Incorporating Prosody into Neural Speech Processing Pipelines

Applications on automatic speech transcription and spoken language machine translation

Alp Öktem

TESI DOCTORAL UPF / ANY 2018

DIRECTORS DE LA TESI
Dra. Mireia Farrús i Dr. Antonio Bonafonte

DEPARTAMENT DE TECNOLOGIES DE LA INFORMACIÓ I LES COMUNICACIONS

upf. Universitat Pompeu Fabra Barcelona

Write here your dedication

Acknowledgements

Hello world.

This sections before the Introduction should go in Roman numbers (instead of counting twice in Arabic numbers)

Abstract

In the current era where humans build more and more interaction with machines, prosody still is a less-focused phenomena in systems with natural language understanding. Automatic speech recognition systems give focus on “what is said?” and disregard “how it is said?”. In the same way, spoken language translation systems that are built on this kind of “ordinary” recognition, disregard any prosodic information in the input utterance, which might have to be reflected in the translation and synthesis in the target language. Prosody, which is the sum of features such as intonation, stress, and rhythm carried among segments in speech, is an important aspect in human spoken communication. It acts as an essential element from pointing out importance on certain aspects to expressing emotions or to help structure our discourses. In order to arrive to a point where machines can listen to us like humans do, prosody as an essential part of the spoken language needs to be taken into account.

In this dissertation, we study the inclusion of prosody into two applications that involve speech understanding: automatic speech transcription and spoken language translation. We are specifically motivated by two emerging real-world use cases: automatic subtitling and automatic dubbing. These use cases promise an important share in future machine intelligence applications considering the exponential growth of media. In the former case, we propose methods to improve accuracy of automatically generated punctuation through the use of several morphosyntactic and prosodic features. Punctuation, which proves to be essential both for human and machine understanding, is normally missing in ASR output. Taking into knowledge that punctuation is influenced by both syntax and prosody, we first report on the effect of each prosodic feature besides lexical features to punctuation generation on raw speech transcripts. Through our experiments, the optimal feature set proves to be inter-word pausing, word-level intonation and POS features. Our method that uses an attention mechanism over parallel sequences of prosodic and morphosyntactic features indicate an F1 score of 70.3% in terms of overall punctuation generation accuracy on manually generated transcripts. Our proposed system is further demonstrated on a state-of-the-art ASR system and shows promising results.

In the latter problem we deal with enhancing spoken language translation with prosody. We present a novel method for collecting prosodically annotated bilingual datasets and use it to gather an English-Spanish parallel corpus made up of 7000 aligned and transcribed speech segments from original and dubbed version of a TV series. A neural machine translation system is built and trained with a larger movie-domain text corpus and then adapted to our corpus with added pausing features. Results show that prosodic punctuation generation as a preliminary step to translation increases translation accuracy by 1% in terms of BLEU scores. Encoding pauses as an extra feature during translation gives an additional 1% increase to this number. We furthermore extend our translation system to jointly predict pause features in order to be used as an input to a text-to-speech system. We believe our results pave the way for expressive speech translation by inclusion of prosody in both encoding and decoding side of a speech translation system.

The work carried out calls for recently popular neural network based systems that prove to be successful in real-world settings to consider a dimension necessary in human communication, such as prosody. This is done through both experimental and practical methodologies with results indicating that inclusion of prosody can indeed improve these systems.

Resum

En l’era actual en què augmenta considerablement la interacció dels humans amb les màquines, la prosòdia és un fenomen que rep poca atenció en sistemes que inclouen la comprensió del llenguatge natural. Els sistemes de reconeixement de veu se centren en el “què s’ha dit?” i no en el “com s’ha dit?”. De la mateixa manera, els sistemes de traducció de la llengua oral que es construeixen sobre aquest tipus de reconeixement “ordinari”, no tenen en compte la informació prosòdia, la qual s’hauria de reflectir en la traducció i la síntesi de la llengua de destí. La prosòdia, que és la suma de característiques tals com l’entonació, l’accentuació i el ritme al llarg dels segments de veu, és un aspecte molt important en la comunicació humana oral. La prosòdia actua com un element essencial, des del marcatge de la importància en certs aspectes, fins a l’expressió d’emocions a l’ajuda en l’estructuració del discurs. Per tal d’arribar a un punt en què les màquines puguin escoltar-nos de la mateixa manera que ho fem els humans, la prosòdia com a element essencial del llenguatge parlat s’ha de tenir en compte.

En aquesta tesi estudiem la inclusió de la prosòdia en dues aplicacions que involucren la comprensió de la parla: la transcripció automàtica de la parla i la traducció de la llengua oral. Ens motiven especialment dos casos d’ús del món real: la subtitulació automàtica i el doblatge automàtic. Aquest casos d’ús prometen tenir un rol important en les futures aplicacions d’intel·ligència artificial, tenint en compte el creixement exponencial dels mitjans de comunicació. En el primer cas, proposem mètodes per millorar la precisió de la puntuació generada automàticament a través de l’ús de diverses característiques morfosintàctiques i prosòdiques. La puntuació, que s’ha demostrat que és essencial tant per a la comprensió humana com automàtica, normalmente no hi és a la sortida del reconeixement automàtic de la parla. Tenint en compte que tant la sintaxi com la prosòdia tenen una influència rellevant en la puntuació, primer mostrem l’efecte de cada característica prosòdica, a més a més de les característiques lèxiques, per a la generació de la puntuació en les transcripcions de veu. Els experiments mostren que el conjunt de característiques òptim és el que consisteix en les pauses entre paraules, l’entonació a nivell de paraula, i les categories gramatical (POS, part-of-speech). El nostre mètode, que utilitza un mecanisme d’atenció sobre seqüències paral·leles de característiques prosòdiques i morfosintàctiques, indica precisió de F1=70.3% en la generació de la puntuació en transcripcions generades manualment. El sistema que proposem s’ha provat en un sistema de reconeixement de la parla d’última generació i mostra resultats prometedors.

En el segon cas ens ocupem de la millora de la traducció de la llengua oral utilitzant la prosòdia. Presentem un nou mètode per a recopil·lar conjunts de dades bilingües anotades prosòdicament i l’utilitzem per a recopilar un corpus paral·lel anglès-castellà format per 7000 segments de veu alineats i transcrits de la versió original i la doblada d'una sèrie de televisió. A continuació construïm i entrenem un sistema neural de traducció automàtica amb un corpus de text més gran en el domini del cinema i després l'adaptem al nostre corpus amb característiques de pauses afegides. Els resultats mostren que la generació de puntuació prosòdica com a pas previ a la traducció augmenta la precisió de la traducció en un 1% en termes de BLEU. La codificació de les pauses com a característica addicional encara incrementa la precisió en un altre 1%. A més a més, també ampliem el nostre sistema de traducció per a predir conjuntament les característiques de pausa i poder-les utilitzar com a entrada en un sistema de síntesi de veu. Creiem que aquests resultats obren la porta a la traducció expressiva de la parla mitjançant la inclusió de la prosòdia tant en la codificació com en la descodificació del sistema de traducció oral.

El treball dut a terme fa evident la necessitat que els sistemes neuronals d'èxit més recents en el món real tinguin en compte una de les dimensions necessàries en la comunicació humana, com és la prosòdia. La manera de fer-ho és a través de metodologies experimentals i pràctiques i indica que la inclusió de la prosòdia pot millorar aquests sistemes.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

Written form of communication is much newer to the human than spoken communication. Even though it also constitutes a lesser fraction of the ways how humans communicate, it still is the basis of both traditional and computational linguistic studies. This is largely thanks to the discrete form that language took with the invention of the symbology to represent it. This symbology took many forms in time, earlier as images, icons and colors, later as abstract characters assembling words and sentences that constitute what we define as written language. This conversion of paradigm in language facilitated the recording, diffusion, study and automated processing of the human-made information.

Speech inherently carries much more information than the sounds and words that are uttered. Linguistic units like words are enveloped within properly divided measures, with a certain melody that has its highs and lows and even with a rhythm. These "music-like" aspects, which are defined as the *prosody* in language, deals with *how* a certain utterance is delivered. Through the prosodic structure of the language we are able to tell things both about the speaker and the information they want to transmit. For example, which sort of a mode they are in, if they are being serious, ironic or mocking, or where we should pay our attention in their discourse.

Even though an encoding of this dimension of speech seems missing in the written form of the language, it inherently accompanies them both on the way to being written and being read (Chafe, 1988). In fact, as I am writing these words, I imagine explaining my thoughts to the reader with my inner voice prosody. In turn, I trust their own inner voices to enact what I am trying to say and form the same thoughts that I aim to transmit. Even in this form, although prosody is not as present and expressive as in human-to-human communication, words still resonate within ones own melody.

However, one can notice the information loss during this information transformation. The reader can only guess how I want this discourse to sound like and that guess could be completely different than what I intend. A guess that is influenced by the general prosodic structure of the language (English in this case) the reader was exposed to with a bias on their own way of expressing it. Written language gives only a limited set of symbolic tools to convey the information which would normally be encoded with prosody in spoken language. An example of this is punctuation, helping structuring intonational phrasing, marking modalities like question and exclamation, opening subordinate information flow and so on. This role of punctuation, i.e. substituting prosody, has evolved in time and established through grammatical rules and now is considered only partially affected by prosodic structure in language. In fact, written language in general evolved from spoken language so that a neutral prosody is sufficient in transferring of information through this limited symbology, thus minimizing the loss during the conversion.

Because of its discretized structure, written language has been much more convenient for machines to process since human has taught them ways to understand their language. This technological breakthrough, being defined as natural language processing (NLP), has opened ways to collaborate with machines to solve tasks such as translation, information extraction and transcription. These tasks can now be seen applied in great number of ways like spoken/written dialogue systems, automatic captioning, speech-to-speech translation, sentiment analysis etc. As machines function using a pre-defined set of symbols, language in its written form dominates most of these applications. Even in the cases where there is spoken language involved, machines rely on a step where speech has to be converted to text in order to carry on with the subsequent processes in the pipeline.

Just like explained earlier, this conversion means a loss of modality in the language caused by the limited representation of it. An automatic speech transcription (ASR) system acting as the interface to these type of technologies merely transcribes ”what is being said?” and discarding the information on ”how is it being said?”. This eventually affects the quality of the understanding the communicative intention in spoken language as a whole. If we desire to have machines understanding us in a complete way like humans do, this information residing in prosody needs to be considered besides the linguistic information.

1.1. Motivation

The motivation of this dissertation is to explore ways to incorporate prosodic features of speech into systems that involve spoken language understanding. As sketched in the introductory section, the current tendency in these systems is carrying on with text processing once spoken input is converted into written form. This causes an irreversible loss of the information which is essential in human communication. This information is encoded through suprasegmental features of speech which are intonation, tone, duration and rhythm and contributes to the para-linguistic structure in communication. Examples of this para-linguistic structure are: phrasing that helps structure discourse, emphasis, modality defining if the communication is a question, assertion, exclamation or command, information structure that defines givenness etc. Systems that do not account for this information is bound to fail in reaching the level of a human in understanding natural language.

I particularly focus my research on two primal applications that involve conversion of speech into written form: Automatic speech transcription and spoken language machine translation. The former involves conversion of speech into written form and is applied in many type of applications such as dictation systems, automatic captioning and spoken dialogue systems. Spoken language machine translation (SLMT) involves automatic speech transcription as its first step but further translates the transcription into a second language and in some cases synthesize it. Its uses include, for example, automatic subtitling, automatic dubbing of broadcast media and speech-to-speech translation. This type of technology promises growth in the near future due to the huge influx of media created by internet users in the era of Web 2.0. While automated transcription facilitates indexing, tagging and accessibility for the hearing-impaired, automatic translation help reduce language barriers in accessing global information.

Key to machine learning driven development is directly related to the availability of quality data. It shows that, development of speech data annotated with acoustic/prosodic features shows two main challenges. The first challenge is posed by the suprasegmental and multidimensional structure of prosody which makes alignment of linguistic and acoustic features difficult. Although some toolkits exist that assists prosodic annotation of speech data (Rosenberg, 2010), they fall short in terms of applicability in machine learning based approaches. Second challenge is caused by the nature of prosody being directly influenced from context and environment where speech is taking place. Data collected in laboratory environments often fail to reflect expressivity normally present in spoken language, and in turn, influencing models to bias on unnatural data. To aid collection of expressive speech data, a form of collecting “found data” that accommodates prosodic annotation is needed. Scalable methods to process, visualize and store this type of data is also necessary in the process of obtaining this type of data.

The process of automatic speech transcription involves use of an automatic speech recognition (ASR) system to convert the spoken input to text. The raw text output of an ASR system generally lacks any form of punctuation. Depending on the application, punctuation proves to be important for two reasons: Firstly, in the cases where transcriptions will be read by humans, lack of punctuation reduces readability to a large extent. This is demonstrated in the work of Tündik et al. (2018) where watchers of broadcast news were asked to compare punctuated and unpunctuated captions. Both for manually and automatically created transcriptions, punctuated transcriptions were preferred in helping follow the video content. Second case where punctuation has an important role is when the ASR output is further used in subsequent processes like machine translation or parsing. Both these processes require sentence-like units as input and cannot function with long unsegmented text. Furthermore, commas and other punctuation marks that are defined within the orthography of the language pose as important cues in understanding and interpreting the text. Although rules of punctuation is formally defined within the grammatical and orthographic rules of the language, spoken language punctuation is predominantly influenced by prosody. Sentence structures and phrasing are often marked with intonational phrasing and breaks. Modality influences the intonation style of a sentence which in turn influences punctuation. Topic changes are marked with intensity and pitch resets. Emphasized information is delivered with a change in pitch or breaking.

Looking at raw ASR output is often not enough in determining punctuation especially in cases of spontaneous speech. This type of speech often does not follow a regular syntactic structure as in written language, thus making it difficult to determine punctuation based on grammatical rules or data-driven methods that are modeled for written text (Ballesteros and Wanner, 2016). Neural network based work that gets use of prosodic cues report improvement in accuracy of the punctuation marks generated (Tilk and Alumäe, 2016), but still rely on huge chunks of textual data thus biasing models on written language. Also, there is no study reporting the effect of a wider variety of prosodic features into the task.

1.2. Objectives

Enveloped in the motivation that prosody should be incorporated in technology that involves spoken language understanding, I have assigned the following objectives for the course of my research:

- Development of tools that enable creation, prosodic annotation, handling and visualization of spoken language data, based on open sourced resources and publicly available,

- Development of an automated methodology based on open source tools for creating monolingual and bilingual prosodically annotated dataset from prosodically rich found data,
- Compilation and publication of monolingual and bilingual corpora suitable for machine learning based development that involves prosodic-linguistic modelling,
- Evaluation on the effect of various prosodic features on the task of automatic punctuation restoration and subsequent automatized processes like parsing and machine translation,
- Evaluation on the use of raw prosodic features on the task of spoken language machine translation.

My hypothesis is that systems that take spoken language as input will benefit through the recovering of the lost prosodic information during the conversion of speech into text. The research and experimentation involved in this dissertation have this aim in general.

1.3. Outlining the Dissertation

The dissertation is structured in six chapters as follows:

- **Chapter 2** first gives an overview on speech processing systems encompassing the motivated applications of this dissertation. Emphasis is given on deep neural network based setups. Second, state-of-the-art work is presented in the specific topics which this dissertation focuses: Punctuation restoration in automatically generated transcripts and utilization of prosody in spoken language machine translation.
- **Chapter 3** presents methodologies revolving around prosodically annotated corpus compilation and visualization and the corpora used and published within the dissertation's frame.
- **Chapter 4** focuses on the theme of punctuation restoration in raw speech transcripts with a focus on the use of prosodic features. Experiments involve testing of various feature sets incorporated into a punctuation restoration pipeline, evaluation for dependency parsing and accompanying an ASR system.

- **Chapter 5** explores the use of prosodic features within a neural spoken machine translation setting. An expressive parallel corpus collected from a TV series is used to do prosodic adaptation on movie-domain translation. Incorporation of prosody into both the encoding and decoding modules of a speech-to-speech translation pipeline is experimented.

- **Chapter 6** puts the final conclusions on the thesis in terms of the objectives reached and also outlines possible venues for future research and applications.

Chapter 2

STATE OF THE ART

As presented in the introductory chapter, this dissertation focuses on two main applications: automatic transcription and spoken language translation. In the first part of this chapter, I will give an overview on the main research areas that are related to these two topics with a focus on neural network based approaches. Automatic transcription, which is made possible with *automatic speech recognition (ASR)*, is explained in Section 2.1.2. Approaches to the problem of punctuation recovery in ASR output are presented in Section 2.1.3. *Neural machine translation* is presented in Section 2.1.4 with a focus on *spoken language machine translation*. After a brief introduction to text-to-speech synthesis (TTS) systems in Section 2.1.5, I will present the concept of speech-to-speech translation with recent work on its field in Section 2.1.6.

The second part of this chapter reviews relevant work on the inclusion of prosody into these systems. Focus is given on usage of prosody in punctuation recovery in Section 2.2.1 and adding prosodic modalities on spoken language translation in Section 2.2.2.

2.1. Neural Speech Processing Overview

A speech processing system consists of at least one of the following modules: Automatic Speech Recognition (ASR) for converting verbal communication into discrete symbolic form (i.e. text), text-to-speech (TTS) system for generating information in spoken form, and a spoken language understanding (SLU) system for mapping between actions and verbal utterances (Huang et al., 2001). Depending on the application, versions and combinations of these systems are employed to solve the task involved with it. For instance, an automatic subtitling system

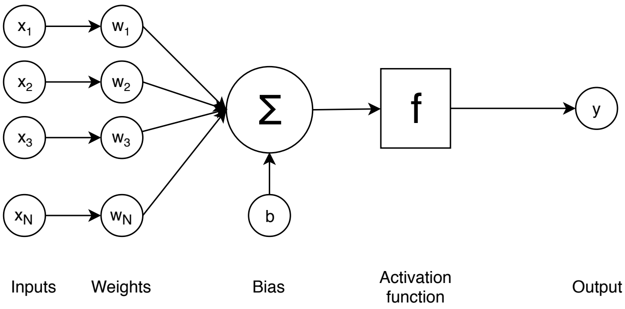


Figure 2.1: An artificial neuron.

would involve an ASR system together with a speech activity detection module to transcribe the spoken parts in a media. If the subtitling is to be done in another language, the same pipeline would be followed by a machine translation system. A complete speech-to-speech pipeline would result in an automatic dubbing system where translated content would be synthesized using a TTS system.

Research on these subareas of speech technology has recently experienced a great shift towards the usage of *artificial neural networks (ANN)*. Popularity of ANN in general has risen in the recent years mostly due to advancements in computing power. Specifically, training of large and deep neural networks in a reasonable amount of time has been made possible with *Graphical Processing Units (GPUs)*.

I will introduce briefly the concept of *deep neural networks (DNNs)* as they form the basis of the experimentation presented in this dissertation. The information presented in this section can be consulted in Katagiri (2000) and Goodfellow et al. (2016) for a deeper understanding.

2.1.1. Deep Neural Networks

An artificial neural network consists of a group of nodes that are modeled after the neurons in a biological neural system and connections between them modelling the axons in the neural system. Figure 2.1 illustrates the structure of one neural node, which is also referred as *perceptron* or simply *neuron*. Each connection towards a neuron is an input (x_i) and is associated with a weight coefficient (w_i). The basic function of a neuron defines the input signal to the neuron as:

$$a = \sum_i w_i x_i + b \tag{2.1}$$

The input signal is then passed into an activation function to produce the output y.

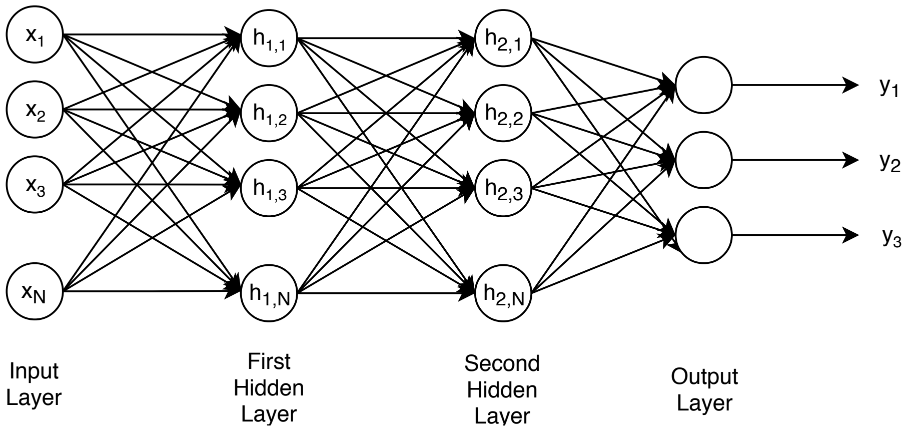


Figure 2.2: A fully connected feed-forward neural network with two hidden layers.

$$y = f(a) \tag{2.2}$$

Activation function is a differentiable function that resembles a step function so the neuron *fires* with certain input. The original Rosenblatt's perceptron had Heaviside step function as the activation function (Rosenblatt, 1958). Activation functions commonly being used today are the *sigmoid* and the *hyperbolic tangent* functions.

One single neuron is evidently not sufficient for modelling complex functions. A basic neural network consists of a layer of input neurons fully connected to a layer of output neurons. This setting produces an N-to-M mapping. Extra layers are added between the input and output layers to introduce even more complexity to the network. These layers are called *hidden layers* and are fully connected between each other between input and output layers. An illustration of a neural network with two hidden layers is given in Figure 2.2. Number of hidden layers can be determined according to the task-at-hand. A neural network with more than one hidden layer is called a *deep neural network* (Goldberg, 2016).

Although there exist many types of neural network taxonomies, one important characteristic that divides neural network architectures into two is the direction of the signal flow in the network. A *feed-forward* architecture (as in the example in Figure 2.2) allows information to be passed in only one direction. Whereas a *recurrent neural network (RNN)* architecture allows the output signal of some nodes to be passed again to a neuron coming previously, or to the neuron itself. Recurrent neural networks are especially suitable for representing time-series data. Because of this, it is currently being preferred as the principal architecture on

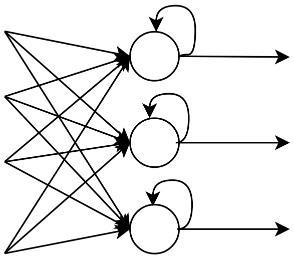


Figure 2.3: Information flow in a recurrent neural network (RNN). Output signals are allowed to go back as input signals to the neurons.

many state-of-the-art applications of machine translation, speech recognition and speech synthesis.

As it can be seen in Figure 2.3, a neuron in a RNN can have its output connected back to itself as an input. This model allows the neuron to keep a form of a *memory* from previous inputs and decide on the next output according to it together with the current input. Another advantage the RNN architecture gives is the ability to represent different mappings between nodes at each time step. Figure 2.4 shows types of connections between *RNN cells*, which can be one-to-one, one-to-many, many-to-one or many-to-many. An application to a many-to-many network would be translation, where a sequence of words are encoded and then decoded in another language. An example of a many-to-one network would be e.g. voice classification given an acoustic feature sequence. This group of neural networks are sometimes called *encoder-decoder networks*. Many-to-many type RNN is sometimes referred as *sequence-to-sequence network*.

Neural network training involves the following steps: Introduction of samples in the training set to the network, computing the error with help of a *loss function* comparing desired output to the networks output, computing the gradient given by the error and then shifting the network weights in the direction and magnitude of the gradient. One common method used in performing this operation is *stochastic gradient descent (SGD)* (Goodfellow et al., 2016; LeCun et al., 1998).

Although the aim of training is to minimize the total loss given by all training samples, optimization is performed at introduction of each sample. To avoid unnecessary amount of fluctuations (i.e. noise) in weight updates, samples are generally inputted in batches, an average loss is calculated and then network is updated according to that average loss.

A key step in training is the step where gradients are calculated according to the loss that the network gives. This is performed through a method called *backpropagation algorithm*, which is the process of going backwards through the

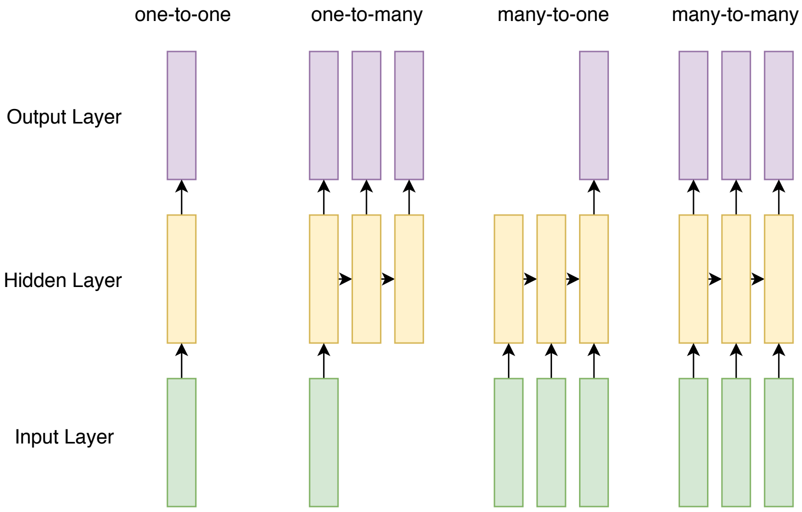


Figure 2.4: Four types of RNNs in terms of input and output type.

network calculating the partial derivative of the error at each weight (Rumelhart et al., 1988). These derivatives are then used to update the weights using a method called *gradient descent*. This update is scaled by the *learning rate*, which is a predetermined *hyperparameter* that is sometimes allowed to decay in time to enable fine-tuning of the network during later steps of the training.

2.1.2. Automatic Speech Recognition

In its most simple sense, automatic speech recognition (ASR) is the conversion of speech in its acoustic form into a symbolic form such as words or letters. It is the probabilistic modelling of the question ”What is the most probable word sequence among all possible word sequences given an acoustic input?”. Figure 2.5 illustrates this process. Speech signal captured by microphone is first encoded into a sequence of acoustic feature vectors. Following, the acoustic feature vectors are decoded into the words that represent the linguistic information that lies in the speech signal.

Classical approaches to ASR employ a modeling of spoken language that uses Gaussian mixture model-hidden Markov model (GMM-HMM). HMM is a powerful statistical method for representing time-series data (Huang et al., 2001; Rabiner, 1989). A GMM-HMM ASR system has a modular architecture: The feature extraction step converts the input speech signal into a sequence of fixed size

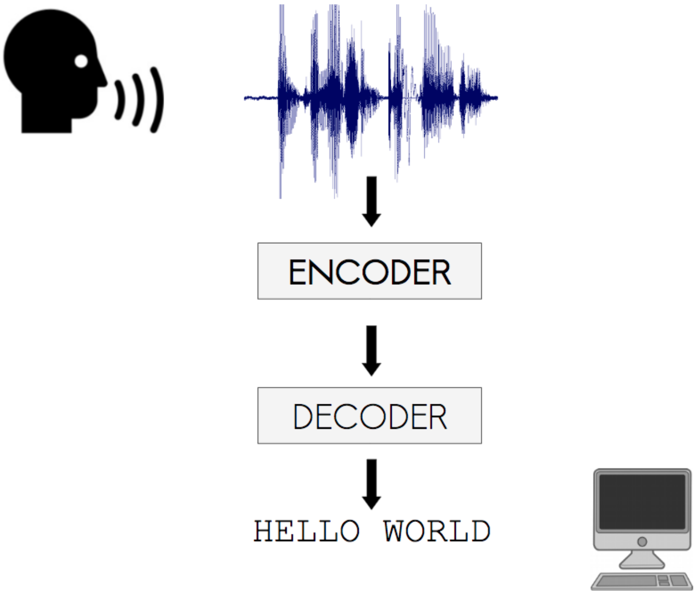


Figure 2.5: Speech recognition is the conversion of an acoustic signal with spoken language into its written form.

acoustic vectors. Later, the decoder makes use of the acoustic model, the language model and the pronunciation dictionary in order to decide the most likely word sequences they represent. Acoustic and language models are trained with a corpus of transcribed speech samples and a text corpus. While the acoustic model stores the information of the statistical behaviour of the sounds in a language, the language model stores the likelihood of the tokens (words) occurring and co-occurring in a language.

ASR systems experienced a breakthrough with the use of deep neural networks from 2012 on with its introduction in Dahl et al. (2012). The DNN-HMM ASR replaced the feature representation step that used Gaussian mixtures with a recurrent neural network architecture. This configuration replaced the modular architecture that separated language and acoustic models in the traditional setup and thus minimized the complexity of the system. The graphical comparison of two models is illustrated in Figure 2.7. The DNN-HMM based ASR showed an improvement of 20% compared to the GMM-HMM based.

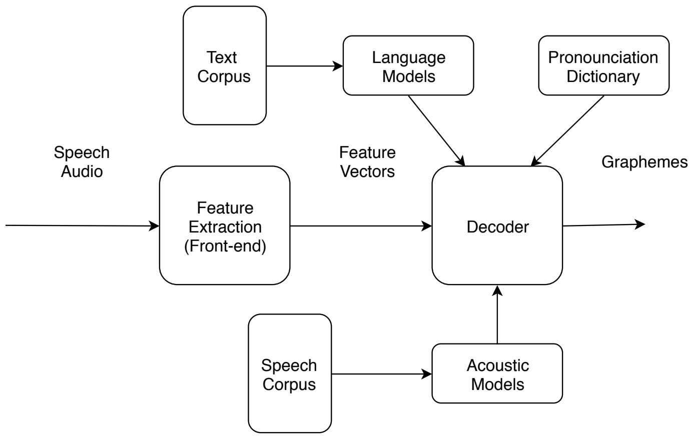
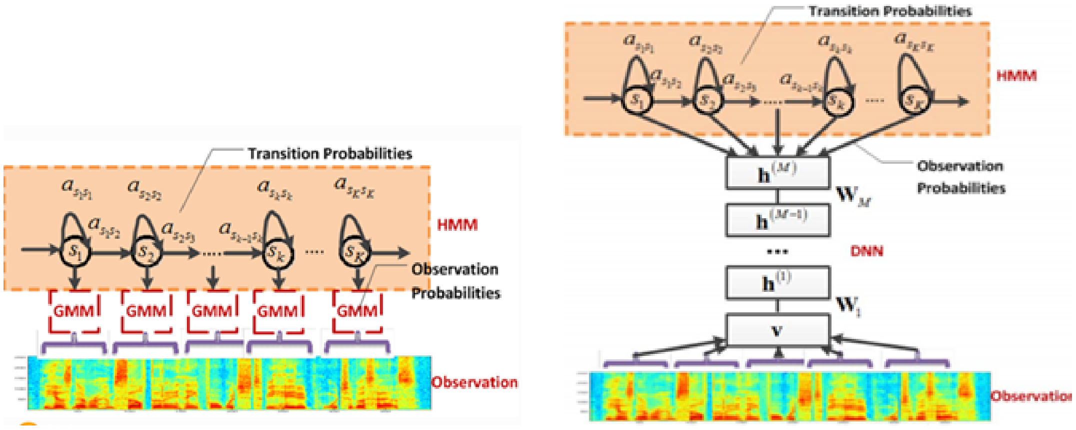


Figure 2.6: General architecture of a traditional ASR system.

2.1.3. Punctuation Restoration in ASR

As applications of automatic speech recognition vary greatly, the objective of ASR is only focused on the recognition rate of the words. Aspects such as capitalization and punctuation, which are crucial elements for readability of the ASR output, is generally considered apart from an ASR system. For example, for applications such as automatic captioning or transcript extraction, punctuation and capitalization prove to be essential for improving readability. Tündik et al. (2018) evaluate the effect of presence of punctuation in captions from a end-user perspective and show that punctuated captions are easier to read both when transcriptions are manually or automatically generated. Another case where punctuation proves to be essential is when subsequent processing steps in a pipeline are optimized to work with it. Syntactic or semantic parsing, which is an important module in dialog based systems necessitates input segmented into sentence-like units to function. Most machine translation systems are trained with single sentence input. Furthermore, it is proved that both of these processes function better with properly placed in-sentence punctuation and especially commas (Vandeghinste et al., 2018; Jones, 1994).

The problem of punctuation restoration has been addressed in several works in the literature —as has been the closely-related issue of boundary detection. Both problems have been tackled from diverse perspectives. In terms of which types of features are used, the approaches fall into three categories: (1) models based only on only textual (lexical and syntactic) features, (2) models based only on only prosodic/acoustic features and finally (3) models where both textual and acoustic/prosodic features are used. In this section I will focus on models based



(a) GMM-HMM

(b) DNN-HMM

Figure 2.7: Comparison between GMM based (a) and DNN based (b) ASR (Figure from Dahl et al. (2012))

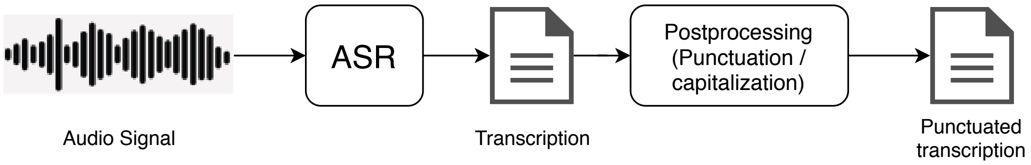


Figure 2.8: Punctuation and capitalization as a postprocessing step after ASR.

only on textual features. That is, as illustrated in Figure 2.8, punctuation process is only applied on the raw ASR output without any other cues.

Punctuation using only textual features is relevant when e.g. punctuation restoration is needed for written data (Jakubıcek and Horák, 2010) or in the case when corresponding audio information is lost (Lu and Ng, 2010). In Jakubıcek and Horák (2010), for instance, the punctuation detection is addressed from a syntax-based perspective by using the output of an adapted chart parser, which provides information on the expected punctuation placement. In Ueffing et al. (2013), several textual features including language model scores, token n-grams, sentence length and syntactic information extracted from parse trees are combined using conditional random fields (CRF). They demonstrate that syntactic features help only when the input language is well-structured (as e.g. newspaper texts). In Lu and Ng (2010), the task is based on dynamic conditional random fields and applied to a conversational speech domain where sentence boundaries and types are detected.

Another reason that facilitates the usage of solely textual features is the abundance of well-punctuated written data. Using a purely text-based n-gram language model, Gravano et al. (2009) demonstrate the performance improvement induced by large textual training in punctuation detection and capitalization. Although narrow-range grammatical constructions are recognized well for comma and period placement, n-gram approach fails in discovering long-range dependencies for the correct placement of question marks.

Punctuation placement is also approached as a monolingual machine translation problem in Peitz et al. (2011); Cho et al. (2017); Paulik et al. (2008) where target sequence is the punctuated version of the source sequence. Also, in Klejch et al. (2017), frame-level prosodic features (only pitch and pause) are integrated in a neural machine translation based system with a hierarchical encoder.

Recently, recurrent neural networks are shown to be suitable for punctuation restoration for their ability to capture long-range dependencies in sequential data. Ballesteros and Wanner (2016) introduce a language-independent model with a transition-based algorithm using LSTMs (Dyer et al., 2015), without any additional syntactic features. Treviso et al. (2017) experimented with different word embeddings model within an RNN setting and proves that a good word embeddings model improves punctuation restoration accuracy.

2.1.4. Neural machine translation

Machine Translation is defined as the automatic conversion of a sequence of symbols in one language to a sequence of symbols in another language (Goodfellow et al., 2016). It has evolved through years from rule-based systems (RBMT) to statistical approaches (SMT), which modeled the probabilities of mappings between sub-phrases of various sizes. These probabilities are learned in a statistical fashion from *parallel texts* where sentence aligned translations are available in the languages involved (referred as source and target languages).

Neural machine translation (NMT) quickly replaced SMT in the recent years for its relatively simpler architecture. It no longer needed fine-tuning of smaller components involved in a SMT system and directly learned mapping from input text sequence to the target text sequence (Bahdanau et al., 2014; Wu et al., 2016).

A commonly used architecture for NMT is the encoder-decoder architecture, which was introduced earlier in this section[3]. As illustrated in Figure 2.9, token sequence in the source language input through an encoder is sent over a decoder to output tokens of the target language. Similar to the data-driven approach of SMT, this network is trained with parallel text to maximize the probability of a correct translation given a source sentence (Bahdanau et al., 2014).

[3] There exist more types of neural architectures for machine translation task such as the *Transformer*. They are not reviewed in this section as they are not used in this dissertation.

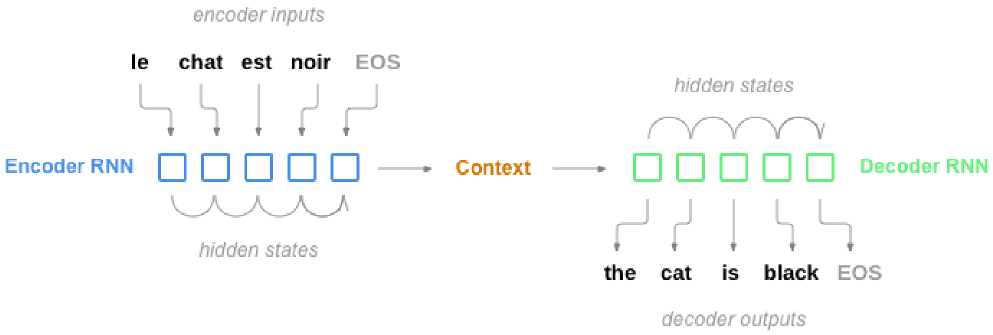


Figure 2.9: Architecture of an encoder-decoder neural machine translation system. (Diagram taken from *spro*'s sequence-to-sequence translation tutorial on github[1])

One weakness that this model introduces is the connection between two RNNs that squeezes the input sequence into one single-length vector before being decoded as target token sequence. This is analogous to reading a phrase from beginning to end and then translating it into another language without looking at it again. Normally, a translator would break a input sentence into smaller portions and translate step by step giving attention to a different parts each time. An analogy of this approach was implemented in NMT with the introduction of *attention mechanism* (Bahdanau et al., 2014; Luong et al., 2015). As illustrated in Figure 2.10, the attention mechanism helps focus on different parts of the input at each step of decoding. This relieves the decoder from having to predict target language tokens in one go without any spacial context of the input phrase (Wu et al., 2016).

Spoken language machine translation is a type of MT where input and/or output to the system is spoken language. Spoken input translation can be employed through the usage of ASR prior to MT and translation can be generated as speech with a TTS to obtain spoken output. Machine translation with spoken input introduces its own specific challenges. For example, written and spoken language show linguistic differences thus models trained on one domain do not necessarily serve to be optimal in the other domain.

One challenge that spoken language translation introduces is the possible incompatibility between ASR output structure and MT input structure. MT models are usually trained with sentence-like structures as samples and therefore show low performance on partial sentence or long sequences of words as input. In text translation domain, processing of long text documents is performed by translating it sentence by sentence using punctuation information as segmentation cues. A similar approach needs to be followed when input is spoken utterances as well. Figure 2.11 TODO[this figure is two pages later: too far from text] illustrates an

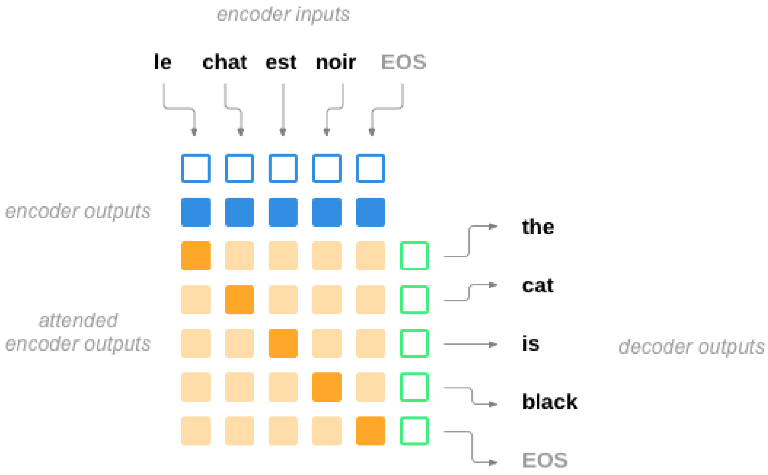


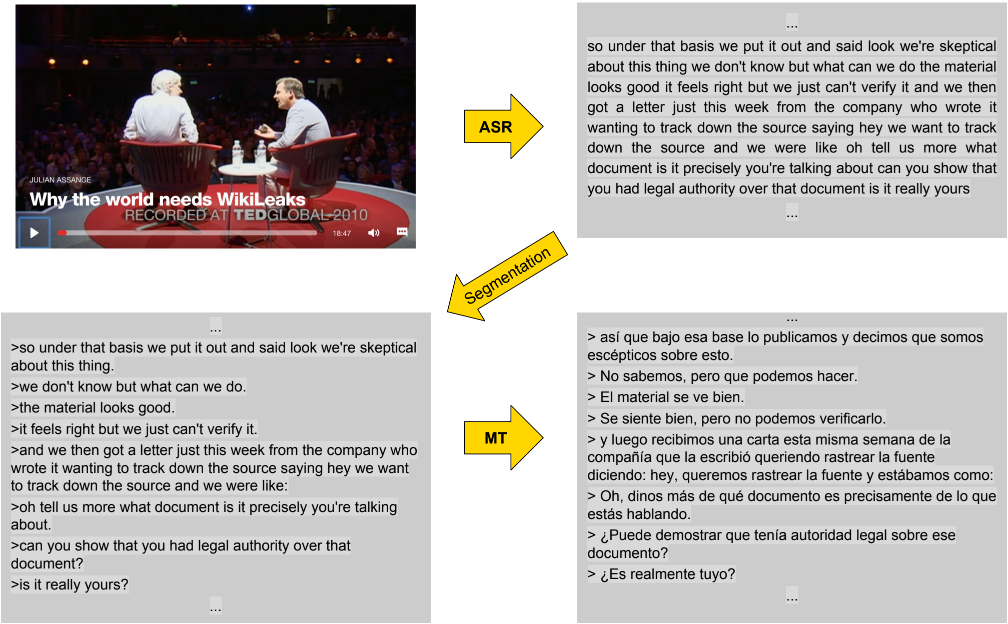
Figure 2.10: Attention mechanism in encoder-decoder NMT architecture keeps track of portions of the input sequence that affects each decoder output. (Diagram taken from *spro*'s sequence-to-sequence translation tutorial on github[2])

example of spoken language translation of a conference talk. A standard MT system would be unable to translate the unsegmented transcription of the talk. Translation is made possible only through a segmentation process, such as boundary detection or punctuation restoration.

Another characteristic involved in spoken language translation is the loss of prosodic information in the transcription step of a classic spoken language translation system pipeline. Prosodic features that would normally give cues for target translations e.g. through resolving ambiguity or phrasing are left out.

2.1.5. Text-to-speech synthesis

Speech synthesis involves production of a human-like speech given a text input with computational methods. Before the advent of deep learning, there were two main approaches to text-to-speech (TTS) synthesis: concatenative TTS, and parametric TTS. Concatenative TTS is based on the technique of combining short audio clips together to synthesize the desired text. It is a highly complicated process building concatenative TTS systems and its results generally lack naturalness. In contrast, parametric TTS relies on statistical methods by generating speech with a combination of parameters like F0 and energy. Figure 2.12 illustrates the workflow of a parametric TTS system. First, morphemes in the input text are converted to phonemes through a linguistic analysis. Next, features like



...
so under that basis we put it out and said look we're skeptical about this thing we don't know but what can we do the material looks good it feels right but we just can't verify it and we then got a letter just this week from the company who wrote it wanting to track down the source saying hey we want to track down the source and we were like oh tell us more what document is it precisely you're talking about can you show that you had legal authority over that document is it really yours
...

...
>so under that basis we put it out and said look we're skeptical about this thing.
>we don't know but what can we do.
>the material looks good.
>it feels right but we just can't verify it.
>and we then got a letter just this week from the company who wrote it wanting to track down the source saying hey we want to track down the source and we were like:
>oh tell us more what document is it precisely you're talking about.
>can you show that you had legal authority over that document?
>is it really yours?
...

...
> así que bajo esa base lo publicamos y decimos que somos escépticos sobre esto.
> No sabemos, pero que podemos hacer.
> El material se ve bien.
> Se siente bien, pero no podemos verificarlo.
> y luego recibimos una carta esta misma semana de la compañía que la escribió queriendo rastrear la fuente diciendo: hey, queremos rastrear la fuente y estábamos como:
> Oh, dinos más de qué documento es precisamente de lo que estás hablando.
> ¿Puede demostrar que tenía autoridad legal sobre ese documento?
> ¿Es realmente tuyo?
...

Figure 2.11: Spoken language translation demonstrated on a conference recording.

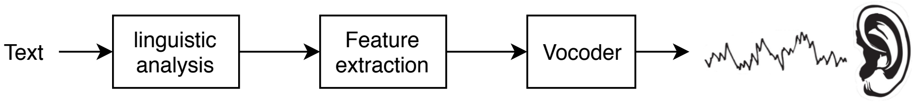


Figure 2.12: Basic workflow of a text-to-speech system.

cepstra, F0, duration, break are calculated to be fed into the *vocoder*. The vocoder finally generates the waveform using these parameters.

What makes relevant TTS in this dissertation is that the Vocoders can take additional parameters in some cases to generate samples with desired features. This is particularly useful in speech-to-speech translation applications where transfer of input prosodic cues to the synthesized target speech is desired. State-of-the-art TTS applications such as *IBM Watson TTS*[4] allow this kind of additional parameters to be input in form of *SSML tags*. An example of an input text to a Spanish TTS system with additional prosodic conditioning is shown below:

```
<p><s>Consciente de su patrimonio espiritual y moral<break time
    ="300ms"/>, la Uni[U+FFFD]n est[U+FFFD] fundada sobre los
    valores indivisibles y universales de la dignidad humana, <
```

[4] https://text-to-speech-demo.ng.bluemix.net/

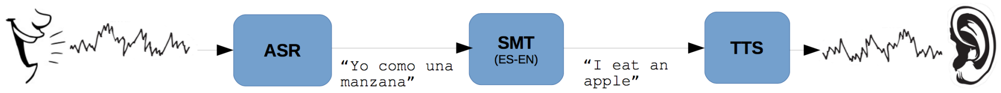


Figure 2.13: A conventional speech-to-speech translation pipeline.

```
prosody rate="-15%"> la libertad, la igualdad y la
solidaridad, </prosody> y se basa en los principios de la
democracia y el Estado de Derecho<break time="500ms"/>.</s> <
s><prosody rate="+20%">Al instituir la ciudadan[U+FFFD]a de
la Uni[U+FFFD]n </prosody> y crear un espacio de libertad,
seguridad y justicia, sit[U+FFFD]a a la persona en el centro
de su actuaci[U+FFFD]n.</s></p>
```

2.1.6. Speech-to-speech translation

Speech-to-speech translation enables human-to-human communication where each of the agents involved speaks in a different language. A device capable of enabling such a communication is able to accept spoken input in language A, translate it to language B and then synthesize it for hearing. By performing this process in both ways, it acts as an interface for a turn-based inter-lingual communication. Conceptually, such a system is the concatenation of the three following processes: (1) Automatic speech recognition, (2) machine translation, and (3) text-to-speech synthesis. A diagram of the one-way process in S2S translation is illustrated in Figure 2.13.

There exist various examples of S2S translation solutions resulting from both academic and commercial research. *Verbmobil* is considered as the pioneer in the field as it is the oldest and most extensive research project dealing with S2S translation (Wahlster, 2013). It was designed for translation of spontaneous dialogues in mobile situations for the languages English, German and Japanese. *IBM MASTOR* was developed in a defense oriented framework for facilitating spoken communication in low-resource languages (Gao et al., 2006). *TECNOPARLA* was developed with the motivation of spoken translation in the broadcast radio and television domain for the languages Catalan, English and Spanish (Schulz et al., 2008). *EMIME* project was the first work that aimed voice personalization through S2S translation, where synthesized voice is adapted to sound like the recognized voice (Kurimo et al., 2010). Two projects with Swiss origin, SP2 SCOPES (Szaszak et al., 2014) and SIWIS (Garner et al., 2014) that focus on Swiss and Eastern European languages report cross-lingual prosodic transfer as their main objectives. Although parallel speech corpora has been published from these projects, there are no recorded results on the accomplishment of these objectives.

Spoken parallel corpora

Corpus	Languages	Speech style
EPIC	EN/IT/ES	spontaneous/interpreted
MSLT	EN/FR/DE	constrained conversations
EMIME	FI/EN, DE/EN	prompted
EMIME Mandarin	ZH/EN	prompted
SP2-Speech-Corpus	EN/FR/DE/HU/MK/SR	prompted with emphasis
Japanese-English emphasis	JA/EN	prompted with emphasis
SIWIS database	EN/FR/DE/IT	prompted with emphasis
MDA (Almeman et al., 2013)	Four Arabic dialects	prompted
Farsi-English (Melvin et al., 2004)	FA/EN	read/semi-spontaneous

Table 2.1: A selection of available parallel speech corpora for use in S2S translation.

The availability of large parallel corpora is one of the major challenges in developing machine translation systems. Bilingual corpora, which are needed to train statistical translation models, are harder to acquire than monolingual corpora since they presuppose the implication of labour in translation or interpretation. Working on the speech domain introduces even more difficulties since interpretations are not sufficient to capture the paralinguistic aspects of speech. Several attempts have been made to compile large spoken parallel corpora. Some of these corpora published in literature are listed in Table 2.1. All of them have been manually compiled in laboratory conditions but show some differences in certain aspects. The EPIC corpus has been compiled from speeches from the European Parliament and their interpretations (Bendazzoli and Sandrelli, 2005). The EMIME database is a compilation of prompted speeches to serve for the task of speaker conversion (Wester, 2010). The MSLT corpus has been collected in bilingual conversation settings, but there is no one-to-one alignment between sentences in the different languages as they are lightly guided conversations (Federmann and Lewis, 2016). There is a number of corpora collected for projects focusing on the emphasis translation task: SP2 Speech Corpus (Sečujski et al., 2016), SIWIS database (Goldman et al., 2016) and the database collected by Truong et al. (2014). These corpora contain sentence recordings with acted emphasis on the same word or word groups in both languages.

2.2. Prosody in speech processing

In previous chapter I gave a review of the systems where spoken language is being processed to serve for a certain purpose. However, in these systems, speech is considered only with the linguistic content (i.e. words, phrases, etc.) it carries. Prosodic features that are encoded through various acoustic phenomena like intonation, energy, breaks etc. are disregarded in any further analysis.

In this section, I will review recent as well as some historical works that regard prosody as an essential dimension in a speech processing framework. These works not only argue that prosodic features in spoken language are important for spoken language applications, but also suggest methodologies for their inclusion and report progress through it.

I will present two applied areas where prosodic cues are utilized as an advancement for speech processing systems. First in automated speech transcription where effect of prosodic cues in punctuation restoration is investigated, then in speech-to-speech translation where a complete linguistic and paralinguistic information transfer is desired.

2.2.1. Utilizing prosody in punctuation restoration in transcribed speech

It has been shown that prosodic features are highly indicative of sentence boundaries as well as of punctuation placement in many works (Nunberg, 1990). Therefore, a great deal of effort has been put in several works into the use of prosodic features when original speech is available. In Levy et al. (2012), the authors successfully detect automatically full stops in ASR output with no language modeling using only weighted pause, f0 changes and amplitude range values. Commas are shown to be more difficult to detect when only prosodic features are used. In Baron et al. (2002), it is demonstrated that combination of language and prosodic models performs better than single-model approaches.

Many studies consider punctuation restoration as a problem of determining the probability of a certain label at a boundary point in speech, e.g. between words or at pauses, calculated in the vicinity of that point. Prosodic and textual cues around each inter-word boundary are taken as features for a decision tree classifier to detect sentence boundaries in Liu et al. (2006). Similarly in Khomitsevich et al. (2015), word and grammatical n-gram features are combined with prosodic features to detect punctuation marks in Russian ASR system. Kolář et al. (2004) focus on Czech broadcast news speech to detect commas and sentence boundaries by using a prosodic model based on decision trees and language model based on n-grams.

A combination of lexical-, prosodic-, and speaker-based features is also found in Batista et al. (2012) for the detection of full stops, commas, and question marks in a bilingual English-Portuguese broadcast news corpus. Similar works deal with the punctuation generation problem by using statistical models of prosodic features (Christensen et al., 2001), the combination of both textual and prosodic features based on adaptive boosting (Kolář and Lamel, 2012), and a cross-linguistic study of prosodic features through two different approaches for feature selection: a forward search wrapper and feature filtering (Fung et al., 2007).

Che et al. (2016); Schlangen and Hough (2017); Batista et al. (2012)

Combining lexical and prosodic models has been employed in a bidirectional neural network setting in Xu et al. (2017) for sentence boundary detection and in Tilk and Alumäe (2016) for punctuation restoration. Both approaches are based on training of the language model (on large amounts of textual data) separately to the acoustic model (from a smaller corpus), eventually leading the models to bias on written data.

2.2.2. Utilizing prosody in spoken language machine translation

There has been considerable work on inclusion of prosody into speech-to-speech translation pipelines. Most of the research based systems give some of the focus onto this area as it is believed that spoken translation is truly complete only through conveying of prosody as well as linguistic information between source and target phrases. On another aspect, some research focus on the fact that ASR output is not optimized to be inputted to machine translation. ASR outputs only a raw sequence of words without any further information on sentence or phrase boundaries and thus harms MT that necessitates a certain input size and context.

It is observed that there are three main objectives when it comes to incorporation of prosody into a S2S framework. These objectives are: (1) segmentation of the source phrase into meaningful units through use of prosody to aid the machine translation step, (2) transfer of prominence (emphasis) in input speech into the synthesized translations and (3) using context information to boost translation accuracy.

First use of prosody within a spoken language translation system was within the *Verbmobil* project (Noth et al., 2000). A group of prosodic features were computed for the word hypotheses computed by the ASR module. These were: probabilities for clause boundaries, accentuation and sentence mood. Among these, a major improvement was achieved through the classification of clause boundaries in the input phrases. Syntactic parsing of the recognized words was improved in terms of readings and computation time only through the segmentation of the input phrases. Boundary classification was based on a combination of textual and prosodic features (energy, duration and F0). A similar approach is investigated in Matusov et al. (2007). A lexical-prosodic boundary prediction algorithm is introduced and compared with various other segmentation algorithms in terms of their effect on translation quality. They show that translation is optimized through usage of a boundary prediction algorithm based on prosodic features and phrase probabilities using a language model.

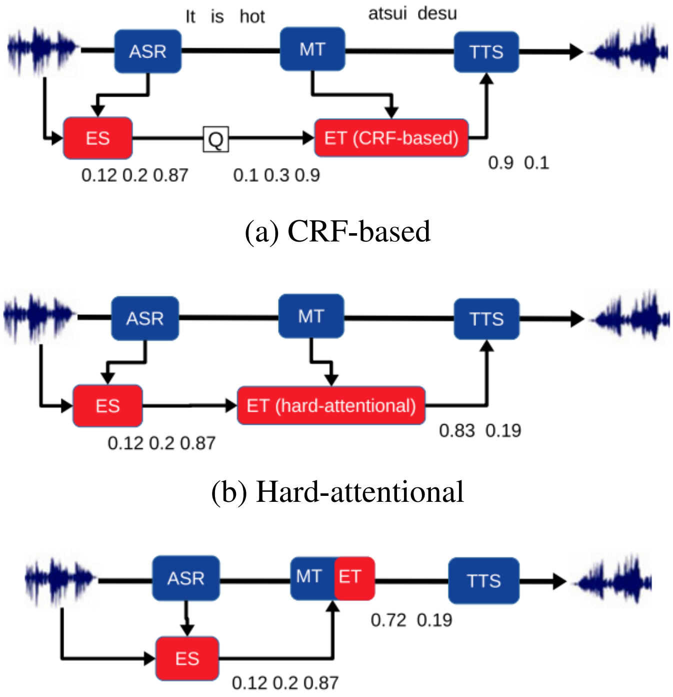
(Agüero et al., 2006) can be considered as the first example where objective is to transfer the underlying paralinguistic features in the source speech to the synthesized target speech. The methodology they present aims to find transfer patterns of F0 contours in source and target speech in a S2S framework. This is done with an extension on the intonation prediction module of the TTS that does not only consider linguistic features of the target translation but also features derived from the source speech. The intonation patterns of phrases of the input sentence is first classified and then mapped into intonation patterns of the target language. These transformations are learned from a bilingual corpus and integrated as an enhancement to a phrase-based translation system. They report improvement over preferences of the synthesized translations in terms of naturalness.

A similar approach is followed in Anumanchipalli et al. (2012) for word-level emphasis transfer. They explain their motivation with experimentation on a subset of the bilingual speech corpus they collected. By manual inspection, they see that there's a match of 48% of the emphasized words in the parallel languages. Their cross-lingual intonation transformation methodology is based on learning the mapping between word-level intonation contour parametrizations between two languages from a single-speaker bilingual dataset. Since they do not perform the machine translation itself, they are able to compare intonation contours generated in a neutral way and with their enhancement. They show that through this process generated contours get closer to the reference contours in their dataset.

Do et al. approach cross-lingual prosodic transfer from a perspective based on transferring of word-level emphasis. Their general approach is to label each word in the source token sequence with a real-numbered emphasis level and then map it into the words in target sequence using a transformation function. Emphasis modelling is performed with linear-regression hidden-semi Markov models (LR-HSMMs) that is trained on F0, duration and energy features (Do et al., 2017b). Their methodology for mapping input emphasis estimations to target emphasis weights show change over various works. In Do et al. (2017b), this is performed using a model based on conditional random fields (CRFs) (Figure 2.14a). Following, in Do et al. (2016), they utilize a LSTM based model with attention that exploits the word alignment information of machine translation. They record an improvement of 1% in terms of emphasis prediction F-measure (Figure 2.14b). Both of these approaches incorporate prosodic transfer process as an additional module besides MT and assume perfect translations. This is later addressed in Do et al. (2017a) and Do et al. (2018) where emphasis and word prediction are done jointly within a sequence-to-sequence MT system (Figure 2.14c). The lack of parallel spoken data is covered by a two-stage training procedure. Translation model is first trained on a large text corpus and then emphasis modelling is generated from a smaller laboratory generated English-Japanese parallel corpus. Their results show that text translation does not improve with inclusion of emphasis weights. As a simpler system is introduced, gain on computational time is recorded, however, without an improvement on emphasis prediction compared to previous works. All in all, they report that their models are speaker dependent and is demonstrated on a highly controlled setting. This can be explained by the corpus they use at hand which consists of a small set of samples with acted emphasis.

Pausing in speech is an important prosodic feature that affects both emphasis perception and phrasing. Transfer of pauses within S2S translation is addressed in the works: Do et al. (2015) and Agüero et al. (2018). In the former one, pause prediction is incorporated into the CRF-based emphasis prediction module and shows improvement in terms of emphasis perception in the synthesized examples. The latter work focuses on the transfer of the phrasing and follows a rule-based approach exploiting alignment information from SMT.

Some work on use of prosody in S2S translation focuses on employing prosodic features available through acoustic or linguistic analysis to further boost translation accuracy. These works are mostly inspired from approaches where factored SMT is enhanced with linguistic features (e.g. POS features) and employ a similar approach in spoken translation. Guo et al. (2016) is an example where additional prosodic features based on pronunciation, boundary marks and emphasis is integrated as factors to a factored translation model based system. They record slight improvement in terms of translation accuracy with inclusion of boundary marks when translated from Chinese to English. In the opposite direction, they record improvement with inclusion of all three features. Again in Rangarajan Sridhar et al. (2013), factored translation models used for phrase-based translation is extended to accept additional prosodic information on source and target sides. They test inclusion of dialog information such as question types in source side and pitch accent based prominence features on the target side. Modest improvements are recorded in terms of translation accuracy.



(a) CRF-based

(b) Hard-attentional

(c) Joint model

Figure 2.14: Various implementations of S2S translation systems with emphasis transfer. (Diagrams are taken from Do et al. (2018))

Chapter 3

COMPILING CORPORA FOR BUILDING DATA-DRIVEN PROSODIC MODELS

In order to develop data-driven models related to speech prosody, one needs access to a sufficient number of speech samples that are annotated with prosodic features. Spoken samples collected to form a corpus serve for developing models for machine-learning applications as well as empirical research. A prosodically annotated spoken corpora usually consists of speech samples, their transcriptions and certain acoustic and prosodic labels associated with it. There is a few number of publicly available speech corpora that serve for prosody research as it is hard to process and annotate (Rosenberg, 2018). One major work that this dissertation involves is collection of corpora to develop prosodic models on spoken language applications. This chapter presents: two prosodically annotated corpora that were used during this work, methodologies followed in compiling them and also further tools that were used and developed during the process.

There are two different methods for compiling spoken language corpora. First approach involves recording of designated speakers reading prepared text material in a controlled environment. Although this approach is the best way to obtain noiseless data, it is very expensive and hard to re-scale. Moreover, it poses a further disadvantage if prosody is an important aspect. As speakers are placed in a controlled environment their speech lacks the prosodic features that would normally be present in a more natural setting.

Another approach to corpus development lies in exploiting readily available recorded media material such as conference talks, TV shows or movies. TODO:[The concept of ”found data” is used in theses cases. [1]]This kind of data source is more prone to noise and necessitates various automated and manual processing methodologies to shape it to the need of the application need. However, two big advantages that this method gives are that (1) the speakers can have more naturally expressive prosody, and (2) relatively low-cost scalability that comes with development of automated extraction methods.

For the sake of obtaining data to be used in the methodologies presented in this dissertation, automated approaches that exploit readily recorded material were followed Öktem et al. (2017); Öktem et al. (2018b). Two main sources of spoken data have taken advantage of: conference talks and subtitled movies and TV shows. The methodologies developed to process this material were compiled as open-source software libraries accessible online[2] (Sections 3.1 and 3.3). Two corpora were obtained through the result of these processes: First one is the re-compiled and published *TED talks corpus*, which is modified from Farrús et al. (2016) to suit experiments related to prosodic punctuation recovery task (Section 3.2). Second is the *Heroes corpus*, which consists of parallel English and Spanish speech segments gathered from a TV series to suit prosodic translation task (Section 3.4). Both of these corpora are made accessible openly through UPF Digital Repository: `repositori.upf.edu`.

Another task as important as obtaining of prosodically annotated corpora is analyzing them in terms of various prosodic features. The nature of prosodic data introduces its challenges and thus necessitates specialized tools to accommodate its analysis. In this chapter, I will also present *Prosograph*, which helps analyze data of this type in a simple and clear way (Section 3.5).

3.1. Toolkit for Prosodically Annotated Speech Data Creation

In this section, I will introduce some of the principal tools employed that served an important role in the presented corpus development processes.

Proscript for Prosodic Data Handling

Handling speech data together with prosodic annotations introduces its own challenges. Prosody can be seen as a phenomena parallel to the words uttered

[1] See e.g. http://tts.speech.cs.cmu.edu/pbaljeka/www/Thesis_Final.pdf

[2] `www.github.com/alpoktem`

in speech. In its essence, written representation of speech only is only sufficient in representing a limited dimension of it through orthographic symbols. There has been work to represent aspects of prosody together with the written form of speech (orthography). For example, ToBI convention introduced in Silverman et al. (1992) represents speech prosody in 4 tiers. These four tiers are agreed annotation styles for representing intonation, accents and breaks in correspondence with utterance.

Computational applications that deal with prosody necessitate a standard in digitally representing the structure of speech with its orthography and prosody together. One of the most popular of these conventions is the *TextGrid* file format, which is used by *Praat* (Boersma, 2001). This XML based file format stores any number of tiers that can be used to label prosodic features. Although very useful for visualization in Praat, this format is not designed to be functional for viewing and manipulating for itself. Every tier defines which event occurs at what time on its own and it is difficult to associate events that occur in parallel in different tiers. Also, for handling raw acoustic features Praat uses different file formats. Due to this design, a complete prosodic/acoustic representation of a short utterance ends up being represented with a clutter of files.

An optimal and standard data structure was needed in this study for two reasons: for accommodating creation and storage of prosodic data and also for easy processing with machine learning applications. *Proscript* framework was created to remedy for this deficit. It is both a data representation format and a specialized library for creation, manipulation, reading and writing of this sort of data. The name *Proscript* is a portmanteau of the words *prosody* and *transcript*. It is seen as an enhanced way of representing a speech transcript. Instead of tiers, speech is represented with its features that occur in parallel at discrete bounded intervals. These bounded intervals can be words or a group of words that is called "segments". A segment can represent, for example, a prosodic phrase, a sentence or a group of sentences. Any type of feature can be stored within these boundaries. Be it acoustic features such as intonation, intensity or morphosyntactic features as part of speech or speaker tags.

Proscript file format is based on the CSV file format. First line is the names of features that particular file stores and the following lines are the sequence of syntactic units together with the features that go parallel with them. See Table 3.1 for an example of parallel features stored in a Proscript file. In this particular example the discrete units are defined as words. The set of features is changeable. For example, a most simple set would be the words and their timing information without any syntactic or prosodic features. This set would be suitable for storing word-alignment information.

Feature	Details
word	as a token
id	unique word id
speaker id	unique speaker id
start time	start time of the word in an associated audio file
end time	end time of the word in an associated audio file
pause	coming before and after the word
punctuation	coming before and after the word
POS	part-of-speech
ToBI	ToBI label
mean f0	in Hertz and log-scaled (semitones)
mean intensity	in decibels and log-scaled
f0 contour	as a list in Hertz or semitones
intensity contour	as a list in Hertz or log-scaled
speech rate	relative to syllables TODO: in which units?

Table 3.1: Word-level information kept in an example Proscript format file.

A Proscript file can represent a short utterance as well as a whole dialog between two speakers. Dialog turns, for example, can be represented as segments with the speaker id tagged. Use is kept highly customizable through the library designed to handle Proscript files.

Proscript Python library was developed in order to make creation, manipulation and annotation of Proscript files as easy as possible. It can be imported from a Python script to batch process transcripted speech files, annotate them with the desired features and output as files. Both word alignment and prosodic acoustic tagging software (explained in following subsections) is accessible through the library.

Proscript as python package is accessible online[3]. Guide and example scripts are provided in the repository on development. A “Proscripter” script is provided to obtain Proscript file from a audio file and its transcription. [4]

[3] github.com/alpoktem/proscript

[4] Proscript file format is used as the accepted format in the other frameworks (Prosograph, punkProse, transProse) introduced in this dissertation.

Montreal Forced Aligner for Speech-to-text Alignment

Speech-to-text alignment is the process of determining boundary points of words and phonemes present in speech audio recordings defined by their text transcriptions. It proves to be essential for the work in this dissertation as it helps to align prosodic features in speech within their morphologic boundaries in their transcription.

For this task, the open-source *Montreal Forced Aligner* (MFA) (McAuliffe et al., 2017) is employed. Forced alignment process is built on an automatic speech recognition system and requires its own acoustic models and a pronunciation dictionary. Although pre-trained models for both English and Spanish is provided through the website of the tool[5], Spanish pronunciation dictionary is not openly available. For this reason, a Spanish pronunciation dictionary has been created that uses the same phoneme set as MFA[6]. Vocabulary has been gathered from the open source spell checker tool *ISpell*[7]. Phonetic transcriptions of each word in this dictionary was obtained by using *TransDic* software (Garrido et al., 2018).

3.1.1. *ProsodyTagger* for prosodic feature annotation

In order to augment speech data with prosodic/acoustic features *ProsodyTagger* is used. *ProsodyTagger* (Domínguez et al., 2016) is a part of the *Praat on the Web* service[8] (Domínguez et al., 2016) and was provided by its main author Dr. Mónica Domínguez for carrying out prosodic feature annotation task within the Proscript library. The tool is based on *Praat* and simplifies the process of extracting mean f0 and intensity features in speech given its segmentations as a TextGrid file. See Figure 3.1 for an illustration of the prosodic features extracted for a speech utterance using this tool.

3.2. Compiling the *Prosodically Annotated TED Talks Corpus*

In this section, I will introduce the TED talks corpus that was recompiled and published to serve for the automatic punctuation restoration work (explained in

[5] montreal-forced-aligner.readthedocs.io/

[6] Resource available in: https://github.com/TalnUPF/phonetic_lexica

[7] https://www.gnu.org/software/ispell/

[8] kristina.taln.upf.edu/praatupf

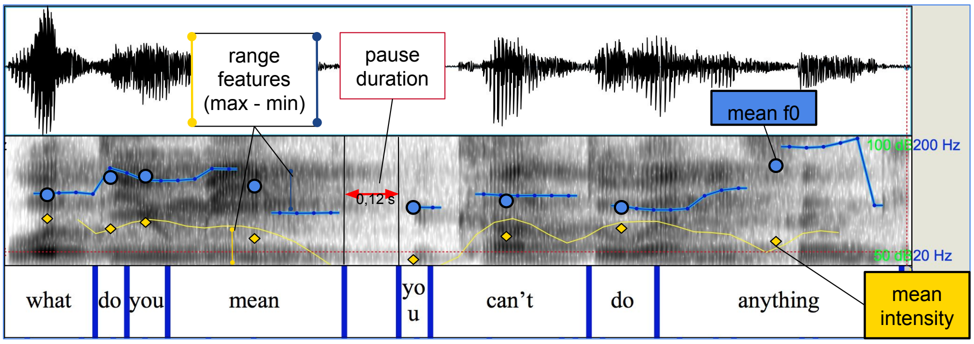


Figure 3.1: Word-level prosodic feature labelling.

Chapter 4). TED (Technology, Entertainment, Design) talks are a set of conference talks lasting in average 15 minutes each that have been held worldwide in more than 100 languages. They include a large variety of topics, from technology and design to science, culture and academia. The corresponding transcripts, as well as audio and video files, are available openly on TED's website[9]. For its public availability, TED talks have been the source of many corpora for linguistic analysis and machine learning based applications. Different formats of corpora based on TED talks cover areas from automatic speech recognition (Hernandez et al., 2018) to document classification (Hermann and Blunsom, 2014) and machine translation (Cettolo et al., 2012).

Farrús et al. (2016) studies paragraph-based prosodic cues in TED talks for the aim of improving naturalness in synthesizing spoken discourse. The dataset used for this work consists of 1365 talks published before 2014. Using the punctuation and paragraph annotated transcriptions available on the website, several prosodic analyses has been performed and stored at various lengths: words, sentences, segments (from subtitles) and paragraphs. Word and sentence timings were extracted using forced alignment. Pause durations between words were extracted from the provided word timings. Acoustic annotations are done at each interval automatically using the Praat software (**?**). Fundamental frequency (F0) and intensity contours were extracted at 10 ms precision and then converted to semitones relative to speaker mean value. Speaker mean values were represented by zero values in both cases.

Although available on demand, this extensive corpus is not published in an open way. Moreover, it was found out that words, word timings, punctuation information and acoustic features associated with words were scattered among

[9] www.ted.com

many files in the corpus. This made it difficult to process and create training data for machine learning based experiments. For these reasons, the corpus was re-processed, taking the information as it is, but making it easily readable.

Due to some talks lacking acoustic annotations, the recompiled corpus consists of a subset of 1038 talks in the original corpus. These talks were given by 877 English speakers, which means that some speakers were present in various talks. Through counting of sentence-ending punctuation marks, 155174 sentences were calculated to be present in this version. The dataset is published online as *Prosodically annotated TED talks* and is accessible through repositori.upf.edu/handle/10230/33981 with Attribution 4.0 International (CC BY 4.0) license[10]. Source code used during the recompilation of this corpora is also provided online[11].

3.3. Automatic Extraction of Parallel Speech Corpora from Dubbed Movies

Dubbing is the process of voice acting on top of the dialogues in a movie, TV series or documentary to make it accessible to viewers of another language. Popularity of dubbing of media material for a language depends greatly on the language culture of the country where the language is mainly spoken. For countries that prefer watching films on their mother-tongue, most movies and TV series go through this process before being released. Dubbing is carried out in professional studios and with professional voice actors.

There are certain characteristics of art of dubbing that makes it an interesting candidate as a resource for parallel corpora. The process as a whole can be considered as a translation process. However, it has much more points to be heeded than just merely translating the movie script. Actor and actresses lines have to be translated in a way that the lengths of the dialogues match. This is to ensure that voice-overs match the lip movements of the original actors. Once a translation that fits a line is found, voice actors record the segment over the original movie respecting the way of acting of the original actors. It can be seen as a way of re-enactment of the line but with another language. Eventually, the voice-over doesn't only carry the content to the dubbing language but also the paralinguistic aspects that go with it. A methodology has been built around getting advantage of this kinds of resources to obtain parallel speech corpora. In contrast with a methodology based on collecting bilingual samples in a controlled environment,

[10] https://creativecommons.org/licenses/by/4.0/

[11] https://github.com/alpoktem/ted_preprocess

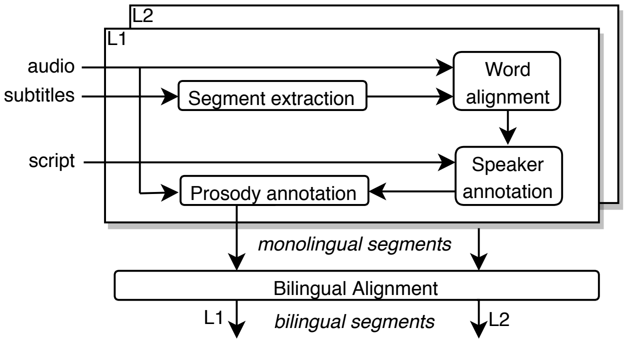


Figure 3.2: Overall corpus extraction pipeline. Audio excerpts are first processed in each language and then aligned to obtain bilingual segments.

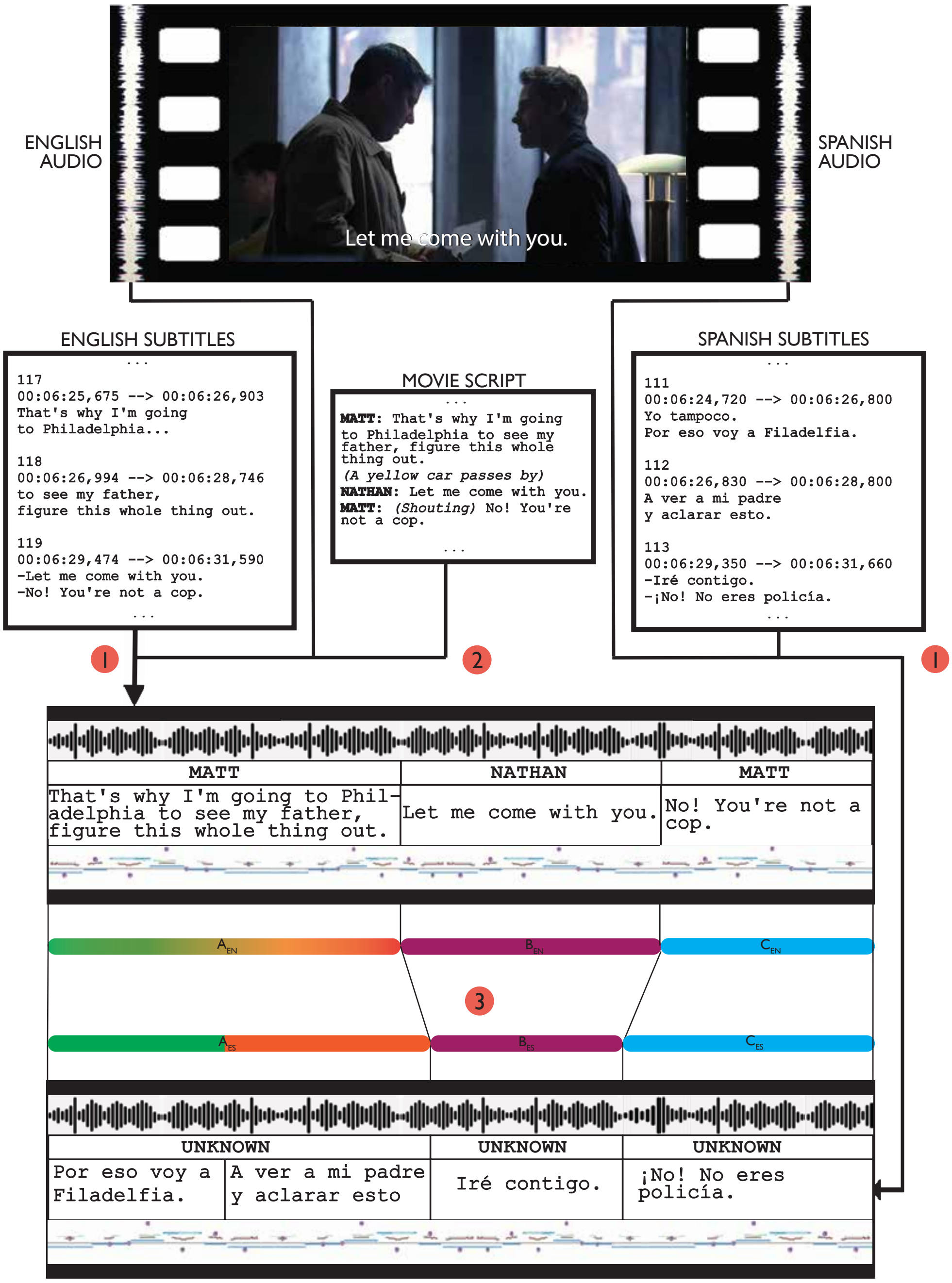
I propose to exploit dubbed movies where expressive speech is readily available in multiple languages and their corresponding aligned scripts are easily accessible through subtitles. The time information in subtitles makes it easy to align sentences of different languages since timing is correlated to the audio.

The proposed methodology needs only raw movie data, does not require any training (as is the case of Tsiartas et al. (2011)) and satisfies the following requirements: (1) it is easily expandable, (2) it supports any pair of languages where dubbed material is available, (3) it can handle any domain and speech style, and (4) it delivers a parallel spoken language corpus with annotated expressive speech. I will refer to "Expressive speech" as that speech that is prosodically rich, which is a feature needed in developing speech-to-speech translation applications where prosody is being taken into account.

The methodology for obtaining a parallel corpus from a dubbed media consists of three stages: (1) a monolingual step, where audio+text pairs are extracted from the movie in both languages using transcripts and cues in subtitles, (2) paralinguistic feature annotation (speaker information and prosody) and (3) alignment of monolingual material to extract the bilingual segments. See Figure 3.2 for an overview of the system pipeline. Figure 3.3 further illustrates the whole process on an example portion of a movie. Following subsections explain in detail each process.

Audio segment mining using subtitles

Subtitles are the source for obtaining both (1) audio transcriptions, and (2) timing information related to utterances in a movie. These information are contained



ENGLISH SUBTITLES

```
...
117
00:06:25,675 --> 00:06:26,903
That's why I'm going
to Philadelphia...

118
00:06:26,994 --> 00:06:28,746
to see my father,
figure this whole thing out.

119
00:06:29,474 --> 00:06:31,590
-Let me come with you.
-No! You're not a cop.
...
```

MOVIE SCRIPT

```
...
MATT: That's why I'm going
to Philadelphia to see my
father, figure this whole
thing out.
(A yellow car passes by)
NATHAN: Let me come with you.
MATT: (Shouting) No! You're
not a cop.
...
```

SPANISH SUBTITLES

```
...
111
00:06:24,720 --> 00:06:26,800
Yo tampoco.
Por eso voy a Filadelfia.

112
00:06:26,830 --> 00:06:28,800
A ver a mi padre
y aclarar esto.

113
00:06:29,350 --> 00:06:31,660
-Iré contigo.
-¡No! No eres policía.
...
```

Figure 3.3: Processes 1,2 and 3 of the methodology illustrated on a portion of a movie.

in a standard *srt*[12] subtitle file, entry by entry like the structure below:

```
1
00:06:25,675 --> 00:06:26,903
That’s why I’m going
to Philadelphia...
2
00:06:26,994 --> 00:06:28,746
to see my father,
figure this whole thing out.
3
00:06:29,474 --> 00:06:31,590
-Let me come with you.
-No! You’re not a cop.
```

Each subtitle entry is represented by an index, time cues and the script being spoken at that time in the movie. The script portion can consist of multiple (#2), complete TODO:next ,3 → #3? I mean, if sentence is ”That’s why ... thing out.”, 2 does not seem complete(#2,3) or incomplete sentences (#1) and from single (#1,2) or multiple speakers (#3). Using only these time cues does not suffice for extracting audio segments with complete sentences of a single speaker. To achieve this, word boundaries extracted with Montreal Forced Aligner is combined with punctuation mark positions to split and merge segments as needed. Two entries are merged if the first one does not end with a sentence-ending punctuation mark and the second one starts with a lowercase letter. Multi-speaker segments were split from the words following speech-dashes. This process is marked with the label ”1” on Figure 3.3.

The resulting segments from the subtitle excerpt from above would be:

1. That’s why I’m going to Philadelphia to see my father, figure this whole thing out.

2. Let me come with you.

3. No! You’re not a cop.

Speaker annotation through scripts

Movie scripts, which contain dialogue and scene information, are valuable pieces of information for determining the segment speaker labels. Scripts follows approximately the same format: Actor/actress name is followed by the line they

[12] SubRip text file format https://www.matroska.org/technical/specs/subtitles/srt.html

say. In between, there might be non-spoken information in brackets. An example excerpt from a movie script of TV series *Heroes* is given below:

```
MATT: That's why I'm going to Philadelphia to see my father,
      figure this whole thing out.
      (A yellow car passes by)
NATHAN: Let me come with you.
MATT: (Shouting) No! You're not a cop.
```

Unlike subtitles, scripts do not have timing information. In order to map subtitle segments with the speaker information, an automatic procedure is followed. First, all non-spoken text included in brackets is removed. Then, speaker tags and corresponding lines are extracted with regular expressions depending on the format of the script. Next, segments coming from subtitles are mapped one by one to lines in the script. If 70% of the words in a subtitle segment is included in a script turn, then the segment is labeled with the speaker of that turn. If it doesn't, up to five next script turns are checked as candidates.

It should be noted that scripts are usually only available in the original language. However, since segments are aligned on a later step with their dubbed matches, they can share the speaker labels. In Figure 3.3, speaker labels are extracted from the English script and matched with the subtitles. Spanish segments are left with an ”UNKNOWN” label until they are aligned with their respective English counterparts.

Word-level acoustic feature annotation

Each word in the extracted segments is automatically annotated with the following acoustic features: mean fundamental frequency (f0), mean intensity, speech rate and silence intervals (pauses) before and after. The first two features are extracted with *ProsodyTagger*. Pause information is calculated from word-boundary information and speech rate is calculated using:

$$word\,speech\,rate = \frac{\#syllables\,in\,word}{word\,duration} \tag{3.1}$$

To represent speaker independent, perceptual acoustic variations in the segments, both f0 and intensity values are converted into logarithmic semitone scale relative to the speaker norm value. Thus, speaker mean values were represented by zero values in both cases. Semitone values are calculated with the corresponding formula:

$$semitone(x, norm) = 12 * \log(\frac{x}{norm}) \tag{3.2}$$

The prosodic annotations are shown under the extracted segments with Prosograph feature visualizations in Figure 3.3.

Cross-lingual segment alignment based on subtitle cues

The first three methodologies presented in this section dealt with extraction of segments in each language. Audio and subtitle pair of each language result in a number of speech segments. This subsection explains how these segments are aligned to create the bilingual segment pairs.

As explained earlier, the dialogues in the original and dubbed language correspond to each other time-wise. So, in order to align segments extracted for each language, timing information of the segments can be exploited. However, as subtitles show slight differences, alignment cannot be performed one-to-one. Number of segments extracted from each language can differ. This means that the segment alignments can be one-to-one, one-to-many, many-to-one or many-to-many depending on the sentencing structure in the subtitles.

In order to create an alignment algorithm based on time cues, a metric is defined that measures the correlation percentage between two sets of ordered segments S=$\langle s_1, ..., s_N \rangle$ and E=$\langle e_1, ..., e_N \rangle$:

$$segments\, correlation = max(0, \frac{correlating}{span} \times 100) \tag{3.3}$$

$$correlating = min(e_N^s, s_N^e) - max(s_1^s, s_1^s) \tag{3.4}$$

$$span = max(e_N^e, s_N^e) - min(e_1^s, s_1^s) \tag{3.5}$$

where e_x^s and e_x^e denote the starting and ending time of the x^{th} segment in set E, s_x^s and s_x^e denote the starting and ending time of the x^{th} segment in set S.

The alignment procedure is as follows: two indexes i_E, i_S are kept which slide through the segments of each language. First, segments corresponding to each index are checked if they correlate more than the T_{Sure} threshold. If they do, they are assigned as a one-to-one matched pair. If not, the possibilities of one-to-many, many-to-one or many-to-many matches are considered. This is done through computing the correlations between combinations of the current and two following segments and selecting the most correlating segment set pair. While considering combinations of the segments it is made sure that two merged segments belong to the same speaker and are not more than 10 seconds far from each other. If the combined segment set pair with highest correlation has a correlation of more than T_{Merged} threshold, then the combinations are merged into one segment and paired with each other.

Although the T_{Sure} threshold catches most of the one-to-one mapping segments, many of them still fall below this threshold even if they map. So, another decision step is added where if one-to-one mapping correlation scores higher than merged pairings and it scores above a T_{OK} threshold, then it is preferred as a matched pair.

Figure 3.3 illustrates two examples of segment matching. First two Spanish segments are merged to align with the first English segment. Following segments are aligned one-to-one as their durations correlate enough.

TODO: Explain a little bit more the figure. For instance, add ”the speakers are labelled in the source language and paired in the target language”

Using the parallel corpus extraction framework

This methodology is developed as a open source framework called *movie2parallelDB* and is accessible online [13]. The usage instructions are included in the online repository.

The scripts are run with audio and their corresponding subtitles. Therefore, audio tracks needs to be extracted from the respective video prior to the process. Matching subtitles also needs to be acquired. One challenge that this method poses is that although it is easy to find subtitles in both original and dubbed languages of a movie, dubbing script might differ from subtitles. This is due to the difference in process between subtitling and dubbing. As it is mandatory to obtain exact transcription of the audio segments, subtitles need to be corrected prior to the process.

3.4. Compiling the *Heroes Corpus*

The methodology presented in the previous section was put into practice by compiling a corpus from 2000’s popular science fiction TV series *Heroes*[14]. Originating from United States, Heroes ran in TV channels worldwide between the years 2006 and 2010. The whole series consists of 4 seasons and 77 episodes and is dubbed into many languages including Spanish, Portuguese, French and Catalan. Each episode runs for a length of 42 minutes in average.

[13] `github.com/alpoktem/movie2parallelDB`

[14] Produced by Tailwind Productions, NBC Universal Television Studio (2006-2007) and Universal Media Studios (2007-2010)

Raw data acquisition

The DVD's of the series were obtained from the Pompeu Fabra University Library. Episodes were extracted using the *Handbrake* software and were saved as *Matroska format (mkv)* files. Mkv files can hold multiple channels of audios and subtitles embedded in it like DVDs. In order to run *movie2parallelDB* scripts, audio and subtitle pairs for both languages needed to be extracted. Audio is extracted using the *mkvextract* command line tool[15]. As subtitles were embedded as bitmap images in the DVD, an optical character recognition (OCR) software[16] was used to convert them to *srt* format subtitles. As OCR is an error-prone process, the resulting srt files needed to be spell checked.

In total, 21 episodes were processed to obtain 25 hours English and Spanish audio with their corresponding subtitles. The episode scripts were obtained from a fan web page[17].

Manual subtitle correction work

The Spanish subtitles needed slight correction in order to match the Spanish audio. The reason for this mismatch was explained in the previous subsection. As dubbing and subtitling are two distinct processes, there often are differences. It was observed that the Spanish subtitle transcripts were matching the Spanish audio in approximately 80% of the cases. As exact correspondence between audio and transcription was aimed, a correction process was carried out. Both subtitle transcripts and time-stamps had to be corrected to match exactly what is being spoken on the dubbing audio and when. This process was done using a subtitle editing program *Aegisub*[18].

An advantage the manual correction process gives is the opportunity to filter out unwanted audio portions that would otherwise end up in the corpus. This process is necessary especially in the case the source material is noisy. During the correction process, subtitle segments that contained noise and music, overlapping or unintelligible speech and speech in other languages (e.g. Japanese) were removed. The spell checking and timestamps and script correction of 21 episodes was done by two annotators and took 60 hours in total.

For each episode to be processed, the annotators were provided with the episode video, English and Spanish subtitles extracted with the OCR software. The cor-

[15] https://mkvtoolnix.download/

[16] Through a functionality provided by Subler: https://subler.org/

[17] https://heroes-transcripts.blogspot.com/

[18] http://www.aegisub.org/

Subtitle Entry	**Audio Transcript**
-Te presento a tu compañero. -¿Me vas a cambiar?	-Te presento a tu compañero. -¿Me cambiarás?
Me han tenido dos años, pensarán que los abandoné.	Me han tenido encerrado dos años, pensaran que los abandoné.
Discutimos,...	Empezamos a discutir...
-Nunca quise... -Cierra la boca! Escucha.	-Yo nunca quise... -Cierra la boca! Escucha.
Hablaremos cuando vuelva, ¿vale?	Hablaremos mas cuando vuelva, ¿vale?

Table 3.2: A selection of non-matching subtitle entries and dubbing scripts in Heroes series episodes.

rection procedures for each episode are as follows:

1. Automated correction of OCR errors in English subtitles.

2. Manual correction of English subtitles with a spell checker.

3. Automated correction of OCR errors in Spanish subtitles.

4. Manual correction of Spanish subtitles with a spell checker.

5. Proofing and correction of the Spanish subtitles.

The automated correction process involved a basic substitution procedure for the character errors that the OCR software did consistently. For example the letter 'ñ' would be mistaken almost always as 'fi' or 'I's would be mistaken as lowercase 'L's. For further non-standard errors, the spell checker provided in *Aegisub* software was employed. Each spelling mistake in the subtitles were replaced with its corrected version.

Last step involves checking of the transcripts and timings of each entry in the Spanish subtitles. Entries that do not correspond to the speech in the dubbed audio are corrected. Also, start and end time of the subtitle entries are adjusted so that it fits perfectly to the spoken segment. See Table 3.2 for a selection of entries that show difference in transcript between subtitle entries and dubbing transcript. Depending on the episode, about 10% to 20% of the subtitle transcript needed to be corrected.

Heroes corpus in numbers

Statistics of the first preparation sprint of *The Heroes Corpus* are presented in this section. 21 episodes from season 2 and season 3 were processed. Total audio durations of 7000 parallel segments is about 9.5 hours (see Table 3.3). Counts of several linguistic units in the final parallel corpus are presented in Table 3.4. A summary of how much of the content in one episode ended up in the dataset in average is presented in Table 3.5.

	English	Spanish
Total duration	4:45:36	4:43:20
Avg. duration/segment	0:00:02.44	0:00:02.42

Table 3.3: Corpus duration information.

	English	Spanish
# words	56320	48593
# tokens	72565	63014
# sentences	9892	9397
Avg. # words/sentence	5.69	5.17
Avg. # words/segment	8.04	6.94
Avg. # sentences/segment	1.41	1.34

Counts	English	Spanish
# words	56 320	48 593
# tokens	72 565	63 014
# sentences	9 892	9 397
Avg. # words/sentence	5.69	5.17
Avg. # words/segment	8.04	6.94
Avg. # sentences/segment	1.41	1.34

Table 3.4: Word, token, sentence counts and average word count for parallel English and Spanish segments.TODO:explain somewhere roughly what are tokens

	English	Spanish
Avg. # sentences (subtitles)	647	554
Avg. # sentences (extracted)	628	513
Avg. # segments	526	459
Avg. # parallel segments	334	

Table 3.5: Averages numbers for each episode.

3.4.1. Discussion

The first version of the Heroes corpus shows that the proposed methodology for bilingual corpus building is successful in terms of the quality of the segments extracted. Our manual inspections show that the segments are correctly aligned and the transcriptions are correctly stored with the audio segments in general.

The Spanish subtitle correction task was the only time-consuming part of the whole process. However, the task showed its usefulness for obtaining clean parallel segments. Subtitle segments that were removed during the correction process ensured the elimination of unwanted audio portions.

Table 3.5 shows the amount of information loss at various stages. The first one being the word-alignment process where in average 5% of the sentences are lost due to the failure of the word aligner in segmenting words. Many of the segments that are lost this way were either noisy or erroneous. The biggest loss happens at the stage of alignment where in average 30% of the segments in each language are left unaligned. This percentage is directly affected by the alignment parameters explained in Section 3.3. For example, selecting a lower T_{Sure} leads to detecting more aligned segments but also to more mismatches. A similar logic applies to T_{OK}. Also, choosing a lower T_{Merged} leads to more coverage of the sentences but again more ... rewrite this sentence more as combinations with others, leading to fewer and longer segments. After experimenting with a handful of parameter combinations, this parameter combination proved to be the most optimal for this task: $T_{Sure} = 70\%$, $T_{Merged} = 80\%$ and $T_{OK} = 30\%$.

TODO: In previous paragraph: first you use word aligner and the stage of alignment The second one, put adjective to the new alignment. E.g.: language-pair alignment ?

TODO: The sentence more as combination with others sounds not very english to me, and repeat more. Maybe is enough ”Also, choosing a lower T_{Merged} leads to more coverage of the sentences but leading to fewer and longer segments. ”

3.5. *Prosograph* for Aiding Study of Large Speech Corpora

Prosody conveys several communication elements such as meaning, intention, and emotions, among others. Being able to clearly visualize the different elements involved in prosody –intonation, rhythm, and stress– may be helpful for computational prosody research. Several speech analysis tools (e.g. Praat (Boersma, 2001)), together with derived scripts and tools (Xu, 2013; Mertens, 2004; Domínguez et al., 2016) partially cover these needs by helping to visualize quantifiable speech features like fundamental frequency ($F0$) and intensity contours, word stress marking, or prosodic labeling. These tools work well when showing detailed analyses on data and visualizing one single utterance at a time, but fail in visualizing generalized word-averaged speech features of many utterances, e.g., a discourse or a collection of speech samples, at once.

Prosograph was born from the need to study prosody of long segments of speech to see the relation of prosodic features with punctuation in text. Inspiration was taken from music scores and piano rolls that help reading and visualizing music. Similar to a musical analysis tool, Prosograph helps visualize acoustic and prosodic structure in speech together with its transcript. Also through an interactive interface it makes it easy to listen to any portion of the displayed speech to accommodate auditory analysis (Öktem et al., 2017b).

3.5.1. Implementation

Prosograph is written in Python mode of Processing[19] because of its simplified access to graphical and interactive interface. In order to simulate music scores, the speech prosodic features are plotted in the vertical axis over a temporal horizontal axis. Words are put in order together with pauses and punctuation, and the prosodic features are drawn under each corresponding word. An overview of the tool can be seen in Figure 3.4.

Two modes of Prosograph have been implemented: monolingual (standard) mode and bilingual mode. Bilingual mode makes it possible to view aligned parallel corpora. Aligned samples are displayed side by side to accommodate e.g. prosodic comparison. Figure 3.5 illustrates an overview of Prosograph in bilingual mode.

Prosograph reads prosodically annotated speech data from Proscript format files (see Section 3.1). Data path, and names and types of features in the files to be visualized has to be set in a configuration file before running the software.

Data types in Prosograph

Prosodic features differ in the way they encode words or sentences. For instance, word stress is a feature that represents salience among a group of words, intonation and intensity are continuous encodings throughout successive voiced phonemes, accent is a peak that occurs at a certain point in a word, etc. Because

[19] py.processing.org/

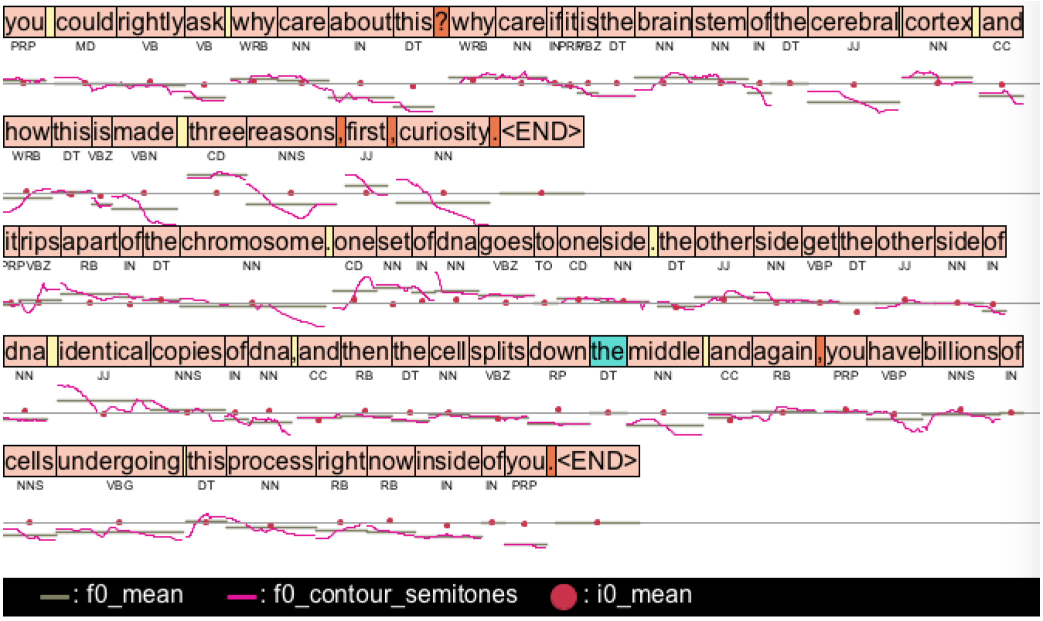

you could rightly ask why care about this ? why care if it is the brainstem of the cerebral cortex and
how this is made three reasons , first , curiosity . <END>
it rips apart of the chromosome . one set of dna goes to one side . the other side get the other side of
dna identical copies of dna , and then the cell splits down the middle and again , you have billions of
cells undergoing this process right now inside of you . <END>
: f0_mean
: f0_contour_semitones
: i0_mean

Figure 3.4: An example of a visualization frame of segments from a conference talk with Prosograph.

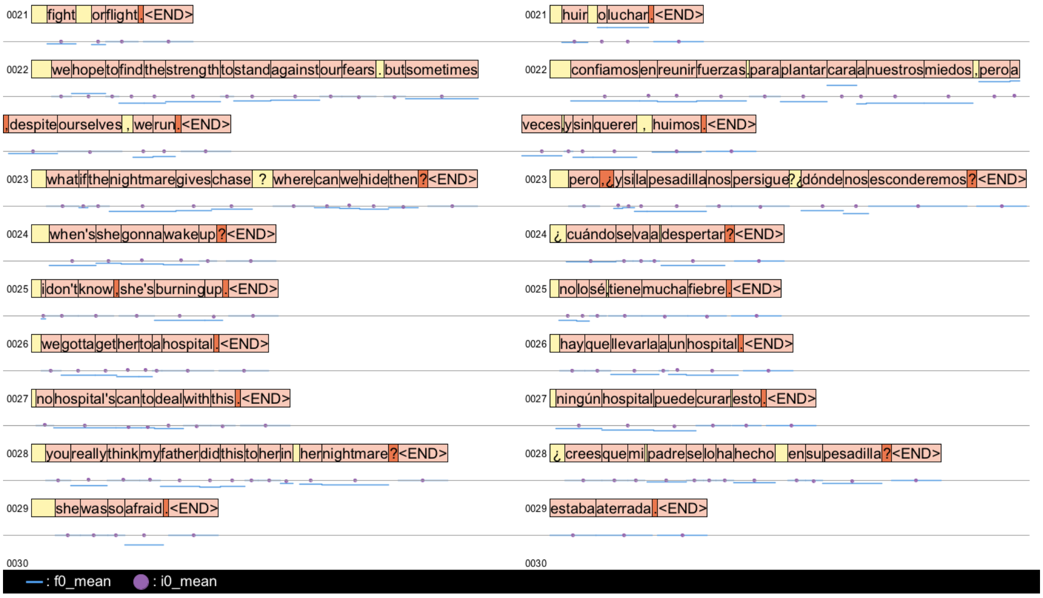

0021 fight or flight . <END>
0021 huir o luchar . <END>
0022 we hope to find the strength to stand against our fears . but sometimes
0022 confiamos en reunir fuerzas , para plantar cara a nuestros miedos , pero a
, despite ourselves , we run . <END>
veces , y sin querer , huimos . <END>
0023 what if the nightmare gives chase ? where can we hide then ? <END>
0023 pero , ¿ y si la pesadilla nos persigue ? ¿ dónde nos esconderemos ? <END>
0024 when's she gonna wake up ? <END>
0024 ¿ cuándo se va a despertar ? <END>
0025 i don't know , she's burning up . <END>
0025 no lo sé , tiene mucha fiebre . <END>
0026 we gotta get her to a hospital . <END>
0026 hay que llevarla a un hospital . <END>
0027 no hospital's can to deal with this . <END>
0027 ningún hospital puede curar esto . <END>
0028 you really think my father did this to her in her nightmare ? <END>
0028 ¿ crees que mi padre se lo ha hecho en su pesadilla ? <END>
0029 she was so afraid . <END>
0029 estaba aterrada . <END>
0030
0030
: f0_mean
: i0_mean

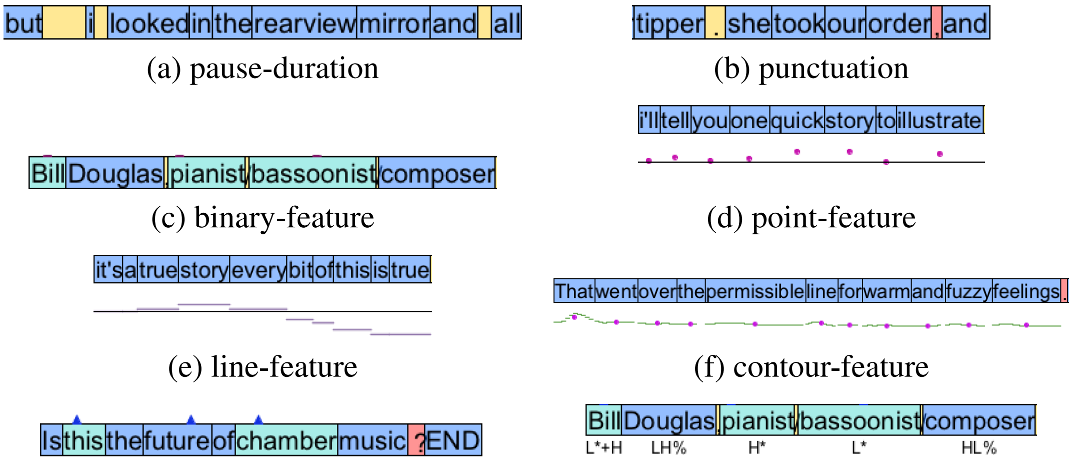
Figure 3.5: Visualizing parallel samples from an episode of Heroes Corpus with bilingual mode of Prosograph.

of these variations each prosodic feature demands a special way for its storage and visualization. In order to visualize different types of prosodic encodings, following data sequence types that are encoded word-wise were compiled:

TODO: In previous par., check writting of: ”that are encoded word-wise were compile”

TODO: In the next list, you say ”can be used”, ”could be used”; this suggest that the tool is customizabled by user, but no explained .. In fact it is explained later, but you should say something here, because I was expecting just the description of the tools

- **pause-duration** holds the silence duration coming after the corresponding word. Paused intervals are visualized as an empty yellow box between words with a width proportional to the length of the pause (see Figure 3.6a).
- **punctuation** holds the punctuation mark coming before or after the corresponding word. Punctuation marks are placed in the same axis with words. If a punctuation mark coincides with a pause, then it is placed inside the pause interval (see Figure 3.6b).
- **binary-feature** holds a binary value determining if the corresponding word carries a certain feature (1) or not (0). This feature type can be used e.g. for word-stress. Bounding boxes of these words are drawn with a salient color (see Figure 3.6c).
- **point-feature** holds a real numbered value that belongs to the corresponding word (e.g. standard f0 deviation, mean f0, median f0, etc.). It is placed at its value below the middle of the word’s bounding box (see Figure 3.6d).
- **line-feature** holds a real numbered value as point-features. They are visualized as a line below and parallel to the word. This feature type could be useful e.g. for visualizing better the mean f0 movement across the utterance (see Figure 3.6e).
- **contour-feature** holds sequences of same length corresponding to each word. Each value is treated as curve bins and drawn as a line below the word in same length intervals. It is to be used e.g. for visualizing f0 curves or intensity curves or quantiles (see Figure 3.6f).
- **percentage-feature** holds sequences of varying lengths where each value in the sequence corresponds to a percentage of time with respect to the duration of the word. A mark is placed at the corresponding time position below the word’s bounding box. This feature type can be used e.g. to mark


but i looked in the rear view mirror and all
tipper . she took our order , and
Bill Douglas , pianist , bassoonist , composer
i'll tell you one quick story to illustrate
it's a true story every bit of this is true
That went over the permissible line for warm and fuzzy feelings .
Is this the future of chamber music ? END
Bill Douglas , pianist , bassoonist , composer
L*+H LH% H* L* HL%

(a) pause-duration (b) punctuation

(c) binary-feature (d) point-feature

(e) line-feature (f) contour-feature

(g) percentage-feature (h) label-feature

Figure 3.6: Word-aligned feature data types in Prosograph.

the point where the accent occurs in a word, f0 or intensity peaks (see Figure 3.6g).

- **label-feature** holds a string label for their respective words. The label is written just below the respective word's bounding box. This feature type can be used to visualize prosodic labels such as ToBI or part of speech (see Figure 3.6h).

3.5.2. Access and Usage

Prosograph is made publicly available as an open-source software in the online repository github.com/alpoktem/Prosograph under the GNU General Public License[20].

Once it is installed and the configurations are set, utterances in the dataset are shown in batches and user can navigate over the batches using keyboard shortcuts N(next) and B(previous). The current batch frame can be saved as an image by pressing S.

Colors of different prosodic features are set randomly at run-time. A legend showing which color belongs to which feature is shown at the bottom of the screen. If not easily distinguishable, the colors can be changed (again randomly) by pressing C on keyboard.

[20] http://www.gnu.org/licenses/

To listen to a particular sample, beginning and end word of the utterance need to be selected. When P key is pressed, the utterance is played between the selected word interval. This is made possible if an audio file accompanies the Proscript file in the same directory with the same name.

3.5.3. Conclusion

I have presented Prosograph, which can be used for the analysis of prosodic features and patterns in a speech corpus. It has been designed to be robust for handling different types of prosodic data annotated on word level. By simplifying the process of observation and comparison of prosody, this application can be used in many areas of research such as language learning and acquisition, comparative studies in different languages, tone languages, audiovisual prosody, etc.

Prosograph was first implemented to aid feature selection process in the punctuation restoration methodology that this dissertation presents in Chapter 4. After its development, it was used to demonstrate the results of this system and to reason how a neural punctuation restoration system behaves with respect to various prosodic features. Through the development of the bilingual mode, it proved its use in studying prosodic transformations in the dubbed translations in the Heroes corpus and helped inspire the prosodically enhanced translation system that is to be presented in Chapter 5. Also, with its easy integration with Proscript library, it simplifies creation of visualizations of speech samples for linguistic study. Visualizations of speech samples in this dissertation are made with Prosograph.

It should be noted that Prosograph is not a program that obligates its usage "as-it-is". It is a framework, written in a highly visual and experimental programming language to be customized to the needs of its user. For instance, a linguist wanting to observe certain characteristics in a recorded corpus can set it up to her like and display segments together with speaker information. The bilingual mode can be further extended to display utterances of two speakers together with a reference utterance and facilitate comparison. Another use-case could be real-time prosodic display and assessment for accent and intonation training. I hope the framework inspires researchers from different fields to study prosody and speech corpora in a visual and customizable way.

Chapter 4

PUNCTUATION RESTORATION USING PROSODIC CUES

This chapter deals with the theme of punctuation restoration in speech transcripts focusing on its relation with prosody. My first aim is to convince the reader of the crucial role of prosody when determining placement of punctuation marks in raw speech transcripts (Section 4.1). Next, the claims are further elaborated with some quantitative analyses on punctuation usage and prosody-punctuation relation in a corpus of conference talks (Section 4.2). Then, I will present a deep learning based framework (Öktem et al., 2017a) for carrying out experiments on testing effects of various morphosyntactic and prosodic features on the task of punctuation restoration (Section 4.3). Experiments explained in Section 4.4 focus on testing which feature set works best for the problem (4.4.3), quantifying the influence of prosodic feature usage into dependency parsing quality (4.4.4) and finally evaluating the system incorporated on a real speech recognition application (4.4.5). Final remarks and conclusions are given in Section 4.5.

4.1. Motivation and Background

The introduction of punctuation marks into the output of automatic speech recognition (ASR) is an important issue in applications such as automatic transcription/subtitling, speech-to-speech translation, language analysis, etc. Punctuation is essential for grammaticality, understandability, and –in the case of a number of different tasks–, subsequent processing. Thus, correct sentence segmentation and punctuation of recognized speech improves the quality of machine translation (Matusov et al., 2006; Peitz et al., 2011; Cho et al., 2017; Lu and Ng, 2010), and missing periods and commas in machine generated text results in sub-optimal information extraction from speech (Favre et al., 2008; Hillard et al., 2006). On a more end-user perspective, Tündik et al. (2018) show that punctuated captions are preferred by viewers of television shows in both manually and automatically generated transcriptions. Also, most of the data-driven parsing models require segmentation of recognized text into sentence like units and use punctuation as features (Jones, 1994; Spitkovsky et al., 2011; Ma et al., 2014).

Punctuation marks support understandability and readability in written language. Sentences generally form an enclosed unit with subject, object and verb and are marked by sentence-ending punctuation marks such as period, question mark and exclamation mark according to their modality (statement, interrogative etc.) Intra-sentence punctuation marks such as comma are required by certain syntactic phenomena like enumeration, clause separation, dislocation etc. In some languages (such as, e.g. English), punctuation is also essential for the realization of the information structure (Moore, 2016).

In spoken language, punctuation of the transcribed speech is influenced by two intertwined phenomena: (1) syntax and (2) prosody. Syntax determines the distribution of punctuation marks in accordance with the orthography of a language. Prosody realization in speech (such as, e.g., word grouping, pausing, emphasis, rising-falling intonation, etc.) tends also to signal the position and type of the punctuation marks. As a matter of fact, it has been debated in history whether prosody is influenced by punctuation or vice versa (Chafe, 1988). Early works on English grammar regard the use of punctuation as a mere symbology of how the language sounds. According to Lowth (1762) *point marks* (period, colon, semicolon and comma) indicate breaks with different lengths, question mark and exclamation marks indicate "an elevation of the voice" and parentheses indicating a "moderate depression of the voice". Modern linguistic definitions, on the other hand, state that punctuation is directly dictated by grammatical rules with prosody influencing it from time to time (Algeo, 1987). Regardless of a formal standpoint, it can be seen that prosody is related many times with punctuation. For instance, a pause after consecutive words might signal an enumeration, which requires comma, and rising intonation at the end of a sentence is a likely indicator of a question. Sentence and discourse boundaries are often marked with pauses and a reset in pitch.

During the manual transcription of an audio recording, both modalities, syntax and prosody, are used in determining the phrasing structure and punctuation. Example below illustrates the effect of prosody on punctuation, where the raw text could be punctuated in two different ways, eventually leading to two different meanings and syntactic structures.

Raw *and with all sincerity I can say I am glad I lived those two years of my life* *that way*

(1a) *And with all sincerity, I can say, I am glad I lived those two years of my life that way.*

(1b) *And with all sincerity, I can say I am glad, I lived those two years of my life that way.*

Although it is ambiguous where to place the commas in this example just by looking at the transcription, by listening to the voice sample[1], there is only one possible punctuation: (1a), where *I lived those two years of my life that way* is a subordinated clause of *I am glad*.

Many state-of-the-art approaches to automatic punctuation restoration are driven by textual criteria only (Cho et al., 2017; Lu and Ng, 2010; Ueffing et al., 2013; Gravano et al., 2009; Jakubıcek and Horák, 2010; Che et al., 2016). However, it is proved that combination of prosodic and acoustic features with a textual model improves accuracy in ASR output (Baron et al., 2002; Khomitsevich et al., 2015; Tilk and Alumäe, 2015, 2016). Some approaches that use textual and prosodic models (or a combination of them) consider punctuation placement with narrow-range features such as n-grams (Liu et al., 2006; Khomitsevich et al., 2015; Kolář et al., 2004). Recent data-driven approaches that use recurrent neural networks (RNN)[2] proved to be competitive for the task due to RNN's ability to capture long and short term syntactic dependencies. These models, moreover, get use of word vectors which proves to capture well both syntactic and semantic structure of the language (Treviso et al., 2017; Che et al., 2016). However, such neural models that account for prosodic features (e.g. Tilk and Alumäe (2015, 2016)) rely merely on pause duration between words, while other prosodic features such as pitch and intensity information are ignored. Another shortcoming of these approaches is that the models are trained either only on written data (Ballesteros and Wanner, 2016; Che et al., 2016) or on a combination of written and spoken data (with, again, a dominance of written material) (Tilk and Alumäe, 2016). This makes the trained models biased towards written data.

My motivation extends from the necessity seen in inclusion of prosody in a more complete way into the problem of punctuation restoration on raw speech transcripts. In applications where automatic speech recognition is employed, it is possible to integrate a prosodic feature extraction framework which would contribute to the accuracy of the punctuation placement. Also, there is no earlier

[1] Accessible from `github.com/alpoktem/punkProse/tree/master/audio-samples`

[2] Explained in Section 2.1.1

study mentioning the individual and combined effect of various prosodic features (e.g. intonation, intensity, speech rate) to the generation of various punctuation marks in a neural-network based setting. For these reasons, this chapter gives focus to the development of a framework that enables testing of various prosodic features in the problem of punctuation restoration. Furthermore, the applicability of the introduced model is put into test on two distinct settings. Firstly, the effect of prosodic punctuation restoration is studied on quality of dependency parsing which is a method often used in natural language processing (NLP) applications. Secondly, the methodology is put into test with a real ASR system.

4.2. Analyzing Punctuation in Conference Talk Transcripts

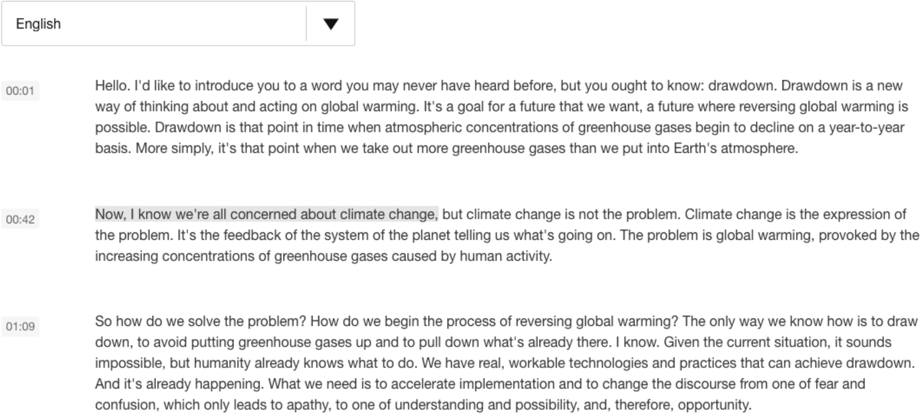
In this section, I will study the *TED Talks Corpus* presented in Section 3.2 in terms of punctuation usage and correlation of punctuation marks with pausing in speech. This kind of a quantitative analysis is performed both for helping design an automated methodology to solve the problem of punctuation restoration and also to help interpret its results.

The speech style involved in conferences is usually defined as semi-spontaneous. This is for the fact that it is delivered without being read from a source text, however, with prior rehearsal possibly utilizing a written form of the talk. Although it restricts to a certain spoken language stype, it still gives a good estimation for extracting knowledge on punctuation placement for spoken language transcription.

Both punctuation and the transcriptions analyzed are manually annotated by volunteers who watch and transcribe the talks. Punctuation marks and paragraph breaks are placed while listening to the talks at the same time of transcribing them meaning that they are related with the prosodic structure of the talk. See Figure 4.1 for an example of the transcription structure available for the talks on TED's website.

The first analysis that I perform involves examination of the frequency of each punctuation mark in the dataset. As demonstrated in Figure 4.2, it is observed that the majority of the punctuation marks in the dataset consists of a comma and a period, corresponding to 94% of all punctuation marks. As most of the talks go in the style of a monologue, questions are seldom made explaining the 3.7% share of question marks.

Inter-word pauses in speech are known to be a pertinent prosodic feature in determining sentence and phrase boundaries, and punctuation marks (Kolář et al.,



00:01 Hello. I'd like to introduce you to a word you may never have heard before, but you ought to know: drawdown. Drawdown is a new way of thinking about and acting on global warming. It's a goal for a future that we want, a future where reversing global warming is possible. Drawdown is that point in time when atmospheric concentrations of greenhouse gases begin to decline on a year-to-year basis. More simply, it's that point when we take out more greenhouse gases than we put into Earth's atmosphere.

00:42 Now, I know we're all concerned about climate change, but climate change is not the problem. Climate change is the expression of the problem. It's the feedback of the system of the planet telling us what's going on. The problem is global warming, provoked by the increasing concentrations of greenhouse gases caused by human activity.

01:09 So how do we solve the problem? How do we begin the process of reversing global warming? The only way we know how is to draw down, to avoid putting greenhouse gases up and to pull down what's already there. I know. Given the current situation, it sounds impossible, but humanity already knows what to do. We have real, workable technologies and practices that can achieve drawdown. And it's already happening. What we need is to accelerate implementation and to change the discourse from one of fear and confusion, which only leads to apathy, to one of understanding and possibility, and, therefore, opportunity.

Figure 4.1: Transcription available in TED web page for the talk ”100 Solutions to Climate Change” by Chad Frischmann.

2004; Christensen et al., 2001). The relation between inter-word pauses and punctuation is analyzed in two ways: First checked is the presence of a pause given that there is a particular punctuation mark, and second checked is the type of the punctuation mark given that there is a pause. Note that the pauses are defined as intervals in speech where no speech signal is detected. This information is obtained from the word alignments available in the corpus.

Figure 4.3 shows results of the first analysis. It is seen that sentence-ending punctuation marks are more likely to be accompanied by a pause. Most paused interval is where periods occur (51.6%). This means that at most half of the sentence boundaries are actually marked by pause. Commas seem to be marked with a pause in only 27.5% of the cases.

Second analysis illustrated in Figure 4.4 shows the pause-punctuation causality in the opposite direction. Paused intervals are analyzed in terms of the type of punctuation event occurring at that interval. Performing a binary analysis, it shows that more paused intervals (52.6%) are punctuated than unpunctuated (47.4%). However, the small difference indicates that it is only slightly more probable that a paused interval infers a punctuation than no punctuation. Moreover, it is seen that the distribution of the punctuation marks is reflected in the order of the frequency of the events. However, although there are more commas in the corpus than periods, the latter type shows to be much more likely to occur in a paused interval.

With respect to durations of the pauses, right side of Figure 4.4 shows the average of non-zero pause lengths that correspond to each punctuation event. Sentence-ending punctuation marks tend to correspond to longer breaks than com-

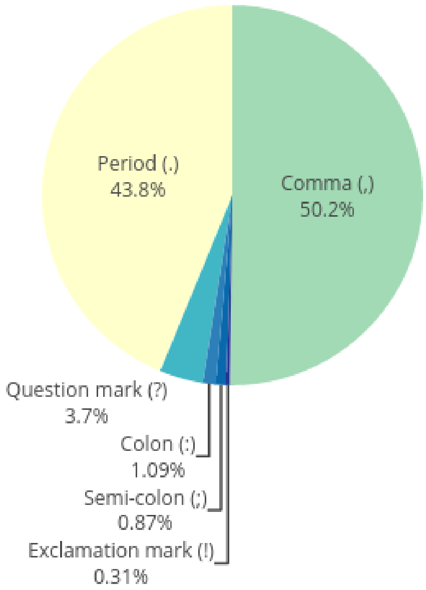


Figure 4.2: Punctuation distribution in the transcripts of the TED Talks Corpus.

mas. Excluding very rarely occurring punctuation marks (colon, semi-colon and exclamation mark), sentence boundary pauses are almost half a second in average. It is also seen that the 47.4% of the pauses, the ones without any punctuation mark, are actually of very short duration in average (40 ms).

The quantitative analyses show that presence of pauses is not a discriminating feature by itself in determining the presence of a punctuation mark. However, length of the pause can be a good feature for both determining presence of a punctuation and also the type of the punctuation. Sentence-endings are more likely to be related with a break and if so, it indicates a longer break compared to commas. However, this distinction fails to show itself between different types of sentence-ending marks, period and question mark. This signals the necessity of other discriminative features, be it syntactic or prosodic, for the differentiation between them.

Another finding is that a data-driven model based on this dataset would be useful in classifying only a group of punctuation marks consisting of period, comma and question mark as the rest is not represented enough in this particular dataset.

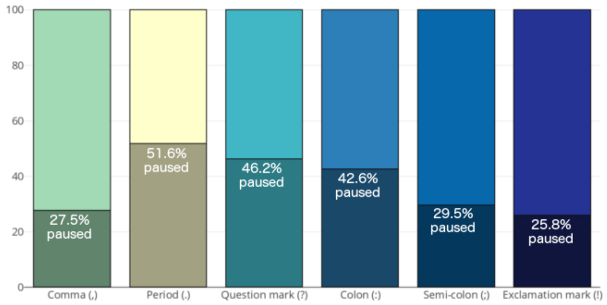


Figure 4.3: Pausing percentage of each punctuation mark in TED Talks Corpus.

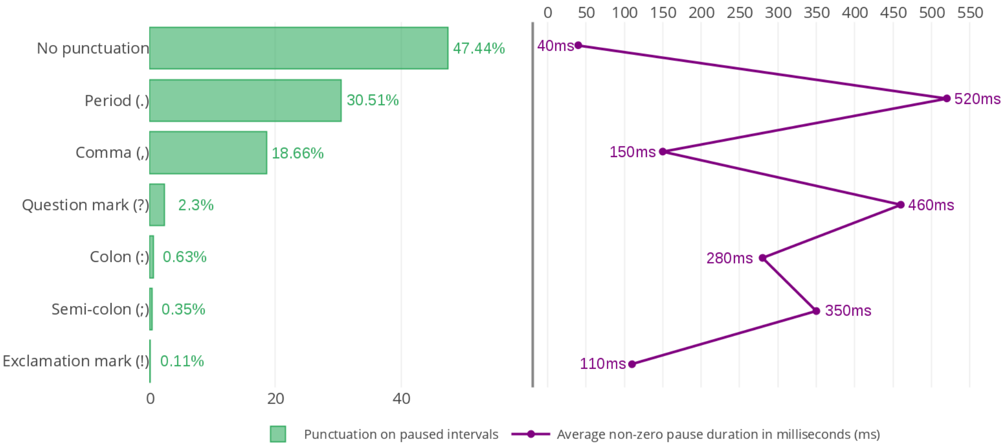
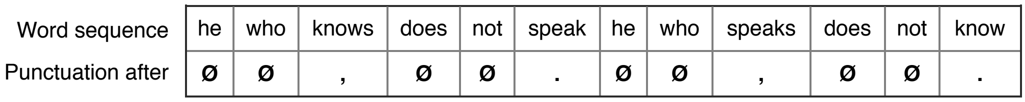


Figure 4.4: Distribution of punctuation presence in paused intervals (left) and corresponding average non-zero pause lengths of each punctuation mark (right).

4.3. Methodology

As explained earlier in Sections 2.1.3 and 2.2.1, state-of-the-art on punctuation restoration shows advance with two main approaches (1) combination of prosodic and lexical features, and (2) employment of RNN-based architectures. A RNN-based architecture defines the problem as prediction of a punctuation class (including ”no punctuation”) at each position coming before or after a word input at each step, as in Figure 4.5). The words are input to the network as vectors (using word embeddings) and are accompanied with prosodic features if there is a prosodic modelling involved.

RNN-based work that combines lexical and prosodic features (Tilk and Alumäe, 2015, 2016) show incorporation of prosodic features into the punctuation mod-



Word sequence	he	who	knows	does	not	speak	he	who	speaks	does	not	know
Punctuation after	ø	ø	,	ø	ø	.	ø	ø	,	ø	ø	.

Figure 4.5: Modelling punctuation as a classification problem at each word interval. (Quote by *Lao Tze*)

elling only through a limited dimension. Firstly, prosodic modelling is only limited within pauses used as sole prosodic feature. Secondly, prosodic modelling is done in a secondary training step which is the only step that involves introduction of spoken language. Model is biased on huge amounts of written data which shows differences in terms of both language and punctuation usage compared to spoken language.

In this section I will address my motivation that prosody should be considered in a more complete way for the task of punctuation restoration. I will first define a set of features that could be used for modelling punctuation in spoken language within a neural network based setting. Secondly, I will explain a RNN-based architecture that is able to process this information and also allows testing of which prosodic features influence punctuation placement to what extent.

4.3.1. Features for Punctuation Modelling

Syntactic information proves to be one of the main features in modelling punctuation as in the works Che et al. (2016); Batista et al. (2012); Ballesteros and Wanner (2016). Syntactic influence to punctuation is defined by the grammatical rules of a language as well as sentence structure. Sentence boundary marking is the most common use of a punctuation mark. The type of the punctuation terminating a sentence is influenced by the modality of the sentence (statement, question, command etc.) Each of these modalities often influence which type of words are used in which order in the sentence. For example in English, a WH-question would include one of the WH-words (what, which, how etc.). Whereas a yes/no question can be discriminated by the order of the verb and subject (*It is...* vs. *Is it...*). Usage of comma, which is a non-sentence-ending punctuation mark, is many times required by certain syntactic structures in a language again signaled by the lexical content. These include relative clauses or presence of initial temporal information as in examples below:

1. *Today, I will start jogging.*

2. *It is, however, extremely difficult to identify all the relevant variables.*

3. *Adam's new van, which is less than a month old, makes a lot of noise.*

A RNN-based network is able to model the language by processing sequences of words being represented as vectors. These vectors, which are also called *word embeddings*, are able to represent words in their morphological forms capturing their semantics, syntactic behaviour and morphological structures (Ballesteros and Wanner, 2016).

In the proposed methodology, as morphosyntactic features, word embeddings and part-of-speech (POS) tags are used. POS tagging of the words were available as a supporting syntactic feature for English language through the NLTK toolkit (Bird et al., 2009).

For modelling prosodic influence on punctuation generation, four main acoustic-prosodic features are employed: pauses, pitch, intensity and speech rate. Pause features indicates the duration of silence between previous word and the current word. As pitch and intensity features vary between speaker to speaker, a scaling method is used for these features to convert the measured values to relative scales. Fundamental frequency (F0) in Hertz and intensity in decibels are converted to scales relative to the speaker's norm using the expression:

$$semitone(x, norm) = 12 * \log(\frac{x}{norm}) \tag{4.1}$$

This is done to ensure the prosodic features represent the variations with respect to the mean rather than absolute values that may differ across speakers.

To align pitch and intensity features to the utterance, mean and range values are calculated at word level so that each word can be associated with the pitch and intensity level corresponding to it. Range values are calculated by subtracting the minimum pitch and intensity values respectively from the maximum pitch/intensity value in the contour corresponding to the word. If a word is unvoiced or a measurement fails, its mean and range values are set to 0 which corresponds to the speaker mean value in the normalized scale.

Farrús et al. (2016) states speech rate as a discriminating feature in determining paragraph boundaries. To test its effect on sentence boundaries, it is included as a feature as well. Speech rate is calculated at each word by dividing the number of syllables in that word with the word's duration. It is then normalized according to the speaker's mean value. A complete list of the features with their abbreviations is given in Table 4.1.

Feature	**ID**
Word vector	word
Part-of-speech tag	pos
Pause before	pause
Mean pitch	mean.f0
Pitch range	range.f0
Mean intensity	mean.i0
Intensity range	range.i0
Speech rate	speech.rate

Table 4.1: Morphosyntactic and prosodic features used in the punctuation restoration framework.

4.3.2. Model Architecture

The architecture of the model is inspired by the methodology presented in Tilk and Alumäe (2015) and Tilk and Alumäe (2016). These works employ a 2-stage training approach, as depicted in Figure 4.6. Two recurrent neural networks (RNN) are chained where the first one processes the words and the second one adds pause duration between two consecutive words as an additional feature to the output of the first network. The two stage architecture is employed for the lack of audio data compared to text data. In Tilk and Alumäe (2016), the network is further enhanced to process words in two directions using a bidirectional recurrent network (Schuster et al., 1997) with attention. As RNN layers, *gated recurrent units* (GRU) are used (Cho et al., 2014) which were explained earlier in Section 2.1.1.

The modifications to Tilk and Alumäe's architecture are that (1) instead of passing prosodic feature values in a second stage, they are introduced to the model through separate parallel GRU layers that are tuned in one single stage, and (2) the proposed network is easily scalable so that it facilitates experimentation with different sets of features and configurations. The system can be configured to take any discrete features (e.g. word, part-of-speech (POS)) and prosodic features (e.g. f0 and intensity) to build a parallel layered network. Suprasegmental acoustic/prosodic features such as fundamental frequency, intensity and speech rate are aligned with words by taking the mean value corresponding to each word.

I will now explain a possible model that could be generated by the proposed framework. For the sake of simplicity, the model will use as input: words (w) as the sole lexical feature and inter-word pause durations (p) and word-level pitch (m) as prosodic features. As for output, a punctuation class (period, question

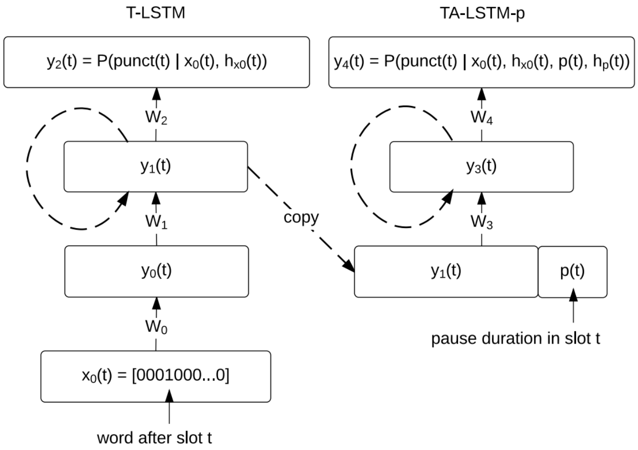


Figure 4.6: Two stage architecture of Tilk and Alumäe (2015) (source of the diagram) which is later extended with bidirectional RNN layers in Tilk and Alumäe (2016).

mark, comma or no punctuation) is given at training and predicted at inference. This architecture is illustrated in Figure 4.7. It can be seen that, the model has 5 input GRU units: bidirectional layers for words, bidirectional layers for pitch values corresponding to words (denoted as *mean.f0*), and a unidirectional layer for pauses coming before the words. Word GRU layers are preceded by an embedding layer (W_e). Inputs to the embedding layers are one-hot encoded vectors of sizes respective to the word vocabulary size. The hidden states of the GRU layers at time step t are:

$$\overrightarrow{h_w}(t) = GRU(x(t)W_e,\ \overrightarrow{h_w}(t-1)) \tag{4.2}$$

$$\overleftarrow{h_w}(t) = GRU(x(t)W_e,\ \overleftarrow{h_w}(t+1)) \tag{4.3}$$

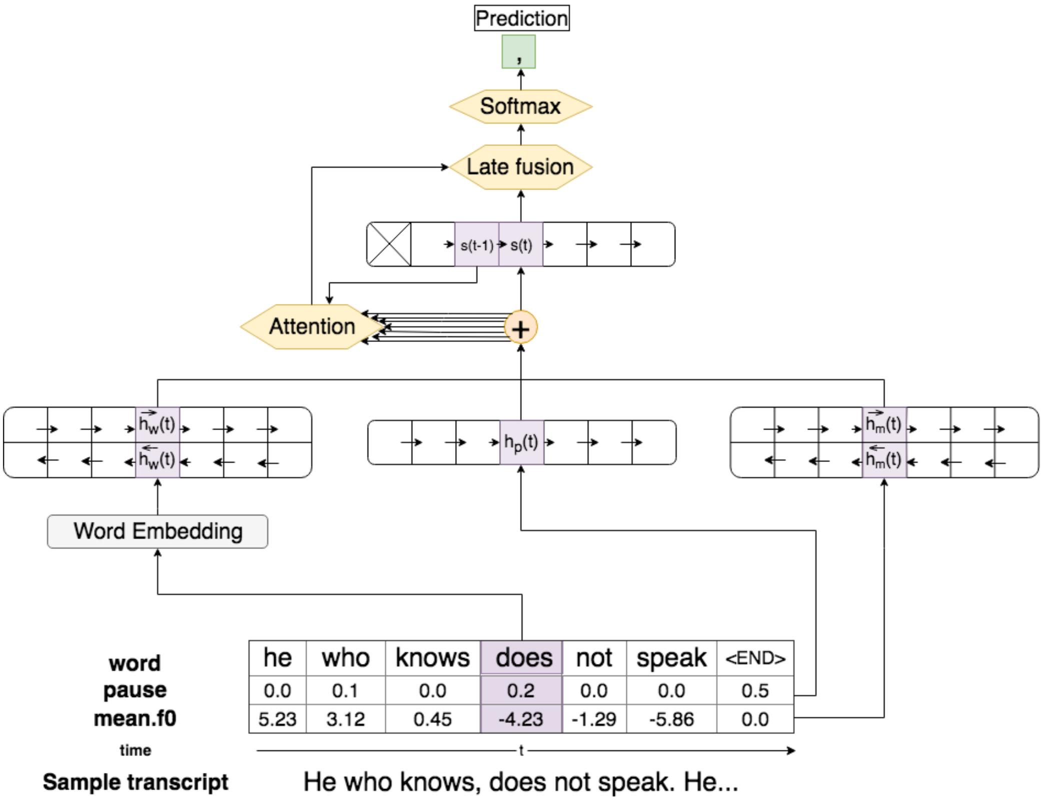
$$h_p(t) = GRU(p(t)W_p,\ h_p(t-1)) \tag{4.4}$$

$$\overrightarrow{h_m}(t) = GRU(m(t)W_m,\ h_m(t-1)) \tag{4.5}$$

$$\overleftarrow{h_m}(t) = GRU(m(t)W_m,\ h_m(t-1)) \tag{4.6}$$

where $x(t)$, $p(t)$ and $m(t)$ are the word index, pause duration and mean f0 value respectively at time step t. The parallel GRU states are concatenated to form the context vector $h(t)$ before being passed over as input to another unidirectional GRU layer:

$$h(t) = \left[\overrightarrow{h_w}(t), \overleftarrow{h_w}(t), h_p(t), \overrightarrow{h_m}(t), \overleftarrow{h_m}(t)\right] \tag{4.7}$$



word	he	who	knows	does	not	speak	<END>
pause	0.0	0.1	0.0	0.2	0.0	0.0	0.5
mean.f0	5.23	3.12	0.45	-4.23	-1.29	-5.86	0.0

Figure 4.7: Our neural network architecture depicting processing of a speech data sample with pause and mean f0 features aligned at the word level.

$$s(t) \;=\; GRU(h(t),\; s(t-1)) \tag{4.8}$$

The attention mechanism combines all input states into a weighted context vector $a(t)$, which is then late-fused with the state $s(t)$ of the output GRU layer:

$$a(t) \;=\; \sum_{i=1}^{N} h(t)\alpha_{t,i} \tag{4.9}$$

$$f(t) \;=\; a(t)W_{fa} \bigodot \sigma(a(t)W_{fa}W_{ff} \;+\; s(t)W_{fs} \;+\; b_f) \;+\; s(t) \tag{4.10}$$

where $\alpha_{t,i}$ is the weight that determines the amount of influence of each input state to the current output and N is sequence size. The late-fusion approach lets the context gradient carry on easily by preventing it to pass through many activation functions (Wang and Cho, 2015). Finally, the late-fused context $f(t)$ is passed through a *Softmax* layer, which outputs a vector containing probabilities of the punctuation classes to be placed between the current and the previous word (starting from the second word in sequence):

$$y(t) = Softmax(f(t)W_y + b_y) \tag{4.11}$$

Two key concepts introduced in Tilk and Alumäe (2016) are: (a) bidirectionality and (b) attention mechanism. Bidirectional layers help to carry information from both past and future context with respect to the currently processed word. In the presented architecture, words as well as some prosodic features are processed bidirectionally. The attention mechanism is useful for the neural network to identify positions in a sequence where important information is concentrated (Bahdanau et al., 2014). For words, it helps to focus on positions of words and word combinations that signal the introduction of a punctuation mark. For prosodic features, it either remembers a salient point in the sequence or detects a certain movement that could help determining a punctuation mark at a certain position.

4.4. Experiments

In this section, I will explain the implementation process of the proposed methodology regarding data preprocessing and selection of hyperparameters. Later, using the models obtained from various setups, I will explore the following questions through experimentation:

1. How do prosodic features in speech affect punctuation placement?
2. What's the effect of punctuation presence to syntactic parsing?
3. How do the obtained models perform within a speech recognition interface?

4.4.1. Data and Preprocessing

Taking into account that the number of words per sentence in the TED Talks Corpus is 15–20 in average, the data is sampled into sequences of 50 words. Samples are extracted sequentially from talks. Each sample starts with a new sentence. Once no more complete sentences fit into the sample, the rest of the sample is padded with empty tokens. Sentences with more than 50 words are discarded. This was to ensure input to the system was complete so that the model always places a punctuation mark at the end of an utterance. Finally, a sample consisted of 2.6 sentences in average.

51,311 samples were extracted this way. 70% percent (39,419 samples) of this data were allocated for training, 15% for testing and 15% for validation (8,446 samples each). The word vocabulary was created with the tokens that occur more than 6 times in the corpus and three extra tokens: *out-of-vocabulary*, *end-of-sequence* and *empty*. This totaled up to 13,031 tokens. The output punctuation vocabulary in the experiments include 4 classes: period, question mark, comma and no punctuation.

4.4.2. Implementation and Hyperparameters

Theano (Theano Development Team, 2016) was used for implementing the models. In the experimental setup, the word embedding vector sizes for words were set to 100 which were initialized randomly at the beginning of the training. The hidden layer dimension of all GRU layers is also set to 100, except for pause durations and POS, where a smaller dimension of 2 and 10 respectively performed better in terms of validation scores. Besides words, mean pitch and intensity values are also processed in bidirectional RNN layers.

The models were trained in batches of size 128. The weight matrices are updated using the *AdaGrad* algorithm (Duchi et al., 2011) with a learning rate of 0.05 for minimizing the negative log-likelihood of the predicted punctuation sequence.

Two ways of inputting prosodic features into the model are tested. First, values are input as their absolute values in a continuous fashion, i.e. pause durations in seconds, mean f0 values in semitones, intensity values in dB, and speech rate normalized within -1 and 1. Secondly, they are inputted as discrete values (in levels) and passed through an embedded layer similar to words and POS features. Leveling of the prosodic values was done by dividing each feature's normal distribution to quantiles of 100, so that more frequent ranges are represented more precisely.

4.4.3. How do prosodic features in speech affect punctuation placement?

This section reports on the results obtained by training various models using different feature settings in terms of punctuation restoration accuracy. As the majority of the punctuation marks in the dataset consists of the punctuation marks in the reduced set (comma, period and question mark), experiments were performed only with this set.

The two-stage method by Tilk and Alumäe is used as a baseline by training over the data twice: first, only with text, and then together with the pause durations. Tilk et al.'s models are based on BRNN with an attention mechanism, which provided the best results when compared to other models (Tilk and Alumäe, 2016).

In the proposed single stage approach, the use of only lexical information (words) provides the same scores as the use of only words in the two-stages approach, since only one step is involved in both approaches. In order to assess the contribution of new prosodic information to our model, more prosodic features were added one by one until no improvement is recorded.

Results

The outcome of experiments in generating periods, commas and question marks with different settings of features are listed in Table 4.2 and illustrated in Figures 4.8a and 4.8b. The results reported here use prosodic feature as continuous values in order to ensure comparability with the baseline. The best performing model was re-trained using discretized features and reported as well with the label *discretized*. Additionally, a model with only prosodic features that ignores word information was tested (labeled as *no words*).

Compared to the two-stage models performance on the same dataset, first improvement is achieved through employing of the parallel processing architecture: An average improvement for all punctuation marks in terms of F_1 score of 0.4% when the same features (word and pause durations) are used. The model opens the way for a further improvement of 2% with the addition of pitch and POS feature into the model, and finally, with the introduction of discretized prosodic features, an overall F_1 score of 70.3% is obtained.

By incrementally adding features on top of the word-based model, it is observed that usage of pause durations and part of speech (POS) improves period and question mark generation, and a combination of them results in an improvement in terms of F_1 score for all punctuation marks. The results also show that each punctuation mark has different sets of prosodic features that work the best for them. The best result for generation of commas in terms of F_1 score is observed with pause and pitch mean features (55.2%). For period, mean intensity helps in terms of recall and a combination of it with pause and pitch mean results in best performing F_1 score of 82.0%. For question mark, however, pitch and intensity features do not lead to an improvement, as the best result is achieved with the pause feature only (71.8% F_1 score).

Even without any textual features, silence, pitch and intensity features are able to determine sentence boundaries to a certain extent. A solely prosodic feature based model gives a precision of 71.3% and an F_1 score of 55.7% in detecting

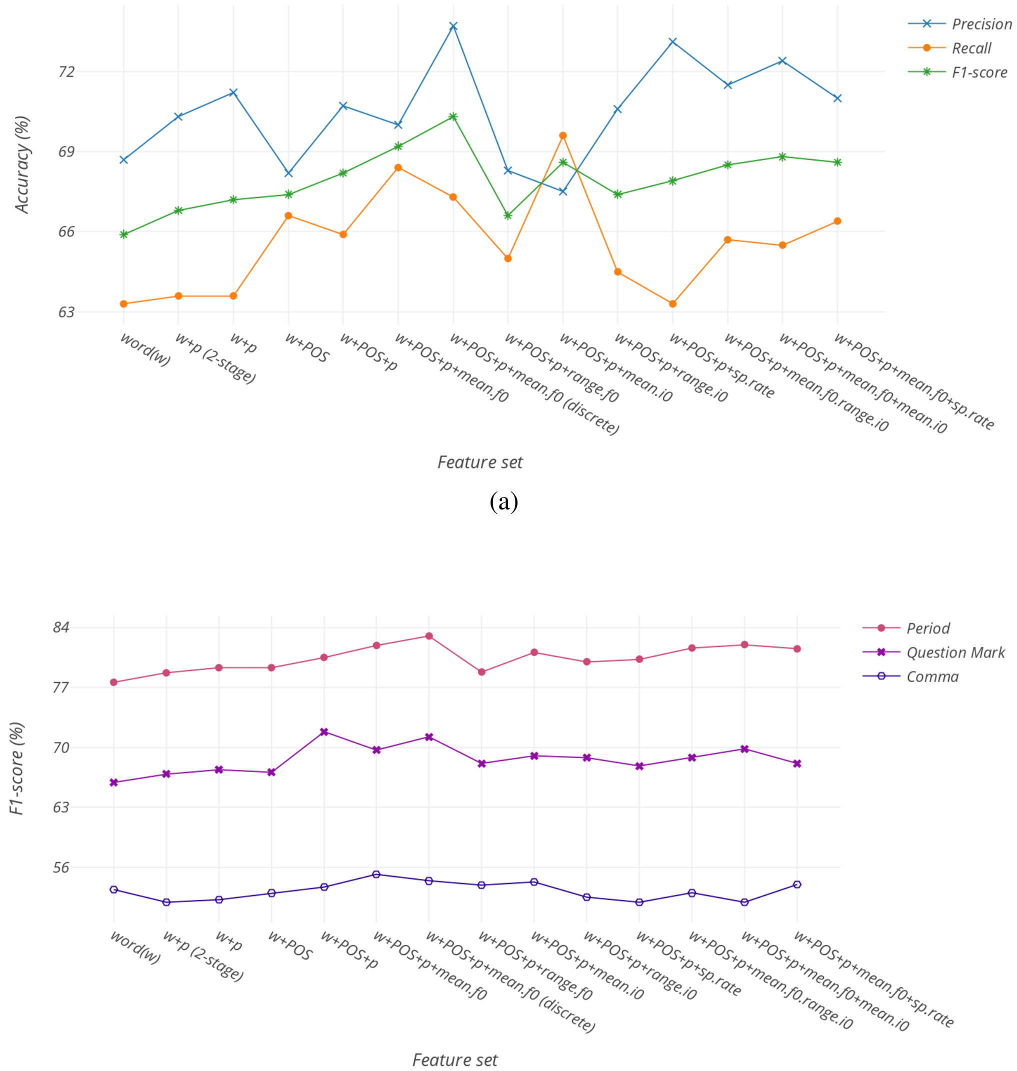
	Comma			Period			Question			All		
Feature set	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Baseline (Tilk and Alumäe, 2016)												
word(w)	54.2	52.6	53.4	82.9	73	77.6	70.89	61.5	65.9	68.7	63.3	65.9
w+pause	58.8	46.5	51.9	78.4	79.1	78.7	70.89	63.3	66.9	70.3	63.6	66.8
Proposed model												
w+pause(p)	58.9	46.9	52.2	80.0	78.6	79.3	73.1	62.5	67.4	71.2	63.6	67.2
w+POS	56.3	50.0	53.0	76.7	81.9	79.3	73.2	62.0	67.1	68.2	66.6	67.4
w+POS+p	56.7	51.0	53.7	82.0	79.1	80.5	75.8	68.2	**71.8**	70.7	65.9	68.2
w+POS+p+mean.f0	55.4	54.9	**55.2**	83.0	80.7	81.9	73.5	66.4	69.7	70.0	68.4	**69.2**
w+POS+p+range.f0	53.0	**54.8**	53.9	**83.8**	74.4	78.8	75.0	62.3	68.1	68.3	65.0	66.6
w+POS+p+mean.i0	53.9	**54.8**	54.3	79.2	**83.0**	81.1	70.1	67.9	69.0	67.5	**69.6**	68.6
w+POS+p+range.i0	56.8	48.7	52.5	81.3	78.7	80.0	74.0	64.3	68.8	70.6	64.5	67.4
w+POS+p+sp.rate	**61.5**	44.9	51.9	80.6	80.0	80.3	76.3	61.0	67.8	**73.1**	63.3	67.9
w+POS+p+mean.f0+range.i0	57.0	49.5	53.0	82.7	80.5	81.6	**77.6**	61.8	68.8	71.5	65.7	68.5
w+POS+p+mean.f0+mean.i0	59.4	46.2	51.9	81.4	82.6	**82.0**	71.5	68.1	69.8	72.4	65.5	68.8
w+POS+p+mean.f0+sp.rate	56.7	51.6	54.0	83.7	79.4	81.5	66.4	**69.8**	68.1	71.0	66.4	68.6
p+mean.f0+mean.i0 (no words)	33.8	1.1	2.1	71.3	45.8	55.7	0.0	0.0	0.0	69.9	23.7	35.4
w+POS+p+mean.f0 (discretized)	61.3	48.9	**54.4**	82.6	**83.5**	**83.0**	71.8	**70.6**	71.2	**73.7**	67.3	**70.3**

Table 4.2: Punctuation generation results for two stages baseline and the proposed single-stage approach. P, R and F_1 stands for precision, recall and F_1 score respectively in percentage (%).

periods. When textual features are added to this set, it performs as the best model for generating periods. Although a sentence-ending punctuation mark, the question mark does not show the same behavior as it is much less represented in the dataset.

The improved scores, which are achieved through discretization of continuous prosodic features, present new parameters of the neural network architecture that could be used to boost its accuracy. It proves that reducing the parameter space and representing leveled features in an embedded space can improve results in similar tasks.

Table 4.3 shows some examples from the testing set punctuated with solely word-based and word-prosody combined model. Listening to the audio samples, one can spot some examples that show improvement caused by prosodic models (sentences 2c and 4c in Table 4.3). However, other examples (sentences 1c and 3c in Table 4.3) also show that there are some cases where inclusion of prosodic features do not necessarily help the correct prediction of punctuation. In the case of sentence 1c, the speaker consistently makes pauses after most words and makes prominent most content words. That might be the reason why prosodic features do not help establish the correct punctuation after the word *axons*. On the other hand, the sample for sentence 3a points out that the model that includes prosodic features has some limitations as it inserts a comma in the middle of a clearly au-



(a)

(b)

Figure 4.8: (a) Overall punctuation results in terms of precision, recall and F_1 score (b) F_1 score of each punctuation mark in different feature settings.

dible prosodic unit *video cassette recorders*. One plausible solution to overcome this limitation may be testing including other features or giving more weight to prosodic features over textual ones.

4.4.4. What's the effect of punctuation presence to syntactic parsing?

The first step of many NLP applications involves the parsing of an input phrase to the system. In a system with human input for example, it allows further interpretation of the input phrase through a syntactic analysis. The output of a syntactic parser is a *dependency tree*, where the sentence is defined as a group of relations between the elements of a sentence. Figure 4.9 illustrates an example of a dependency tree. Most syntactic parsers are statistical in a sense that they determine these relations based on knowledge gained from a huge corpus of hand-annotated dependency trees. Words, as well as punctuation marks are the nodes of dependency trees.

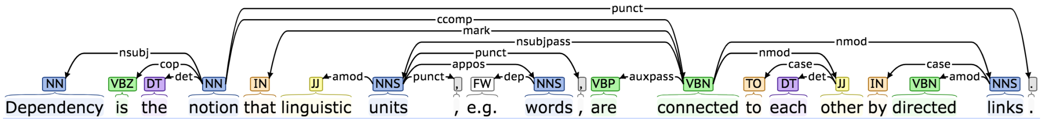


Figure 4.9: An example of a dependency tree generated with an English parser[4].

As much as human understanding of written language is affected by it, syntactic parsers also depend on punctuation marks on input sentences (Jones, 1994). First and most important cue lying in punctuation is the sentence boundaries. Syntactic parsing is generally performed on a sentence. Thus, parsing of huge texts imply segmentation of it from sentence boundaries. Other punctuation marks have also effect on parsing as they are grammatical and semantic elements of a sentence.

In this section, I will perform an experiment on examining the effect of punctuation placed by the trained prosodic punctuation models on dependency parsing. Specifically, I wanted to examine if the commas predicted with our models help the parsing. As wrong placement of commas could decrease the parser accuracy, I wanted to test if the relatively low-scoring comma prediction helps the parser output.

ID	Model	Sentence
1a	Gold	So all of those colored lines correspond to bunches of axons,⃝ the fibers that join cell bodies to synapses⊙
1b	Word	so all of those colored lines correspond to bunches of axons,⃝ the fibers that join cell bodies to synapses⊙
1c	W&Pr	so all of those colored lines correspond to bunches of axons the fibers that join cell bodies to synapses⊙
2a	Gold	Now molecules are really,⃝ really tiny⊙
2b	Word	now,⃝ molecules are really really tiny⊙
2c	W&Pr	now,⃝ molecules are really,⃝ really tiny⊙
3a	Gold	Cassette tapes,⃝ video cassette recorders,⃝ even the humble Xerox machine created new opportunities for us to behave in ways that astonished the media business⊙
3b	Word	cassette tapes,⃝ video cassette recorders,⃝ even the humble xerox machine,⃝ created new opportunities for us to behave in ways that astonished the media business⊙
3c	W&Pr	cassette tapes,⃝ video,⃝ cassette recorders,⃝ even the humble xerox machine created new opportunities for us to behave in ways that astonished the media business⊙
4a	Gold	And you could see how my poor,⃝ manipulated sister faced conflict,⃝ as her little brain attempted to devote resources to feeling the pain and suffering and surprise she just experienced,⃝ or contemplating her new found identity as a unicorn⊙
4b	Word	and you could see how my poor manipulated sister faced conflict as her little brain attempted to devote resources to feeling the pain and suffering and surprise,⃝ she just experienced or contemplating her new found identity as a unicorn⊙
4c	W&Pr	and you could see how my poor manipulated sister faced conflict,⃝ as her little brain attempted to devote resources to feeling the pain and suffering and surprise she just experienced,⃝ or contemplating her new found identity as a unicorn⊙

Table 4.3: Punctuation generation results for a set of sentences. Audio samples can be accessed from the Github repository[3].

Experimental setup

One hundred sentences that represent different types of comma events have been collected from our test set. The collection includes simple to complex sentences with commas used for different functions; e.g., enumeration, dislocation of noun phrases, and clause division, among others. Some sentences in which different placement of commas could lead to different semantic or syntactic structures have also been chosen.

Sentences with gold punctuation (from original annotations), with punctuation taken out, and with predicted punctuation (with and without prosodic features) were parsed using a state-of-the-art dependency parser (Bohnet and Kuhn, 2012; Bohnet and Nivre, 2012). In order to compare the parsing results, the standard dependency parser quality metrics *Unlabeled Attachment Score (UAS)* and *Labeled Attachment Score (LAS)* are used. For assessing the closeness of two dependency trees, UAS measures the number of arcs with correct head and dependencies. On top of UAS, LAS measures whether the dependency labels are correct (Buchholz and Marsi, 2006). For more information on UAS and LAS refer to Nivre and Fang (2017); Green (2011).

Results

The results listed in Table 4.4 show that dependencies are labeled wrong in 16.6% of the cases if punctuations are omitted; cf. the corresponding LAS. Labeled dependency trees get more similar to the gold standard with the introduction of commas using our models. Thus, LAS improves by 5% when only word features and by 5.7% when both word and prosodic features are used, resulting in a decreased error rate of 10.9%. A similar tendency can be observed with UAS. The results show that consideration of prosody improves dependency parsing.

	Similarity	
Setting	LAS	UAS
Unpunctuated	83.4%	86.3%
Punctuated with word feature	88.4%	89.8%
Punctuated with word and prosodic features	89.1%	90.6%

Table 4.4: Parsing similarity results

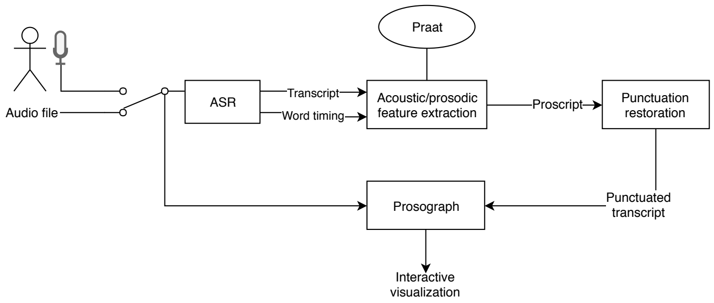


Figure 4.10: Architecture of the interactive ASR testing setup.

4.4.5. Performance with ASR output

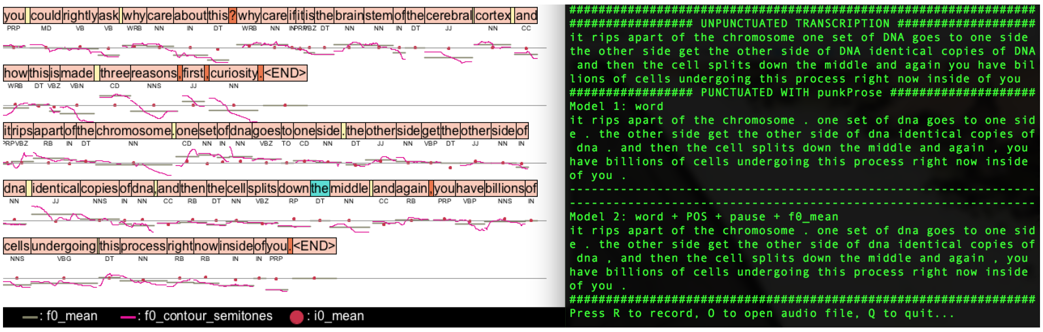
In this section, I further extend on the performance tests by attaching the developed prosodic punctuation restoration models into a speech recognition pipeline. A testing interface is prepared that uses a state-of-the-art ASR system to convert spoken input to raw transcription and *Prosograph* (Öktem et al., 2017b) to analyze the input prosody. Punctuation restoration is then performed on the raw transcripts with the two types of models models that were trained during the experiments: Text-only model and best performing text+prosodic model. Through these tests it is possible to see the performance of the proposed methodology on a setting closer to a real-world use case (Öktem et al., 2018a).

Overview of the testing interface

The testing interface is designed so that spoken input can be given to the interface either by recording with microphone or by presenting a pre-recorded file which is then sent to an ASR system for transcription. The transcriptions, punctuated with our models, are displayed together with their graphical prosodic visualizations.

As depicted in Figure 4.10, the pipeline of the testing interface can be summarized as follows: (1) Obtaining a recording from either microphone or a waveform audio file, (2) transcription using a speech-to-text system[5], (3) prosodic and syntactic feature extraction, (4) punctuation restoration, (5) visualization of punctuated versions of transcript together with acoustic measurements. See Figure 4.11 for example of the testing interface.

[5] Google's Cloud ASR service is employed.


you could rightly ask why care about this ? why care if it is the brain stem of the cerebral cortex and
PRP MD VB VB WRB NN IN DT WRB NN IN PRP VBZ DT NN NN IN DT JJ NN CC
how this is made three reasons , first , curiosity . <END>
WRB DT VBZ VBN CD NNS JJ NN
it rips apart of the chromosome . one set of dna goes to one side . the other side get the other side of
PRP VBZ RB IN DT NN CD NN IN NN VBZ TO CD NN DT JJ NN VBP DT JJ NN IN
dna identical copies of dna , and then the cell splits down the middle and again , you have billions of
NN JJ NNS IN NN CC RB DT NN VBZ RP DT NN CC RB PRP VBP NNS IN
cells undergoing this process right now inside of you . <END>
NNS VBG DT NN RB RB IN IN PRP
: f0_mean
: f0_contour_semitones
: i0_mean

```
################################################################
################# UNPUNCTUATED TRANSCRIPTION ####################
it rips apart of the chromosome one set of DNA goes to one side
the other side get the other side of DNA identical copies of DNA
 and then the cell splits down the middle and again you have bil
lions of cells undergoing this process right now inside of you
################# PUNCTUATED WITH punkProse ####################
Model 1: word
it rips apart of the chromosome . one set of dna goes to one sid
e . the other side get the other side of dna identical copies of
 dna . and then the cell splits down the middle and again , you
have billions of cells undergoing this process right now inside
of you .
----------------------------------------------------------------
----------------------------------------------------------------
Model 2: word + POS + pause + f0_mean
it rips apart of the chromosome . one set of dna goes to one sid
e . the other side get the other side of dna identical copies of
 dna , and then the cell splits down the middle and again , you
have billions of cells undergoing this process right now inside
of you .
################################################################
Press R to record, O to open audio file, Q to quit...
```

Figure 4.11: The two window interactive test environment. Recordings are presented through the command line interface (right) and visualized directly on Prosograph (left).

Selected testing samples

Here I will show some samples from the Heroes Corpus running through the testing interface. I test on examples from the movie domain to get insight on how the proposed model would work on an automatic captioning use case.

Figure 4.12 shows an example that was recognized well with the ASR system. Both text and prosodic punctuation restoration models perform well in recognizing the sentence boundary. Figure 4.13 illustrates an example where the ASR system fails to recognize the speech input accurately. In this example, textual model works better in determining the boundary marked by a comma in the original sentence. Even though the boundary is marked by a long pause, the prosodic model doesn't perform well due to the ungrammatical structure of the recognized utterance.

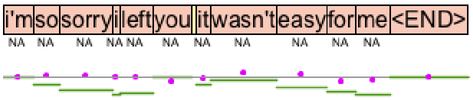
Figure 4.14 also illustrates a misrecognized example. The tag question "don't you" at the end of the speech sample is not recognized. Without that part, the text model marks the sentence end with a period. However, the prosodic model captures the intonation of the sample and predicts successfully the question mark at the end. This example shows that the prosodic model is able to predict well in some cases even though ASR fails to recognize the input speech accurately.

All in all, the models trained on conference speeches show an acceptable and usable performance on ASR output and on a different domain than the models are trained on. With further domain adaptation better results can be obtained.

Original utterance

I’m so sorry I left you. It wasn’t easy for me.

Prosodic visualization



Punctuation restored (Word model)

i’m so sorry i left you. it wasn’t easy for me

Punctuation restored (Word+Prosodic model)

i’m so sorry i left you. it wasn’t easy for me

Figure 4.12: Segment pair s2_5_0227 from the Heroes corpus

4.5. Conclusion

In this chapter, I have presented a recurrent neural network architecture that processes lexical and prosodic information in parallel for the generation of punctuation in speech transcripts, avoiding the dominance of written data, and thus the bias of trained models towards written material. The proposed model allows the integration of any desired feature (lexical, syntactic or prosodic) and thus a further analysis of the impact of every feature used on the punctuation generation. In addition, the current model achieves a significant improvement over previous works that used two stages and were biased to written data. An overall F_1 score of 70.3% is reported for comma, period and question mark generation in the used dataset.

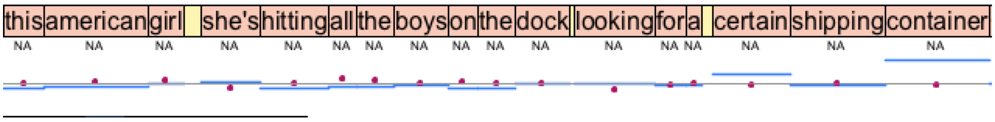
The results are shown to be significantly better when syntactic and prosodic features are added to the lexical information. Solely pauses —when trained with a separate RNN — improve considerably the vocabulary-based scores. Moreover, f0- and intensity-based prosodic features help to achieve a better comma and period detection in terms of F_1 measure. All in all, the best combination of prosodic features is when the model is trained on words, their POS tags together with the preceding pause durations and their normalized mean f0 values.

Further experiments have been carried out to test the performance of the models on parsing and ASR output. Quantitative metrics on parsing of single sentences showed that prosodic models perform better in obtaining correct dependency trees with syntactic parsing. On a demonstrative setting where ASR was employed, reasonable performance is recorded in recovering punctuation marks on out-of-domain spoken input. Through further model adaptation (e.g. vocabulary extension and speaker adaptation) better results can be obtained.

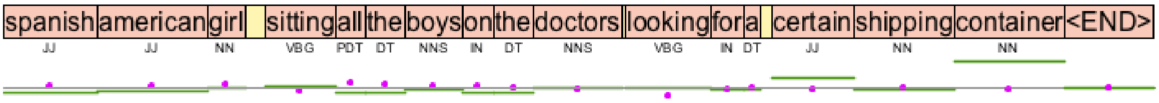
Original utterance

This American girl, she's hitting all the boys on the dock... looking for a certain shipping contain

Original Prosodic visualization


this american girl she's hitting all the boys on the dock looking for a certain shipping container
NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA

Prosodic visualization of recognized speech


spanish american girl sitting all the boys on the doctors looking for a certain shipping container <END>
JJ JJ NN VBG PDT DT NNS IN DT NNS VBG IN DT JJ NN NN

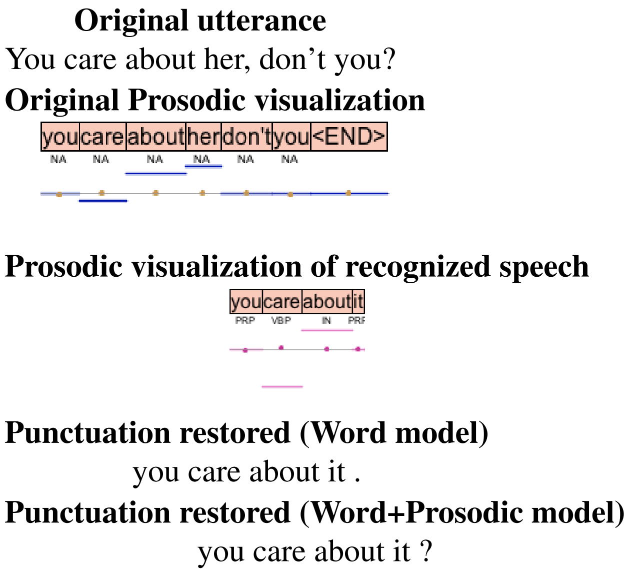
Punctuation restored (Word model)

spanish american girl, sitting all the boys on the doctors looking for a certain shipping container.

Punctuation restored (Word+Prosodic model)

spanish american girl sitting all the boys on the doctors looking for a certain shipping container.

Figure 4.13: Segment pair s2_5_0107 from the Heroes corpus



Original utterance

You care about her, don’t you?

Original Prosodic visualization

Prosodic visualization of recognized speech

Punctuation restored (Word model)

you care about it .

Punctuation restored (Word+Prosodic model)

you care about it ?

Figure 4.14: Segment pair s2_5_0114 from the Heroes corpus

Chapter 5

ENHANCING SPOKEN LANGUAGE TRANSLATION WITH PROSODY

This chapter explores around the question of how can prosody be utilized in the framework of spoken language machine translation (SLMT). The first goal of this chapter is to gain insights and prove that prosody is an essential element to consider in spoken language translation. This is performed through linguistic and corpus-based analysis on the bilingual speech corpus that we collected (Section 5.2). Following, building of a neural machine translation system is explained in Section 5.3. This system serves both as a text translation baseline and a basis for incorporation of prosodic features in both input and output. Next, I perform experiments that utilize this system on the movie-domain translation (speech-to-text and speech-to-speech). First, I explore the effect of prosodic punctuation restoration as a preliminary step to translation in Section 5.4. Secondly in Section 5.5 I aim to improve text translation system through prosodically-enhanced input. And finally, for the aim of generating prosodic synthesis cues in a speech-to-speech translation system, I report on the experiments building a translator that can handle prosodic input and output (Section 5.6).

5.1. Motivation and Background

Spoken language machine translation is a type of machine translation where input and/or output to the system is spoken language. It is usually used in the context of translating from speech to text (through incorporation of ASR), or speech to speech (through incorporation of ASR and TTS). The MT modules in these systems are designed the same way as standard text translation systems. However, spoken language processing introduces its distinct challenges. For instance, in a system with speech input, the output of ASR lacks punctuation, which provides both linguistic and functional cues for translation. In the case of a spoken input and output system, through a basic concatenation of ASR, MT and TTS systems, prosodic information of the input speech is lost already in the first step. Thus, any communicative information residing in the input speech through prosody is not reflected in the translated and synthesized speech.

One can draw an analogy of the difference between spoken language translation and written language translation as the difference between book translation and movie dubbing. A book translator translates a book chapter by chapter, then paragraph by paragraph, and then sentence by sentence. All these segmentations are cued through the layout of the book, paragraph breaks and punctuation. Once at a certain sentence, the translator interprets the sentence in the original language of the book and then transforms it into the translation language following author's intentions.

Although essentially a translation task, the art of dubbing a movie requires many more challenges. A similar segmentation process is followed but this time through scene information and actor turns. Once a line of an actor is transcripted, it can be segmented into sentences by looking both at grammatical and auditory aspects. The lines are then translated into the dubbing language by translators with the paralinguistic information such as the tone, intention and intensity noted. Finally, the voice actors vocalize these translations respecting these paralinguistic aspects in the original language of the movie.

The additional tasks involved in the latter process should somehow be considered in an automatic translation/dubbing system of audiovisual content in order to obtain optimal results. The segmentation part requires tasks such as speech activity detection, speaker turn detection and ASR. The work in this chapter assumes that these tasks are already done perfectly and focuses on the translation part of the system and especially on the involvement of prosody to it.

Specifically, I will address these three principal questions that involve prosody in the spoken language translation framework:

1. How does prosodic punctuation restoration affect translation?

2. Does pause encoding improve translation?

3. Can we transfer pausing in speech-to-speech translation?

Before I embark on answering these questions, I will try to give the context to them with a linguistic and corpus study. This study is designed both to inspire the design of a prosodically enhanced translation model and also to help interpret experimental results. The questions will be answered with a practical methodology. A prosodically enhanced translation model needs to be designed and then configured to suit the needs defined by the questions. Through this study, I aim to prove the need for inclusion of prosody in spoken machine translation pipelines and also to introduce a framework that would allow experimentation in this respect.

5.2. Analyzing Significance of Prosody in Machine Translation

In this section, I perform some example-based and statistical analysis on bilingual segments of the *Heroes corpus*, which was presented in Chapter 3. This corpus contains parallel English and Spanish speech segments from a dubbed TV series. The aim is to show how prosody is reflected in dubbing translation. Particularly, I focus on pauses as a prosodic feature. The first part demonstrates on a few examples in the corpus how pausing information influences translation, both for text and audio output. In the second part, I follow a more statistical approach to prove significance of pausing in spoken translation.

5.2.1. Example-based Analysis

In order to gain linguistic insights before building a data-driven model, selected parallel segments from the Heroes corpus are carefully inspected. Specifically, two phenomena are investigated in terms of spoken language translation: How does pausing as a prosodic feature reflects (1) in the translation script and (2) in dubbing (translated voiceover on actor's lip movements). These are then compared to how a classic automated model (machine translation and text-to-speech respectively) performs with same input sentence. All spoken samples presented throughout this chapter can be found in the thesis repository [1].

Indirect transfer of pauses

I will firstly examine the sample *s2_5_0043* from the Heroes corpus. The original punctuated transcription of the English segment is: *He pushed his way in,*

[1] https://github.com/alpoktem/PhDThesis

shoved a gun in my face. Next thing I know, he's flying through that glass. Figure 5.1 shows the Prosograph visualization of English and Spanish version of the sample. Yellow boxes indicate paused intervals between words.

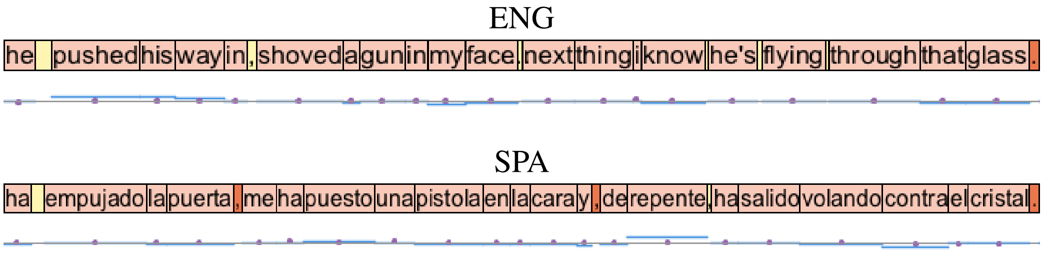

ENG
he pushed his way in , shoved a gun in my face . next thing i know he's flying through that glass .
SPA
ha empujado la puerta , me ha puesto una pistola en la cara y , de repente , ha salido volando contra el cristal .

Figure 5.1: Segment pair s2_5_0043 from the Heroes corpus

Both segments are formed of 4 clauses. English segment consists of two sentences whereas in the Spanish segment these two sentences are joined with a linking word *"y"* (and). Pauses are observed in all clause boundaries in English segment, whereas in the Spanish segment, a clause boundary pause is observed only after *"de repente"* (*lit.* suddenly, non-literal translation of *"next thing I know"*.

A fairly longer non-clause boundary pause is observed at the beginning of both sentences. 0.25 seconds of pause are observed after *"he"* in English and 0.31 seconds of pause are observed after *"ha"* (part of a compound verb to mark past tense) in Spanish. In the last clause in English, two short pauses are observed, which is not reflected in the Spanish sentence. However, when we listen to the Spanish segment, instead of a pause, a lengthening of the word *"ha"* is observed.

With a focus on pauses, we can make the following observations with respect to prosodic realizations in this particular segment pair:

1. Pauses are not necessarily reflected in translation, even though they are induced by grammatical structures like clauses.

2. Even though pauses can be reflected with respect to their position in a sentence, they can come in between different words that do not necessarily correspond.

3. A prosodic phenomena can be expressed in different ways, e.g. by lengthening a word.

This parallel segment shows the complexity of the problem of prosodic transfer. It is hard to predict the prosodic realizations of the translation of a sentence only by looking at prosodic features in the input sentence. It is assumed that the

English and Spanish versions of the segment are expressed in a similar fashion by the two actors, explaining these particular prosodic reflections. However, another voice actor could possibly dub this line in a different way with a different prosodic structure as well.

Next, a state of the art machine translation system is employed to see its performance in translating the this example. Translations are performed using a state-of-the-art commercial neural network based translator[2].

Input sentence (ENG) *He pushed his way in, shoved a gun in my face. Next thing I know, he's flying through that glass.*

MT (ENG → SPA) *Se abrió paso empujándome una pistola en la cara. Lo siguiente que sé es que está volando a través de ese cristal.*

What is noticed first is the mistranslations of some parts of the phrase. However, it is not our point to assess the quality of the translation in terms of correct word usage. The translated phrase represents the actions and objects in the source sentence well enough for our study.

It is examined that the first two clauses in the English phrase are joined into one: *Se abrió paso empujándome una pistola en la cara* (lit. *He opened the way pushing a gun in my face*). Even though there is a comma separating the two clauses explicitly in the input sentence, this is not reflected in the translation. When we translate this section with punctuation marks removed we get a similar result:

Input sentence (ENG) *he pushed his way in shoved a gun in my face*

MT (ENG → SPA) *él se abrió paso empujándome una pistola en la cara*

The phrase, both prosodically and gramatically, is structured in a way that the speaker is explaining a sequence of actions: character pushing in and then pointing a gun on the speaker. Even though this is cued orthographically through punctuation, still the translation system is not able to capture this structure. An ideal translation that takes heed of the prosodic structure would be: *Él se abrió paso, empujó una pistola en mi cara. Lo siguiente que sé, él está volando a través de ese cristal.*[3] This example shows that a translation system that disregards the prosodic structure of the source sentence fails to translate in a way that was originally uttered.

[2] Google Translate: `translate.google.com`

[3] Only changing the structure respecting how the translator performs.

Direct transfer of pauses

Next, I will list some examples where pausing is somewhat more directly transferred between original and dubbing language. Many samples of this type were found in the corpus. Three samples are demonstrated in Figures 5.2 to 5.4.

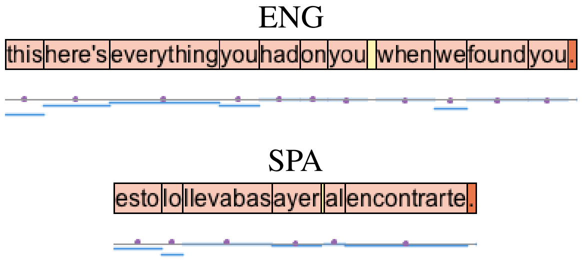

ENG
this here's everything you had on you when we found you .
SPA
esto lo llevabas ayer al encontrarte .

Figure 5.2: Segment pair s2_5_0010 from the Heroes corpus

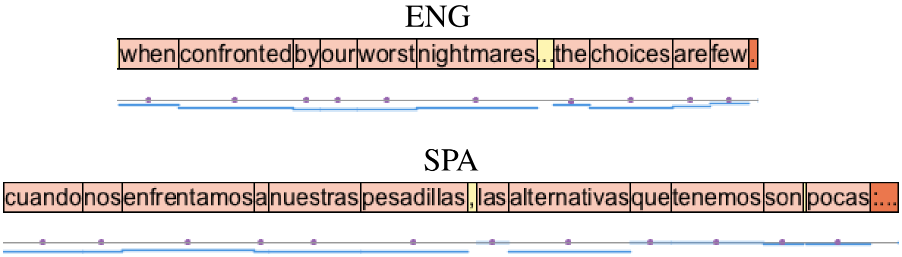

ENG
when confronted by our worst nightmares .. the choices are few .
SPA
cuando nos enfrentamos a nuestras pesadillas , las alternativas que tenemos son pocas :...

Figure 5.3: Segment pair s2_5_0020 from the Heroes corpus

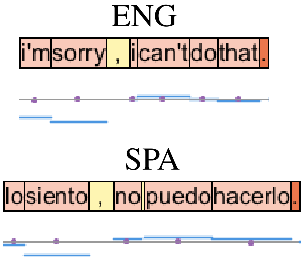


Figure 5.4: Segment pair s2_5_0050 from the Heroes corpus

Pause intervals can be directly traced at the phrase boundaries in both languages. These samples suggest a simpler approach in transfer of pauses. Also, it is observed in most of the examples of this type that the slot where the pause occurs is marked with a punctuation. However, in order to arrive to more concrete conclusions on direct pause transfer and punctuation co-occurrence, a statistical study is conducted in the next subsection.

5.2.2. Corpus-driven analysis

Manual analyses done in the previous subsection are further extended to get a generalized view of the Heroes corpus on pausing. The motivation behind this study is to first, evaluate statistically how pausing is reflected in the dubbing translations, and second, how much pausing is related to punctuation.

How pausing is reflected in translation?

A straightforward scheme is followed to evaluate how much of the pause events in English segments are reflected in the Spanish segments. A segment with a pause event means that there is one or more inter-word interval with a pausing of minimum 0.05 seconds of duration. To quantify this in the Heroes parallel corpus, first, number of segments with a pause event is counted for both English and Spanish segments. Then, number of segment pairs that contain a pause event only in English, only in Spanish and both in English and Spanish is calculated. See Table 5.1 for the results.

	# Segments
Pause in English segment	3050
Pause in Spanish segment	3493
Pause in both English and Spanish	2539
Pause only in English segment	511
Pause only in Spanish segment	954

Table 5.1: Shared pause events in English and Spanish segments of the Heroes corpus.

It can be seen that in 83% of the cases, a pause event in English segment is reflected in the Spanish segment. Other way around, in 72% of the cases, a pause event in Spanish segment is reflected in the English segment. It can be deduced that pausing as a prosodic feature is reflected in the dubbing translations in majority of the cases. In this study, positions of the pauses are ignored.

To what extend pausing is associated with punctuation in subtitle transcripts?

In the manual inspections performed in Subsection 5.2.1, it was observed that many times pauses occur at punctuated slots between words in the subtitle transcription. Below, I explore this on a statistical basis in English segments of the Heroes corpus. The importance of this study is to know how much pausing influence punctuation placement and vice versa. Two directions of co-occurrence are observed: (1) how is a paused interval punctuated? and (2) to what extend punctuation infers a paused interval? I answer the first question in Figure **??** where distribution of punctuation events in paused intervals is shown. In English segments, among 1854 inter-word slots with a pausing, 80% of them are annotated with a punctuation in the subtitle transcripts. The majority of the punctuation marks at these paused intervals are sentence ending punctuation marks (period [.], question mark [?], exclamation mark [!]), whereas comma [,] and ellipsis [...] consist of a smaller percentage. Spanish segments demonstrate a similar behaviour in terms of the ratio of punctuated slots with 78% of them annotated with a punctuation mark. Whereas it is observed that commas tend to be paused more compared to English.

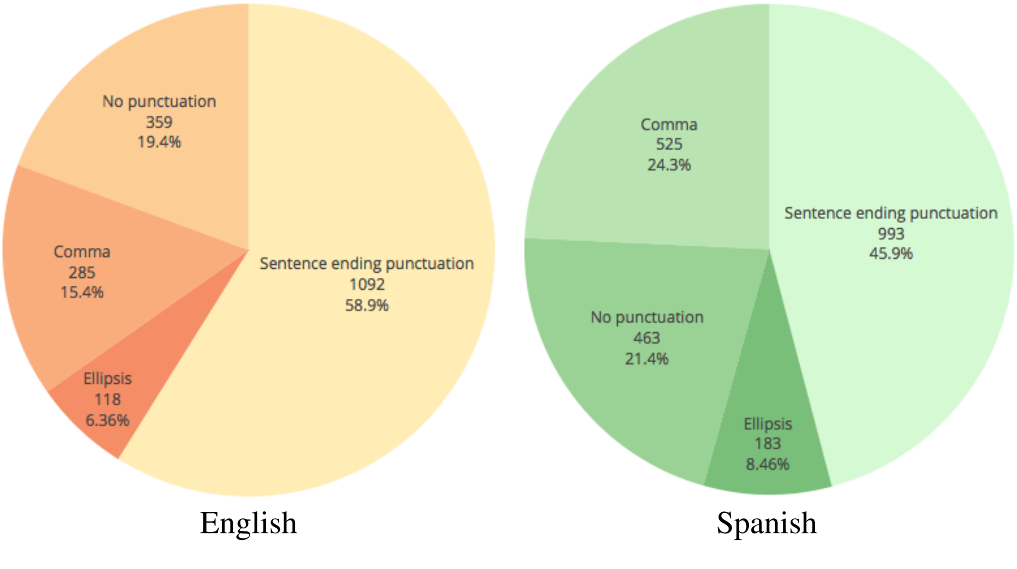


Figure 5.5: Punctuation distribution at paused (> 0.05 s) intervals in English segments(1936 in total) of the Heroes corpus.

Secondly, Table 5.2 shows the distribution of pausing events at inter-word intervals where a punctuation occurs. Looking at English segments, when all punctuation marks are considered, there is a pause in that interval with a 58% of probability. However, when only sentence ending punctuation marks are considered this percentage rises to 75%. It can be deduced that a sentence boundary is a highly discriminating cue for a pausing event between two words. However, the ratio of pause presence at occurrences of comma is quite low (39%). Whereas in Spanish segments, commas seem to be paused much more with a 60% of them marking a short pause of 420 ms in average. Punctuation marks that act as a sentence boundary also mark a pause more than in English segments (86%). Both these contribute to a higher distribution of pausing at punctuation points. 72% of punctuation marks are paused, which is 14% higher than in English segments.

Through these studies it can be confirmed that pausing is a highly correlated phenomena with punctuation. And as much as punctuation acts as an important cue in machine translation, pauses can also have a similar role through complementing punctuation or acting as a sole feature in absence of one.

	#Occurrences	**#Occurrences w/ pause**	**Percentage of paused**	**Mean pause duration (s)**
English				
Punctuated interval	5429	3152	58	0.81 s
Sentence boundary	2549	1913	75	1.02 s
Comma	2652	1038	39	0.38 s
Ellipsis	228	201	88	0.94 s
Spanish				
Punctuated interval	4935	3580	72	0.77
Sentence boundary	1856	1606	86	1.11
Comma	2718	1653	60	0.42
Ellipsis	361	321	88	0.83 s

Table 5.2: Pause presence in punctuated intervals in English and Spanish segments of Heroes corpus.

5.3. Methodology

Having the intuition gained from examining prosodic parallelisms in the bilingual segments of the Heroes corpus, I embark on building a system that can learn and generate prosodic structures in a neural machine translation setting. This section explains the framework built in order to carry out experiments to answer the questions we listed earlier. Before diving in the technical specifications of the system built, I will list the requirements defined prior to the implementations:

1. Translation will be in movie domain. This is mainly because of our motivation for gaining insights for the automatic subtitling and dubbing use cases.

2. System will be extended incrementally i.e. we will start from a basic text translation system and then add on it first prosodic input and then prosodic output.

3. Prosodic encoding and decoding will be built within the translation systeml; i.e., text and prosodic encoding and decoding parameters will be learned jointly.

4. The system should be able to compensate for the scarcity of spoken parallel data.

In order to address these requirements, a system is built that can learn translation of textual and prosodic features. I will refer to this system as *TransProse* for simplicity. Design and subtleties of this model are explained in the next subsection 5.3.1. Next, data sources that suit best for our problem has to be selected. Collected and acquired corpora and our preprocessing steps are detailed in subsection 5.3.2.

5.3.1. Neural Translation Model

Sequence-to-sequence modeling has proved in recent years that it is one of the most advanced models for modelling automatic translation. This model works by *encoding* a phrase in the source language to a single vector and then *decoding* it into the phrase in the target language. A drawback of this classic encoder-decoder architecture is that the entire encoded sequence has to go through one vector acting as a funnel to be decoded into target sequence. This problem was addressed by the introduction of attention mechanