans. A given span (i,j) can be split at any internal point into two subspans, (i,k) and (k,j). Each of these possible splits (i,k,j) is assigned a score, calculated by summing the span scores of the subspans:

$$S_{split}(i,k,j) = S_{span}(i,k) + S_{span}(k,j)$$

Then, to find the best score for this span (i,j), the label and split are found that maxi-

mize the following sum:

$$S_{best}(i,j) = \max_{l,k}[S_{label}(i,j,l) + S_{split}(i,k,j)]$$

All spans are recursively split into subspans, eventually arriving at single-word spans. Since there are no splits possible for a single-word span, the score for a single word span is simply that word's best label score:

$$S_{best}(i,i+1) = \max_{l}[S_{label}(i,i+1,l)]$$

This method requires that the grammar be in Chomsky-Normal form, which the model achieves by collapsing strings of unary rules and using dummy nodes to make n-ary rules into binary rules.

With this method of generating tree scores from span representations, it's possible to calculate the hinge loss for the predicted tree $\hat{T}$ compared to the gold tree $T*$, where Δ represents the Hamming loss on labeled spans:

$$Loss(\hat{T}, T*) = \max[0, \max_{T}[\Delta(\hat{T}, T*) + S_{tree}(\hat{T})] - S_{tree}(T*)]$$

This loss function is then used to train the encoder-decoder, including the CNN input module for prosody.

The segmentation model in the pipeline is a modified version of the model. It uses the same encoder architecture, but the decoder is replaced by a single output layer that performs sequence labeling, marking words as either sentence-internal or sentence-final.

5.6 Results and discussion

Experiments with the end-to-end model show that prosody has a statistically significant effect on parsing F1 score, though this effect is small (see Table 5.2). In the pipeline model, the effect of prosody is larger: The difference in parse F1 score from adding prosody is 0.94 for the pipeline model, where it is only 0.52 for the end-to-end model. However, the pipeline model's parse F1 score is lower than the end-to-end model's.

Interestingly, the end-to-end model is simultaneously *better* at parsing and *worse* at segmentation. In Section 5.6.1, I discuss attempts to raise the end-to-end model's segmentation performance, and in Section 5.6.2, I cover attempts to raise the pipeline

	Gold sentences	E2E	Pipeline
Dev. set:			
Text only	90.31	85.70	84.34
Text+prosody	90.90	86.21	85.28
Test set:			
Text only	90.29	86.03	84.68
Text+prosody	90.65	86.55	85.62

Table 5.2: Development and test set parsing F1 score of the end-to-end and pipeline models (and for comparison, a model that receives gold standard sentences as input). Results averaged over 5 random seeds.

		E2E	Pipeline
Dev. set:			
Text only	F1	66.32	63.74
Text+prosody	F1	72.95	77.38
	Prec	69.46	79.44
	Rec	76.92	75.69
Test set:			
Text only	F1	71.01	66.98
Text+prosody	F1	72.94	77.38

Table 5.3: Segmentation F1 score of the turn-based models compared to the sentence-based model, with precision and recall given for some cases, averaged over 5 random seeds.

model's parse performance. However, in spite of these efforts — some of them successful — the parse and segmentation scores remain inconsistent. Section 5.6.3 covers how this is possible and why it arises from the framing of this task.

To evaluate the effect of different the various acoustic correlates of prosody, an ablation test is conducted where only one type of prosodic input is used at a time. The results are shown in the graph in Figure 5.6.

Across both tasks, the end-to-end model benefits more from pitch and energy features than from word or pause duration features. The effect size is very small for parsing, and could be unreliable, but the effects are larger for segmentation, and follow the same pattern. This match in performance between the two tasks makes sense, since the end-to-end model is doing both jointly. For the pipeline model, the effects of each feature class are less consistent across the two tasks, which is likely because the two tasks are done by separate models. With the small effect sizes in the pipeline's parse scores, it's hard to draw any strong conclusions about the relative importance of different prosodic inputs for this task in the pipeline condition. However, the effect sizes for segmentation are much more pronounced, and suggest that energy is a much less helpful feature than word duration, pause duration, or pitch. This is likely because English sentence boundaries are generally marked by pitch excursions, word lengthening, and longer pauses (Shriberg, 2001).

5.6.1 Improving the end-to-end model

Given the generally low segmentation performance from the end-to-end model, one additional modification is made to it to try to improve its segmentation. Poor segmentation performance is often caused when the end-to-end model predicts many more children for the top node than it ought. Since each direct child of the top TURN node is counted towards the segmentation metric as a sentence predicted by the end-to-end model, these highly branching nodes lower segmentation scores. The parser actually creates these multiply branching nodes by predicting a series of binary branching DUMMY nodes, which are collapsed in post-processing into a single n-ary node. In order to discourage oversegmentation, it is a possible that weighting these DUMMY nodes with a greater loss penalty will reduce the model's tendency to oversegment. Out of several possible penalty weights, a weight of 0.5 for each DUMMY node led to the best improvement parse and sentence segmentation quality on the development set. While weights higher than 0.5 continue to improve segmentation quality, particularly

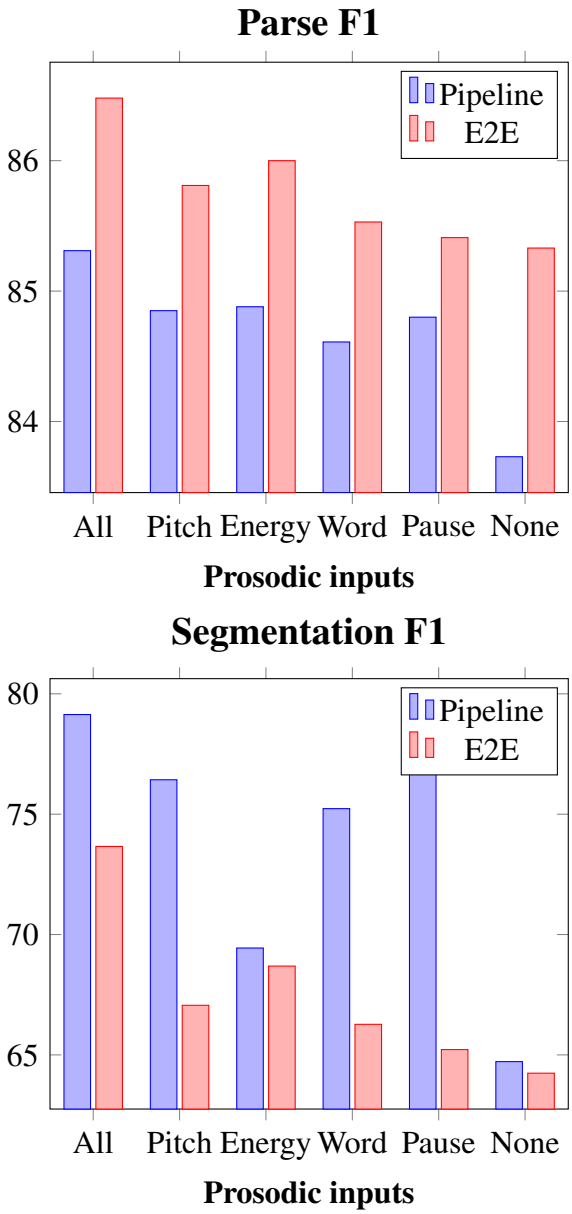


Figure 5.3: Parse and segmentation performance with ablation of prosodic inputs. The labels along the x-axis indicate which prosodic input features the model has access to.

segmentation precision, as expected, they begin to harm parse quality. Table 5.4 shows how different penalties affected the overall performance.

		Segmentation			Parse
	Penalty	**Precision**	**Recall**	**F1**	**F1**
Text	**0**	55.01	75.78	63.74	86.09
	0.5	59.66	69.59	64.24	**85.33**
	1	70.46	64.03	67.09	84.74
Text+prosody	**0**	54.79	77.59	64.22	86.47
	0.5	63.60	75.12	68.88	**86.48**
	1	73.10	69.67	71.35	86.00

Table 5.4: Development set sentence segmentation precision, recall, and F1 scores, and parse F1 for the end-to-end model, given different penalties for the dummy node. These results are reported for one random seed only, instead of five.

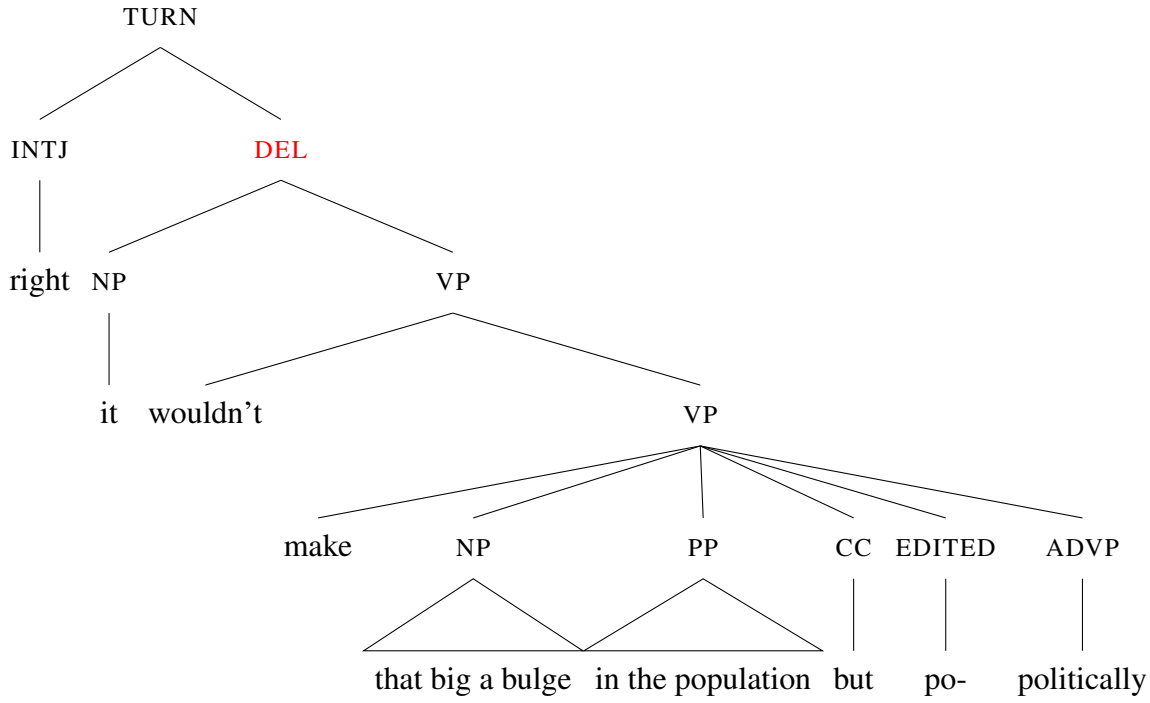
Since this penalty does produce improvements in segmentation performance, it is included in all models for which results are reported for the end-to-end model.

5.6.2 Improving the pipeline model

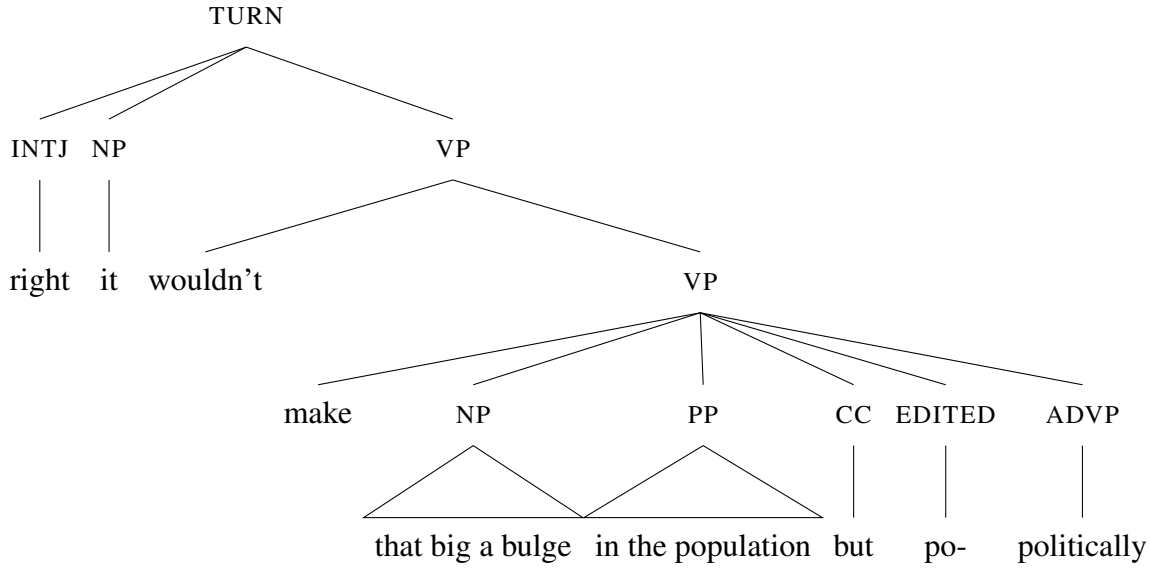
The pipeline model's weakness is a poor parsing score, in spite of a good segmentation score. This seems to be caused at least in part by error propagation from the segmentation to the parsing step: If the segmentation step predicts a sentence boundary, the parser has to predict a sentence dominating the corresponding span of the input. Any errors in the segmentation will cause problems in the parse (as discussed in greater detail in §5.6.3 below).

In an effort to reduce the effects of error propagation, a post-processing step was performed on the trees produced by the pipeline parser in order to give the model the option of *not* predicting a syntactic node dominating every predicted sentence (e.g., the S node shown in Figure 5.5b). In other words, this would give the pipeline model the option of connecting lower-level nodes directly to the top-level TURN node, just like the end-to-end model does. This is done by running the pipeline model through the segmentation stage as normal, but modifying the parsing step so that the model had the option of predicting a DEL node, indicating a node that should be removed from the final parse. Figure 5.4 shows this procedure, as it might apply to the example shown in 5.5. In this case, the procedure would result in a higher parse score, since no incorrect syntactic node would be predicted for the span dominated by the DEL node.

In order to train the parser to predict these DEL nodes, the training data must contain



(a) Predicted parse with DEL node.



(b) DEL node has been deleted.

Figure 5.4: The procedure used to reduce error propagation in the pipeline model. The parser is trained to produce DEL nodes dominating spans that should not actually be grouped as a syntactic constituent. These DEL nodes are removed in a post-processing step, as shown in 5.4b. In this example, the parse score would improve because there would be no more penalty for predicting a constituent for the span dominated by the DEL node in 5.4a.

them. This training data is generated by using the sentence segmentations predicted by the first step, and combining them with the gold parses. When these segmentations conflict with the gold parses in a way that results in an sentence that isn't a fully-connected tree, a node labeled DEL is inserted into the tree dominating that sentence. This allows the parser to learn to predict a DEL node in cases where an sentence shouldn't have a top node. When this parser predicts a DEL node, it is removed in post-processing, and its child nodes are connected directly to its parent node. This essentially allows the parser to fail to predict a sentence where the segmentation step predicts one.

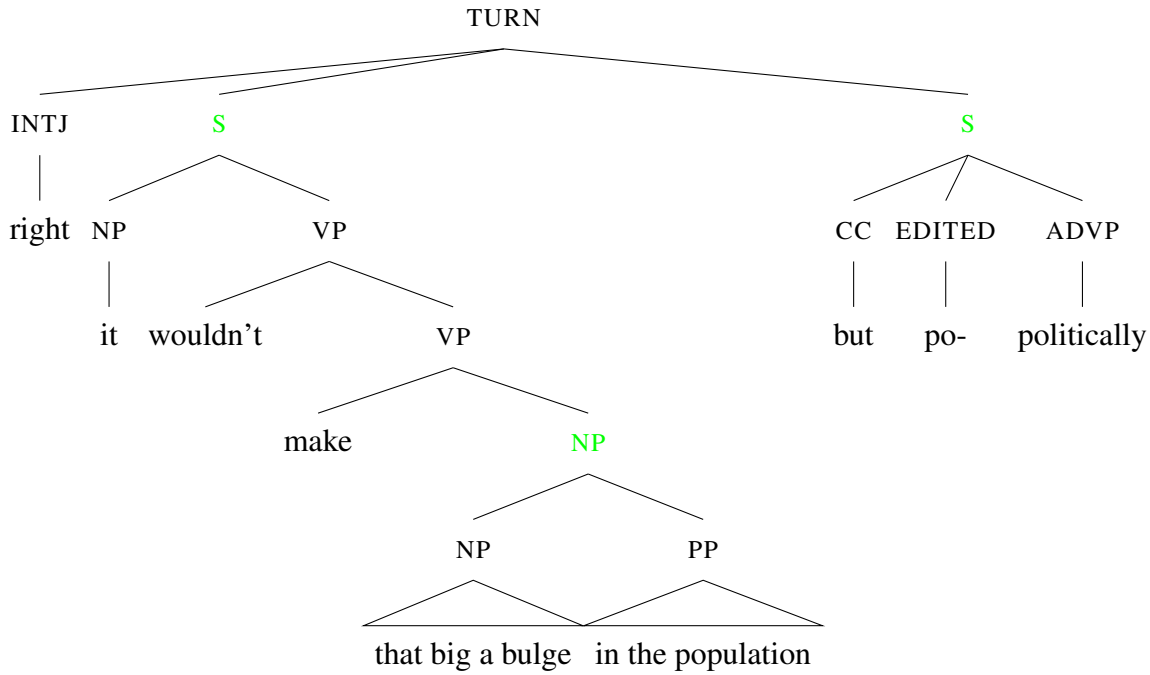
Unfortunately, the effect of this post-processing step on the overall scores is negligible, suggesting that even when the pipeline model doesn't have to predict a node dominating each predicted sentence, it tends to do so. Accordingly, these modifications are not included in the models whose results are reported here.

5.6.3 Inconsistency of parse and segmentation scores

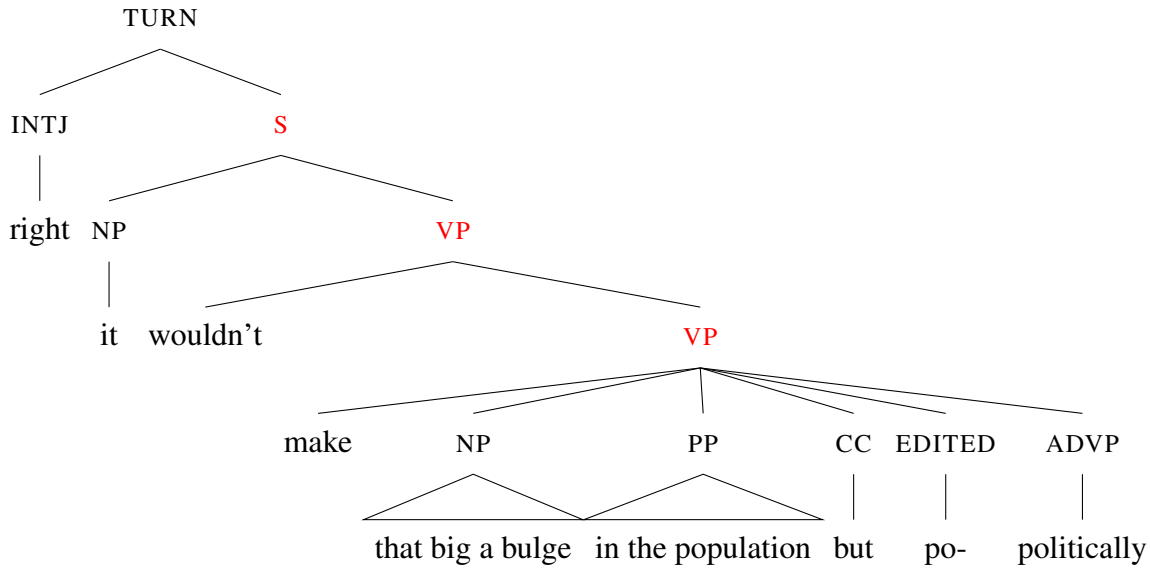
However, in spite of these modifications, the end-to-end model underperforms the pipeline model on segmentation, which is surprising, given that it parses better than the pipeline model. The end-to-end model's higher parse performance is simple enough to explain in isolation: In the pipeline model, errors propagate from the segmentation step to the parsing step; the end-to-end model has no opportunity for error propagation. But the end-to-end model's much lower segmentation score complicates the error propagation account.

In order to explain this phenomenon, it is helpful to further examine the kinds of errors each model makes. First, the end-to-end model tends to err by not predicting top S nodes for each gold sentence, instead connecting lower nodes in the tree directly to the TURN node. All of the nodes that are connected directly to the top TURN node are counted as predicted sentences. This leads to oversegmentation, as shown in Figure 5.5a.

The pipeline model tends instead to undersegment. This tendency is shown in the end-to-end and the pipeline models' similar segmentation recall, compared to the end-to-end model's very low precision, shown in Table 5.3. These metrics reflect the fact that the end-to-end model predicts too many boundaries, where the pipeline model predicts too few. The examples in Figure 5.5 show the effect this has on segmentation: The end-to-end example shown here has a segmentation F1 score of 57.1%, where the



(a) Gold parse. The nodes shown in green are omitted by both the end-to-end and pipeline models.



(b) Parse predicted by the pipeline model. The nodes shown in red are incorrect.

Figure 5.5: Comparison of gold parse to the pipeline's predicted tree. The end-to-end model's tree isn't shown because the only differences between it and the gold parse is the omission of the three nodes that are highlighted in green in the gold parse. This example has been edited for length and clarity, and part-of-speech tags have been omitted.

pipeline has a score of 66.7%.

However, when scoring parses, the end-to-end model is penalized much less. Both models omit three nodes from the tree in Figure 5.5. However, the pipeline model also predicts several nodes high up in the tree that the end-to-end model does not. This leads to a much lower parse F1 score for the pipeline model: 69.2%, compared to the end-to-end model's 88.0%.

In fact, the pipeline's better segmentation seems to actively *worsen* its parse score. The pipeline's undersegmentation comes from predicting nodes that dominate entire predicted sentences — like the S node in Figure 5.5b. Because the S node erroneously spans the entire turn, it is more likely that its VP daughter will also span too many nodes, as it does in this example. This example shows that while errors of *omission* — like the missing S nodes in Figure 5.5a — are penalized once in the parse score, errors of *commission* — like inserted S node in Figure 5.5b — tend to cascade to other nodes and so are penalized more.

The phenomena in 5.5 affect performance on the whole development set. This can be demonstrated by looking at the interaction of parse precision and segmentation performance. If only considering examples where the models predict an incorrect segmentation (as in the example shown in Figure 5.5), the pipeline model predicts many more incorrect nodes overall. This lowers its parse precision by 5.22 percentage points on the development set (from 85.72% to 80.50%). By comparison, the end-to-end model's parse precision is less affected by segmentation quality: On the subset of mis-segmented examples, its parse precision only drops by 3.96 percentage points (from 86.20% to 82.24%).

This pattern suggests that in order to generate a quality segmentation, it is, unsurprisingly, better to include the segmentation task as an explicit objective, rather than treating it as a secondary property of a parse tree. A potential avenue for future work would be to keep the end-to-end approach in place, since it does produce better parses, but to add an explicit sentence boundary labeling objective to the model. Having both tasks represented as part of the objective function used in training the model may help to optimize performance on both tasks.

I propose that the end-to-end model's superior parsing ability comes from the fact that it is able to model all syntactic units, including sentences, in the same way. The pipeline model has to treat sentences as being logically prior to and distinct from all sub-sentence units, which leads to the error propagation described above. By modeling sentences and other syntactic constituents similarly, the end-to-end model is able

to propose the sub-sentence nodes that lead to the best parses overall, without being bound to a certain sentence segmentation.

5.7 Conclusion

Previous work has shown that prosody improves parse quality. The present work shows that prosody improves parse and sentence segmentation quality simultaneously. However, a high parse performance need not always coincide with a high sentence segmentation performance: A pipeline model does sentence segmentation better, but an end-to-end model produces better parses. I propose that this is because the end-to-end parser models sentence boundaries the same way as other syntactic boundaries. By treating sentences as just another kind of syntactic unit, the end-to-end model is able to take advantage of prosody to produce better parses overall.

Chapter 6

Segmentation with noise

6.1 Introduction

So far, this thesis has described two tasks for which prosodic information is helpful, namely parsing and sentence segmentation (Chapter 5). In another case, word order effects in speech translation, prosody seemed to have no measurable effect at all. Generally, when a task does show benefits from including prosodic information, they are not always particularly large, which fits with the findings of several previous researchers (e.g., Tran et al. 2018). The hypothesis underlying this chapter is that in these previous experiments, prosody's ability to help has been limited by the fact that the information in the prosodic signal is largely redundant with information in the lexical signal. Adding prosody to the text inputs is generally adding more of the same information, and so the effects are modest.

However, this redundancy between lexical and prosodic signals potentially has a utility in real-world settings that these previous experiments don't demonstrate. The experiments in Chapter 5 included a human-generated transcript of the audio, which is unrealistic whether the goal is investigating human speech perception or improving machine speech perception. In practice, both speech production and speech perception are noisy processes, and making these processes robust to this kind of noise is important in order to avoid information loss. One way to make the speech process more robust is to convey the same information in more than one channel. If some information is conveyed by both the lexical and prosodic channels, it makes it more likely that this information will get through. Furthermore, the prosodic signal may be particularly robust to some kinds of noise in ways that the lexical signal is not: Features such as pitch, energy, and duration can often be recovered from audio that has even undergone

drastic processing, such as low-pass filtration, that fully obscures the lexical content (Knoll et al., 2009). The experiments in this chapter aim to demonstrate the utility of the redundancy between the prosodic and lexical signals.

In this chapter, as in Chapter 5, the effect of prosody is measured on a downstream task, namely sentence segmentation. However, instead of using human-generated transcripts as input, the models use automatic speech recognition (ASR)-generated transcripts, as a better model of the noisy nature of real-world speech perception. Furthermore, varying levels of acoustic noise are added to the audio that is used to generate the ASR transcripts and features for the acoustic correlates of prosody. The primary hypothesis is that the noisier the setting, the more benefit there will be from including prosodic information, specifically because it creates a useful redundancy in the overall input signal. Furthermore, I hypothesize that prosodic information will be less affected by the introduction of noise than lexical information.

The results of these experiments support both hypotheses: As the level of noise increases, the text-only and text+prosody models both decline in performance. However, the text-only model declines much more sharply. This shows that the benefits from including prosodic information are higher when there is more noise present. Finally, the prosody-only model can't perform as well as any model that has lexical information, but its performance is more robust in the face of noise.

6.2 Method

As in Chapter 5, the dataset used here is SWBD-NXT (Calhoun et al., 2010). The sentence segmentation model is also the same as the one used in the previous chapter: a transformer-based sequence labeling model. It can take text inputs, prosodic inputs, or both, which allows for comparison between the effects of each kind of input. While it's expected that text is necessary to perform well on this task, including the prosody-only model shows how informative prosody is relative to text, and how robust it is to noise.

6.2.1 ASR system

The primary difference from previous work is that the text inputs here are generated by an ASR system, with varying levels of acoustic noise in the input signal. The ASR system selected is based on the Whisper system (Radford et al., 2022). Whisper

has an overall high level of performance, but in its original form, it doesn't produce timestamps that align the word tokens to the audio. These are necessary, since the pitch and energy features are calculated over the acoustic frames corresponding to each word token. For that reason, Louradour (2023)'s reimplementation of Whisper is used, since it has been augmented to generate word-token-level timestamps.

The ASR system is run over entire conversations. The output is then divided into turns, by finding the predicted word boundary that is nearest to each turn boundary from SWBD-NXT (Calhoun et al., 2010).

The word-token-based timestamps are then used to extract the corresponding acoustic features for each word. The pitch and energy features are generated in the same way as those described in Chapter 5, but the ASR-generated timestamps are used to extract each word's pitch and energy features. The pause duration feature is omitted, because the timestamps generated by the ASR system tend to leave no gap between word tokens. This would lead all pause duration features to be identical, and therefore uninformative. Word duration is also calculated similarly to Chapter 5: The duration of each word, as given by the ASR-generated timestamps, is normalized by the average duration of that word type. Rather than recalculating each word's average duration from the ASR transcripts, the average durations calculated from the SWBD-NXT annotations are used to normalize each word's duration. This choice is made for simplicity in processing, but it isn't predicted to have any systematic effects on the informativity of the duration feature. Since the ASR-generated word token durations are less accurate — in unpredictable ways — than the word token durations from the SWBD-NXT annotations, it's not clear that basing the average duration value on these less accurate durations would lead to a more informative normalized duration value.

6.2.2 Noising the audio

In addition to using the original audio to generate ASR transcripts, acoustic noise is also added to the audio to see the effect it has on sentence segmentation performance. The audio is corrputed with additive white Gaussian noise at a variety of signal-to-noise ratios (SNRs). The SNR is the ratio between the power of the original acoustic signal and the power of the added noise:

$$SNR = \frac{P_{signal}}{P_{noise}}$$

The power P_{signal} is calculated as the root mean square of the waveform over each

SNR (dB)	**WER (%)**
Clean	22.89
40	26.30
30	43.86
20	79.49

Table 6.1: Word error rate on the development set for different levels of noise. Note that a higher signal-to-noise ratio (SNR) means a lower level of noise.

input segment (in this case, conversational sides). The SNR levels tested are 40, 30, and 20 dB, as well as the condition with the clean, original audio. The noised audio is input to the ASR system to generate a transcript, and it is also used to generate the pitch and energy features.

As expected, higher levels of noise lead the ASR system to produce output with higher word error rates (WER). Table 6.1 shows the WER for each level of noise.

Using additive white Gaussian noise for this step is a simplification that is used because it makes it easier to measure the strength of the added noise in terms of signal-to-noise ratio. However, in real-life scenarios, noise is more likely to be more varied than white noise. In particular, white noise is unlikely to obscure the fundamental frequency of the speech signal as much as, say, the noise of other background voices would. To assess whether these results hold for other, more realistic types of noise, future work should include experiments with other kinds of noise.

6.2.3 Model training

The sentence segmentation model used here is the same as the one used in the pipeline model in Chapter 5, which consists of a transformer-based encoder. The model can take in text and prosodic inputs, and outputs a label for each word.

For each noise level noted in Section 6.2.2, the audio for the entire corpus is noised, and an ASR transcript of the data is produced. Then, a new segmentation model is trained using the ASR transcripts as the text inputs, and the acoustic features (pitch, energy, and duration) extracted from the noised acoustic data.

6.2.4 Transferring sentence boundaries to the ASR text

One important part of setting up this task is transferring the human-annotated sentence boundaries from the gold transcript to the corresponding locations within the ASR transcript. These translated sentence boundaries are used both for training the sentence

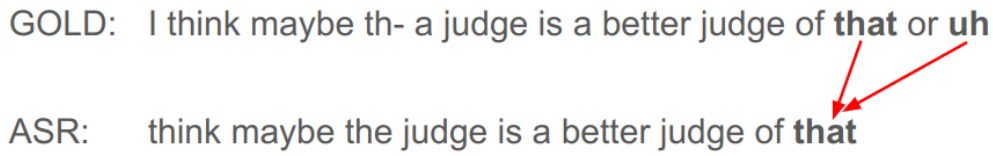


Figure 6.1: Mapping the sentence boundaries to the corresponding locations in the ASR transcript. Some gold boundaries are mapped to the same location in the ASR transcript.

		SNR (dB)			
	Gold	**Clean**	**40**	**30**	**20**
Text + prosody	76.84	54.05	52.03	44.56	29.40
Text only	72.08	48.19	45.07	38.34	19.18
Prosody only	64.01	21.51	22.95	17.62	10.53
Δ btwn text & text+prosody	4.76	5.86	6.96	6.22	10.22

Table 6.2: Segmentation F1 score for each model at various levels of noise, and the difference in performance between the text and text+prosody models (Δ). Shows the same results as Figure 6.2, with the additional inclusion of the F1 scores for the gold data.

segmenter, and as a reference for evaluating it.

As with turn boundaries, each gold sentence boundary is assigned to the nearest word token boundary in the ASR transcript by timestamp. This mapping is not always exactly one-to-one: Two gold sentence boundaries may be closest to the same ASR word token boundary. Figure 6.1 shows an example of this from the development set. In this case, the two gold sentence boundaries are collapsed into a single boundary in the ASR transcript. As a result, there end up being fewer sentence boundaries in the ASR transcripts than in the gold transcripts. For example, there are approximately 39k gold boundaries in the train set, but only 36k in the corresponding ASR examples. These translated ASR sentence boundaries are used both to train the sentence segmentation model and to evaluate it.

6.3 Results and analysis

As shown in Figure 6.2, while noise generally negatively impacts all performance, the benefits of adding prosodic input to the text input rise with the level of noise. The

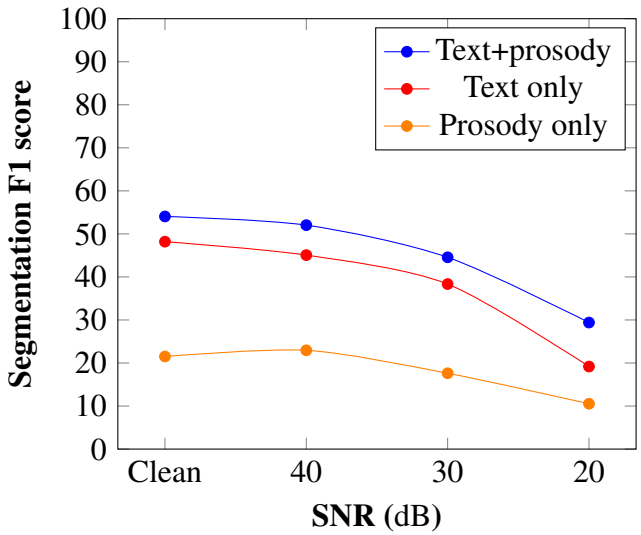


Figure 6.2: Segmentation performance at various levels of noise. Note that a high signal-to-noise ratio (SNR) indicates a low level of noise. These results are also shown in Table 6.2.

same results are also shown in Table 6.2, which also includes the results produced from using gold transcripts and timestamps instead of ASR-generated ones.[1] With no noise, the gap in performance is only 5.86 points, but as noise is added, starting at a signal-to-noise ratio of 40 dB and increasing to a SNR of 20 dB, this gap grows (not entirely monotonically), eventually reaching 10.22 points of difference.

Note that the prosody-only results are, as predicted, significantly weaker than either of the models that include text inputs. However, the prosody-only model is generally more robust to noise than the text-only model: While the text-only model's performance drops by approximately 60% between the no noise and 20 dB SNR models, the prosody-only model's performance, while lower overall in all conditions, only drops by approximately 50%. The text+prosody model falls between these two in robustness: Its performance drops by 55% between the no noise and 20 dB SNR conditions.

These results support the central hypothesis of this chapter in several ways. First, these results are consistent with the idea that there is a high level of redundancy in the information between the prosodic and the lexical channels, especially in the clean conditions. The performance of the text+prosody model is not equal to the performance of the text model plus the performance of the prosody model, showing that much of the

[1]Note that the gold data scores are not wholly comparable to the other scores shown in Table 6.2. These are scored on their ability to predict human-annotated sentence boundaries, whereas all the ASR-based models are scored using sentence boundaries that have been 'translated' to the corresponding spot in the ASR transcript.

information they contribute to the task is redundant. Second, as noted above, these results are consistent with the idea that prosody is more robust to noise than text is. This contributes to the final point: As noise is added, there is more benefit to having two channels available through which the some of the same information can travel. This suggests that as both the text and prosodic channels are degraded by noise, the higher robustness of the prosodic channel helps the text+prosody model perform better.

6.4 Conclusion

The experiments in this chapter are aimed at answering the question of how prosody helps a sentence segmentation model in the presence of noise. They show that prosody is overall more robust to noise than the lexical channel. Of all models, the text+prosody model performs the best. However, without noise, it only modestly out-performs the text-only model. As noise increases, the performance of all models declines, but the model without prosody declines the most sharply, indicating that prosody is more robust to noise. This indicates that in noisy settings, prosody is a much more important information source than in idealized settings, including the settings that many previous experiments with prosody have used.

Chapter 7

Conclusion

The experiments in this thesis have shown that prosody can be a helpful information source in several NLP models, depending on the task and experimental set-up. Chapter 3 shows some general principles for incorporating prosodic inputs, particularly the importance of a wider input context. Chapter 4 discusses experiments with speech translation (ST), where I hypothesized that the ST model would be able to learn that prosodic events in the English source speech corresponded to word order variations in the Russian target. In fact, this effect wasn't observed, likely because the ST model is biased towards reproducing the source word order, and so word order variations aren't an effective way to see effects from prosody. Chapter 5 describes a task where prosody is decidedly helpful: combined sentence segmentation and syntactic parsing. Prosody particularly helps with the segmentation part of this task. Though improved segmentation doesn't always lead to improved parses, prosody consistently raised performance on both of these subtasks. Finally, Chapter 6 revisits the sentence segmentation task and sheds some light on one way in which prosody can be particularly useful: When noise is present, prosody provides a 'back-up' channel through which information can still successfully pass.

These experiments collectively show that the linguistic information in prosody can be useful, given a few conditions. First, the task has to be selected carefully, since some linguistic phenomena (such as sentence boundaries) simply are more robustly signaled by prosody than others (such as syntactic structure). Second, the model has to be selected so that an effect can be measured — a speech translation model with a strong bias towards keeping word order constant isn't a good candidate for measuring word order variations, even though another model for the same task with different biases could work. Finally, prosody is generally more helpful with tasks that are closer to a

real-world language-use scenario — whether because sentences aren't pre-segmented, speech isn't transcribed, or the acoustic signal is obscured by noise. This is because the information in the prosodic channel is largely redundant with information in the lexical channel. The utility of prosody is more clear when there are obstacles to the clear perception of the lexical channel.

This work helps point to one reason why there is a disconnect between the importance of prosody in human language use, and the relatively smaller benefit it has been shown to have in NLP applications. In real-world usage, there are all kinds of obstacles to clear speech communication: There may be external noise present, speakers may produce disfluencies or interrupt one another, the listener(s) may lack important contextual knowledge for comprehension, and so on. Human language has evolved mechanisms to make communication possible despite these kinds of interference, and prosody is a particularly robust channel through which information about phrase boundaries, information status, etc. can be transmitted. However, NLP models usually abstract away from the difficulties of real-world speech communication, generally presenting models with text inputs instead of noisy, unedited speech inputs. Altering a tidy, text-based NLP model to include prosodic inputs doesn't necessarily provide prosody the opportunity to shine. To see the benefits of prosody, the model should better approximate the kinds of interference in the communication environment, which could include acoustic noise (as in Chapter 6), or disfluencies and unsegmented utterances (as in Chapter 5).

7.1 Limitations

The work in this thesis faces several limitations. The most significant limitations are due to the kind of data available. The experiments in Chapter 3 require a corpus of speech data with aligned transcripts annotated by hand with pitch accents. Chapter 5's experiments require a speech corpus with human-generated parses and aligned transcripts. Only a very small number of corpora fit these criteria — Boston University Radio News Corpus (Ostendorf et al., 1995), and Switchboard-NXT (Calhoun et al., 2010) are the ones used here. These corpora are valuable resources, but they have limitations. First, they are relatively small, which limits the complexity of the models that can be reliably trained on them. Second, both are American English corpora, which means that the question of how these findings may generalize to other languages remains unanswered. Third, these corpora only represent two genres (radio broadcasts

and telephone conversations between pairs of strangers), but many more genres exist that merit study, including child-directed speech, conversations with more than two participants, and others. Again, it's not yet clear how or if the results in this thesis generalize to these other genres. Fourth, these corpora are both from the 1990s. Some prosodic phenomena, such as pervasive use of creaky voice in General American English, have been documented primarily after the 1990s (e.g., Yuasa 2010; Ingle et al. 2005). More up-to-date recordings might show different effects from voicing features, which generally were not helpful to the models discussed here. Finally, the audio in SWBD-NXT is relatively low-quality, and in Chapter 3, this seemed to lead to less consistent benefits from the prosodic inputs. Higher quality audio or better feature extraction procedures would help to elucidate whether these differences were an artifact of the audio quality or were a result of something more meaningful, such as the different genres of the corpora.

Beyond the limitations of data, this work is limited by some of the modeling assumptions made. As noted above, one goal in this work was to move away from some modeling assumptions that make prosody-assisted NLP experiments unrealistic models of either deployed applications of these methods, or of human language use. For example, the work in Chapter 5 moved away from using speech that had been presegmented into sentences, while Chapter 6 moved away from having a human-created, hand-aligned transcript altogether. However, this work does retain some unrealistic modeling assumptions. For example, the experiments in Chapters 3, 4, and 5 still use human-created transcripts as inputs. All of the experiments assume that, even if ASR is involved, there are separate speech and text channels, and that there is a reasonably reliable alignment between the two — which is not a realistic model of human speech perception, and may not be a practical approach in a deployed application setting either. The following section outlines some ways in which future research might move away from these limiting assumptions.

Another limitation comes from the types of prosodic input features given to models. The neural models used here allow us to move away from some of the simplifying assumptions that constrained statistical models (e.g., using features such as manually calculated pitch trajectory at the word level), and instead learn feature representations for the input as part of the overall model training. While this improves performance generally, it is still a simplification: the features for acoustic correlates of prosody are still extracted from the original audio signal, and the richer spectral features are discarded. As mentioned in Chapters 3 and 5, this can be particularly problematic for

features such as pitch, where the pitch extraction algorithms can be especially unreliable for telephone recordings (Wang and Seneff, 2000). Additionally, the approaches used in Chapters 5 and 6 to create word and pause duration features are limited by the accuracy of the alignment of text to the speech signal, which may be particularly inaccurate in the case of machine-generated alignments. These limitations in input features may have reduced the amount of benefit from providing prosodic inputs to the models in this thesis.

Finally, there are limitations to this work caused by time constraints. Since the work in this thesis was completed over a period of several years, some of the work uses models that no longer represent the state-of-the-art (e.g., the LSTM-based model of Chapter 3 and the speech translation model of Chapter 4). Revisiting these same experiments with newer model architectures would be a natural follow-up to the research here. Additionally, not all research paths could be taken because of limited available time. These paths are detailed in the following section.

7.2 Directions for future work

Given the findings of this thesis, one important direction for future work is to move away from simplifying modeling assumptions and towards incorporating prosody in models of more realistic language usage. This could be for engineering goals — e.g., building dialog agents that are more robust to the noise that arises in real-world usage — or for scientific goals, such as better understanding prosody's role in sentence processing or language acquisition.

One simplifying assumption made in almost all of the experiments presented here is that the model has access to separate text and acoustic signals, with an explicit alignment between the two (which allows us to extract acoustic features corresponding to each word). This isn't an entirely unrealistic assumption — of course a language user who is able to successfully understand a speech stream is presumably able to segment that stream into words, and successfully process both the lexical and prosodic information for each word. However, the process used here — transcribing the audio into text, performing an alignment between that text and the source audio, and then extracting the acoustic correlates of prosody for each word — is not a realistic model of how language understanding works in practice. The requirement for an explicit alignment between audio and text is a particularly bothersome constraint, since human-generated alignments are costly and don't exist for most corpora, and machine-generated ones are

not always reliable. It would be more realistic — and convenient — to have a model use the original speech signal as input and extract both the lexical and prosodic information simultaneously.

Recent innovations in speech processing could facilitate this kind of modeling. While older speech models often use MFCCs[1] as inputs, which don't preserve all prosodic information well, more recent models such as wav2vec (Baevski et al., 2020) use the original audio signal as input, which allows for better access to all acoustic correlates of prosody. Architectures like these could potentially be adapted for use as end-to-end models for NLP tasks such as parsing, which would allow a model to learn lexical and prosodic information from a single source. One recent example of using this kind of model for purposes other than ASR is wav2tobi (Zhai and Hasegawa-Johnson, 2023), a system for predicting ToBI prosodic annotations from audio input.

Modeling approaches that use the original audio signal as input may also help address the limitations of the feature extraction process used for acoustic correlates of prosody. Representations of prosodic information can be learned directly from the unprocessed audio signal, rather than relying on feature extraction algorithms that may either distort or omit some important acoustic information. In many of these approaches, a model first learns to create speech representations in a self-supervised pre-training step, before the training of a task-specific model. Possible pre-training objectives include predicting future audio frames (e.g., Yue and Li (2021)) or distinguishing future audio frames from distractors (e.g., wav2vec; Baevski et al. (2020)). These learned representations have the potential to outperform manually extracted acoustic features, including on the tasks in this thesis.

While this approach of learning lexical and prosodic information from the speech has engineering advantages, it is less helpful for testing hypotheses about the role of prosody in human language use. When the lexical and prosodic features presented to a model are fully separate, it is possible to answer questions about the effect of prosody, as distinct from the lexical channel. Investigating these questions is much harder if the signals are conflated. There are ways of obscuring some of the prosodic information in the audio signal, such as using encodings like MFCCs that don't preserve pitch, or even processing the audio to level out energy or phoneme and pause duration. However, these methods are not perfect at obscuring prosody, and run the risk of introducing artefacts to the input signal, so this approach is not clearly better suited to answering questions about the role of prosody in human language use.

[1] Mel-frequency cepstral coefficients

In addition to moving toward experimental settings that better mimic real language use, future research in this vein should consider other NLP tasks that might be enhanced by the addition of prosody. One such task is coreference resolution, which requires a model to track relationships between entities over entire discourses. Since prosody plays a role in signaling the information status of entities in English, it should be helpful in coreference resolution. Additionally, any number of tasks relating to spoken dialog are good candidates for benefiting from prosody. These tasks include semantic parsing and dialog act recognition. Insofar as these tasks are often bundled together into end-to-end models, these joint models may also benefit from the inclusion of prosody.

Another possible direction for future work would be to focus on the question of how prosody functions in human language use. For example, it is well established that child-directed speech has distinctive prosodic features cross-linguistically, such as an expanded pitch range (Baron, 1990). A child has to accomplish a number of different tasks in order to acquire a language, including segmenting the speech stream into words and phrases, learning syntactic structures, and learning lexical and phrasal semantics. Modeling these processes both with and without prosodic information could help elucidate the role of prosody in language acquisition. Some work along these lines exists already, such as Pate and Goldwater (2013)'s work showing that the process of grammar induction is helped even by just one type prosodic information (namely word duration).

Other questions about adult language use and understanding could also benefit from a modeling approach that incorporates prosody. For example, one question that I began working on was the question of whether prosody was more important in incremental parsing than traditional parsing. Incremental parsing doesn't give the model access to the forward context, which is the kind of realistic ablation of input information that prosody could potentially help with. Initial experiments with this question didn't show a significant benefit to prosody, but it is a question that would bear further investigation.

As NLP models move closer to approximating human performance on a number of tasks, closing the remaining gap between machine and human often becomes more difficult. NLP models may benefit from imitating humans speakers in their ability to effectively use prosody in processing speech. The work in this thesis suggests that as NLP models move closer to operating in the noisier, real-world settings of human language use, prosody will only become more important as a source of information.

Bibliography

Tanvirul Alam, Akib Khan, and Firoj Alam. Punctuation restoration using transformer models for high-and low-resource languages. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 132–142, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.wnut-1.18. URL https://www.aclweb.org/anthology/2020.wnut-1.18.

D.J. Allerton. The notion of 'givenness' and its relations to presupposition and to theme. *Lingua*, 44(1):133–168, 1978. ISSN 0024-3841. doi: https://doi.org/10.1016/S0024-3841(20)30063-2. URL https://www.sciencedirect.com/science/article/pii/S0024384120300632.

Gayle M Ayers. Discourse functions of pitch range in spontaneous and read speech. *OSU Working Papers in Linguistics*, 1994.

Matthew Aylett and Alice Turk. The smooth signal redundancy hypothesis: A functional explanation for relationships between redundancy, prosodic prominence, and duration in spontaneous speech. *Language and Speech*, 47(1):31–56, 2004. doi: 10.1177/00238309040470010201. URL https://doi.org/10.1177/00238309040470010201. PMID: 15298329.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12449–12460. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/92d1e1eb1cd6f9fba3227870bb6d7f07-Paper.pdf.

Naomi Baron. *Pigeon-Birds and Rhyming Words: The Role of Parents in Language Learning*. Prentice Hall Regents, Englewood Cliffs, New Jersey, 1990.

Alexandra Birch, Miles Osborne, and Phil Blunsom. Metrics for MT evaluation: Evaluating reordering. *Machine Translation*, 24(1):15–26, March 2010. doi: 10.1007/s10590-009-9066-5.

Dwight Bolinger. Accent is predictable (if you're a mind-reader). *Language*, pages 633–644, 1972.

Jason M Brenier, Daniel M Cer, and Daniel Jurafsky. The detection of emphatic words using acoustic and lexical features. In *Ninth European Conference on Speech Communication and Technology*, 2005.

Sasha Calhoun. The centrality of metrical structure in signaling information structure: A probabilistic perspective. *Language*, pages 1–42, 2010.

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*, 44:387–419, 12 2010. doi: 10.1007/s10579-010-9120-1.

Eugene Charniak and Mark Johnson. Edit detection and parsing for transcribed speech. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*, 2001. URL `https://www.aclweb.org/anthology/N01-1016`.

Xiaoyin Che, Cheng Wang, Haojin Yang, and Christoph Meinel. Punctuation prediction for unsegmented transcript based on word vector. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may 2016. European Language Resources Association (ELRA). ISBN 978-2-9517408-9-1.

Seamless Communication, Loïc Barrault, Yu-An Chung, Mariano Cora Meglioli, David Dale, Ning Dong, Paul-Ambroise Duquenne, Hady Elsahar, Hongyu Gong, Kevin Heffernan, John Hoffman, Christopher Klaiber, Pengwei Li, Daniel Licht, Jean Maillard, Alice Rakotoarison, Kaushik Ram Sadagopan, Guillaume Wenzek, Ethan Ye, Bapi Akula, Peng-Jen Chen, Naji El Hachem, Brian Ellis, Gabriel Mejia Gonzalez, Justin Haaheim, Prangthip Hansanti, Russ Howes, Bernie Huang, Min-Jae Hwang, Hirofumi Inaguma, Somya Jain, Elahe Kalbassi, Amanda Kallet, Ilia Kulikov, Janice Lam, Daniel Li, Xutai Ma, Ruslan Mavlyutov, Benjamin Peloquin,

Mohamed Ramadan, Abinesh Ramakrishnan, Anna Sun, Kevin Tran, Tuan Tran, Igor Tufanov, Vish Vogeti, Carleigh Wood, Yilin Yang, Bokai Yu, Pierre Andrews, Can Balioglu, Marta R. Costa-jussà, Onur Celebi, Maha Elbayad, Cynthia Gao, Francisco Guzmán, Justine Kao, Ann Lee, Alexandre Mourachko, Juan Pino, Sravya Popuri, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, Paden Tomasello, Changhan Wang, Jeff Wang, and Skyler Wang. Seamlessm4t: Massively multilingual and multimodal machine translation, 2023.

Nelson Cowan. Short-term memory, working memory, and their importance in language processing. *Topics in language disorders*, 17(1):1–18, 1996.

Anne Cutler, Delphine Dahan, and Wilma van Donselaar. Prosody in the comprehension of spoken language: A literature review. *Language and Speech*, 40(2):141–201, 1997.

Mattia A. Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. MuST-C: a Multilingual Speech Translation Corpus. 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 2012–2017, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

Kaja Dobrovoljc and Joakim Nivre. The Universal Dependencies treebank of spoken Slovenian. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1566–1573, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA). URL https://aclanthology.org/L16-1248.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

Bradley Efron and Robert J Tibshirani. *An introduction to the bootstrap*. CRC press, 1994.

Florian Eyben, Felix Weninger, Florian Gross, and Björn Schuller. Recent developments in OpenSMILE, the Munich open-source multimedia feature extractor. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 835–838, 2013.

R. Fernandez, A. Rosenberg, A. Sorin, B. Ramabhadran, and R. Hoory. Voice-transformation-based data augmentation for prosodic classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5530–5534, 2017.

David Gaddy, Mitchell Stern, and Dan Klein. What's going on in neural constituency parsers? an analysis. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 999–1010, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1091. URL https://www.aclweb.org/anthology/N18-1091.

Mattia A. Di Gangi, Matteo Negri, and Marco Turchi. Adapting Transformer to End-to-End Spoken Language Translation. In *Proc. Interspeech 2019*, pages 1133–1137, 2019.

P. Ghahremani, B. BabaAli, D. Povey, K. Riedhammer, J. Trmal, and S. Khudanpur. A pitch extraction algorithm tuned for automatic speech recognition. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2494–2498, 2014. doi: 10.1109/ICASSP.2014.6854049.

Hussein Ghaly and Michael Mandel. Analyzing human and machine performance in resolving ambiguous spoken sentences. In Nicholas Ruiz and Srinivas Bangalore, editors, *Proceedings of the Workshop on Speech-Centric Natural Language Processing*, pages 18–26, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4603. URL https://aclanthology.org/W17-4603.

John J Godfrey, Edward C Holliman, and Jane McDaniel. Switchboard: Telephone speech corpus for research and development. In *Acoustics, speech, and signal processing, ieee international conference on*, volume 1, pages 517–520. IEEE Computer Society, 1992.

Yoshihiko Gotoh and Steve Renals. Sentence boundary detection in broadcast speech transcripts. In *ASR2000-Automatic Speech Recognition: Challenges for the new Millenium ISCA Tutorial and Research Workshop (ITRW)*, 2000.

Avashna Govender and Simon King. Cognitive load of modern tts systems under noisy conditions. In *Cognitive AI 2023*, 2023. URL `http://hdl.handle.net/10204/13517`.

Michelle Gregory, Mark Johnson, and Eugene Charniak. Sentence-internal prosody does not help parsing the way punctuation does. *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, 04 2004.

Michelle L Gregory and Yasemin Altun. Using conditional random fields to predict pitch accents in conversational speech. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 677. Association for Computational Linguistics, 2004.

Vadim Gudkov, Olga Mitrofanova, and Elizaveta Filippskikh. Automatically ranked russian paraphrase corpus for text generation. *Proceedings of the Fourth Workshop on Neural Generation and Translation*, 2020.

Ulrike Gut and Petra Bayerl. Measuring the reliability of manual annotations of speech corpora. *Interspeech 2004*, 01 2004.

John Hale, Izhak Shafran, Lisa Yung, Bonnie J. Dorr, Mary Harper, Anna Krasnyanskaya, Matthew Lease, Yang Liu, Brian Roark, Matthew Snover, and Robin Stewart. PCFGs with syntactic and prosodic indicators of speech repairs. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 161–168. Association for Computational Linguistics, 2006. doi: 10.3115/1220175.1220196. URL `https://www.aclweb.org/anthology/P06-1021`.

Mattias Heldner, Eva Strangert, and Thierry Deschamps. A focus detector using overall intensity and high frequency emphasis. In *Proc. of ICPhS*, volume 99, pages 1491–1494, 1999.

Zhongqiang Huang and Mary Harper. Appropriately handled prosodic breaks help PCFG parsing. In Ron Kaplan, Jill Burstein, Mary Harper, and Gerald Penn, editors,

Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages 37–45, Los Angeles, California, June 2010. Association for Computational Linguistics. URL https://aclanthology.org/N10-1005.

Jennifer Ingle, Richard Wright, and Alicia Wassink. Pacific northwest vowels: A seattle neighborhood dialect study. *Journal of The Acoustical Society of America - J ACOUST SOC AMER*, 117:2459–2459, 04 2005. doi: 10.1121/1.4787252.

Mahaveer Jain, Gil Keren, Jay Mahadeokar, Geoffrey Zweig, Florian Metze, and Yatharth Saraf. Contextual RNN-T for Open Domain ASR. In *Proc. Interspeech 2020*, pages 11–15, 2020. doi: 10.21437/Interspeech.2020-2986. URL http://dx.doi.org/10.21437/Interspeech.2020-2986.

Paria Jamshid Lou, Yufei Wang, and Mark Johnson. Neural constituency parsing of speech transcripts. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *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 2756–2765, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1282. URL https://aclanthology.org/N19-1282.

Sun-Ah Jun. Prosodic Typology. In Sun-Ah Jun, editor, *Prosodic Typology: The Phonology of Intonation and Phrasing*, chapter 16. Oxford University Press, 01 2005. ISBN 9780199249633. doi: 10.1093/acprof:oso/9780199249633.001.0001. URL https://doi.org/10.1093/acprof:oso/9780199249633.001.0001.

Jeremy G. Kahn and Mari Ostendorf. Joint reranking of parsing and word recognition with automatic segmentation. *Computer Speech and Language*, 26(1):1 – 19, 2012. ISSN 0885-2308. doi: https://doi.org/10.1016/j.csl.2011.03.002. URL http://www.sciencedirect.com/science/article/pii/S0885230811000209.

Jeremy G. Kahn, Mari Ostendorf, and Ciprian Chelba. Parsing conversational speech using enhanced segmentation. In *Proceedings of HLT-NAACL 2004*, pages 125–128, Boston, Massachusetts, USA, May 2 - May 7 2004. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N04-4032.

Jeremy G. Kahn, Matthew Lease, Eugene Charniak, Mark Johnson, and Mari Ostendorf. Effective use of prosody in parsing conversational speech. In *Proceedings*

of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, HLT '05, page 233–240, USA, 2005. Association for Computational Linguistics.

Nikita Kitaev and Dan Klein. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne, Australia, 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P18-1249.

Monja A. Knoll, Maria Uther, and Alan Costall. Effects of low-pass filtering on the judgment of vocal affect in speech directed to infants, adults and foreigners. *Speech Communication*, 51(3), 2009.

Jáchym Kolář, Elizabeth Shriberg, and Yang Liu. Using prosody for automatic sentence segmentation of multi-party meetings. In *International Conference on Text, Speech and Dialogue*, pages 629–636. Springer, 2006.

Ivana Kruijff-Korbayová and Mark Steedman. Discourse and information structure. *Journal of Logic, Language, and Information*, 12(3):249–259, 2003.

Ilse Lehiste. Phonetic disambiguation of syntactic ambiguity. *The Journal of the Acoustical Society of America*, 53(1 Supplement):380–380, 01 1973. ISSN 0001-4966. doi: 10.1121/1.1982702. URL https://doi.org/10.1121/1.1982702.

Gina-Anne Levow. Context in multi-lingual tone and pitch accent recognition. In *Ninth European Conference on Speech Communication and Technology*, 2005.

Gina-Anne Levow. Automatic prosodic labeling with conditional random fields and rich acoustic features. In *IJCNLP*, 2008.

Roger Levy and T. Florian Jaeger. Speakers optimize information density through syntactic reduction. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*, NIPS'06, page 849–856, Cambridge, MA, USA, 2006. MIT Press.

Jérôme Louradour. whisper-timestamped. https://github.com/linto-ai/whisper-timestamped, 2023.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2): 313–330, 1993. URL https://aclanthology.org/J93-2004.

Anna Margolis, Mari Ostendorf, and Karen Livescu. Cross-genre training for automatic prosody classification. In *Proc. Speech Prosody 2010*, page paper 113, 2010.

M.W. Meteer and A.A. Taylor. *Dysfluency Annotation Stylebook for the Switchboard Corpus*. University of Pennsylvania, 1995. URL https://books.google.com/books?id=xkYEkAEACAAJ.

Ani Nenkova and Dan Jurafsky. Automatic detection of contrastive elements in spontaneous speech. In *2007 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 201–206. IEEE, 2007.

Malvina Nissim, Shipra Dingare, Jean Carletta, and Mark Steedman. An annotation scheme for information status in dialogue. In Maria Teresa Lino, Maria Francisca Xavier, Fátima Ferreira, Rute Costa, and Raquel Silva, editors, *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, May 2004. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2004/pdf/638.pdf.

Elmar Noeth, Anton Batliner, Andreas Kießling, Ralf Kompe, and Heinrich Niemann. Verbmobil: The use of prosody in the linguistic components of a speech understanding system. *IEEE Transactions on Speech and Audio processing*, 8(5):519–532, 2000.

Mari Ostendorf, Patti J Price, and Stefanie Shattuck-Hufnagel. The Boston University radio news corpus. *Linguistic Data Consortium*, pages 1–19, 1995.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/

9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

John K Pate and Sharon Goldwater. Unsupervised Dependency Parsing with Acoustic Cues. *Transactions of the Association for Computational Linguistics*, 1:63–74, 03 2013. ISSN 2307-387X. doi: 10.1162/tacl_a_00210. URL https://doi.org/10.1162/tacl_a_00210.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL http://www.aclweb.org/anthology/D14-1162.

Janet Breckenridge Pierrehumbert. *The phonology and phonetics of English intonation*. PhD thesis, Massachusetts Institute of Technology, 1980.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. The Kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, December 2011.

Patti Price, Mari Ostendorf, Stefanie Shattuck-Hufnagel, and Cynthia Fong. The use of prosody in syntactic disambiguation. In *Speech and Natural Language: Proceedings of a Workshop Held at Pacific Grove, California, February 19-22, 1991*, 1991. URL https://aclanthology.org/H91-1073.

Ellen Prince. Toward a taxonomy of given-new information. *Radical pragmatics*, pages 223–255, 1981.

Ellen F Prince. The zpg letter: Subjects, definiteness, and information-status. *Discourse description: diverse analyses of a fund raising text*, pages 295–325, 1992.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision, 2022.

Vivek Kumar Rangarajan Sridhar, Srinivas Bangalore, and Shrikanth Narayanan. Enriching machine-mediated speech-to-speech translation using contextual information. *Computer Speech and Language*, 27(2):492–508, 2013. ISSN 0885-2308. doi:

https://doi.org/10.1016/j.csl.2011.08.001. URL https://www.sciencedirect.com/science/article/pii/S0885230811000428. Special Issue on Speech-speech translation.

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 *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, May 2006.

Elena Rodionova. *Word Order and Information Structure in Russian Syntax*. PhD thesis, University of North Dakota, 2001.

Andrew Rosenberg. AutoBI - a tool for automatic ToBI annotation. In *Proc. Interspeech 2010*, pages 146–149, 2010. doi: 10.21437/Interspeech.2010-71.

Andrew Rosenberg and Julia Bell Hirschberg. Detecting pitch accents at the word, syllable and vowel level. In *Proceedings of NAACL/HLT*, 2009.

Björn Schuller, Stefan Steidl, Anton Batliner, Alessandro Vinciarelli, Klaus Scherer, Fabien Ringeval, Mohamed Chetouani, Felix Weninger, Florian Eyben, Erik Marchi, et al. The interspeech 2013 computational paralinguistics challenge: Social signals, conflict, emotion, autism. In *Proceedings INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association, Lyon, France*, 2013.

Elisabeth Selkirk. *Phonology and Syntax*. MIT Press, Cambridge, MA, 1984.

Elisabeth Selkirk. Sentence prosody: Intonation, stress, and phrasing. *The handbook of phonological theory*, 1:550–569, 1995.

Elizabeth Shriberg. To 'errrr' is human: Ecology and acoustics of speech disfluencies. *Journal of the International Phonetic Association*, 31:153 – 169, 06 2001. doi: 10.1017/S0025100301001128.

Gabriel Stanovsky, Noah A. Smith, and Luke Zettlemoyer. Evaluating gender bias in machine translation. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational*

Linguistics, pages 1679–1684, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1164. URL https://aclanthology.org/P19-1164.

Mark Steedman. Information structure and the syntax-phonology interface. *Linguistic inquiry*, 31(4):649–689, 2000.

Sabrina Stehwien and Ngoc Thang Vu. Prosodic event recognition using convolutional neural networks with context information. In *Proc. Interspeech 2017*, pages 2326–2330, 2017. doi: 10.21437/Interspeech.2017-1159. URL http://dx.doi.org/10.21437/Interspeech.2017-1159.

Sabrina Stehwien, Ngoc Thang Vu, and Antje Schweitzer. Effects of word embeddings on neural network-based pitch accent detection. In *Proc. 9th International Conference on Speech Prosody 2018*, 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 *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 818–827, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1076. URL https://www.aclweb.org/anthology/P17-1076.

Ottokar Tilk and Tanel Alumäe. Bidirectional recurrent neural network with attention mechanism for punctuation restoration. In *Interspeech 2016*, pages 3047–3051, 2016. doi: 10.21437/Interspeech.2016-1517. URL http://dx.doi.org/10.21437/Interspeech.2016-1517.

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 *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 69–81, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. 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 2019*, pages 4190–4194, 2019. URL http://dx.doi.org/10.21437/Interspeech.2019-3122.

Marianne van Zyl and Johan J. Hanekom. Speech perception in noise: A comparison between sentence and prosody recognition. *Journal of Hearing Science*, 1(2):54–56, 2011. ISSN 2083-389X.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

Michael Wagner and Duane G. Watson. Experimental and theoretical advances in prosody: A review. *Language and Cognitive Processes*, 25(7-9):905–945, 2010. doi: 10.1080/01690961003589492.

Chao Wang and S. Seneff. Robust pitch tracking for prosodic modeling in telephone speech. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100)*, volume 3, pages 1343–1346 vol.3, 2000. doi: 10.1109/ICASSP.2000.861827.

C. W. Wightman, N. M. Veilleux, and M. Ostendorf. Use of prosody in syntactic disambiguation: An analysis-by-synthesis approach. In *Speech and Natural Language: Proceedings of a Workshop Held at Pacific Grove, California, February 19-22, 1991*, 1991. URL https://aclanthology.org/H91-1075.

Colin W Wightman and Mari Ostendorf. Automatic labeling of prosodic patterns. *IEEE Transactions on speech and audio processing*, 2(4):469–481, 1994.

Lesley Wolk, Nassima B. Abdelli-Beruh, and Dianne Slavin. Habitual use of vocal fry in young adult female speakers. *Journal of Voice*, 26:E111–E116, 2011.

Tak-sum Wong, Kim Gerdes, Herman Leung, and John Lee. Quantitative comparative syntax on the Cantonese-Mandarin parallel dependency treebank. In Simonetta Montemagni and Joakim Nivre, editors, *Proceedings of the Fourth International Conference on Dependency Linguistics (Depling 2017)*, pages 266–275, Pisa, Italy, September 2017. Linköping University Electronic Press. URL https://aclanthology.org/W17-6530.

Chenglin Xu, Lei Xie, Guangpu Huang, Xiong Xiao, Eng Siong Chng, and Haizhou Li. A deep neural network approach for sentence boundary detection in broadcast news. In *INTERSPEECH-2014*, pages 2887–2891, 2014.

Jiahong Yuan, Jason M Brenier, and Daniel Jurafsky. Pitch accent prediction: effects of genre and speaker. In *Interspeech*, pages 1409–1412, 2005.

Ikuko Patricia Yuasa. Creaky Voice: A New Feminine Voice Quality for Young Urban-Oriented Upwardly Mobile American Women? *American Speech*, 85(3):315–337, 08 2010. ISSN 0003-1283. doi: 10.1215/00031283-2010-018. URL https://doi.org/10.1215/00031283-2010-018.

Xianghu Yue and Haizhou Li. Phonetically Motivated Self-Supervised Speech Representation Learning. In *Proc. Interspeech 2021*, pages 746–750, 2021. doi: 10.21437/Interspeech.2021-905.

Vicky Zayats and Mari Ostendorf. Giving attention to the unexpected: Using prosody innovations in disfluency detection. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *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 86–95, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1008. URL https://aclanthology.org/N19-1008.

Wanyue Zhai and Mark Hasegawa-Johnson. Wav2ToBI: a new approach to automatic tobi transcription. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2023-August:2748–2752, 2023. ISSN 2308-457X. doi: 10.21437/Interspeech.2023-477. Publisher Copyright: © 2023 International Speech Communication Association. All rights reserved.; 24th International Speech Communication Association, Interspeech 2023 ; Conference date: 20-08-2023 Through 24-08-2023.

Giulio Zhou, Tsz Kin Lam, Alexandra Birch, and Barry Haddow. Prosody in cascade and direct speech-to-text translation: a case study on Korean wh-phrases, 2024.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. The united nations parallel corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3530–3534, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA).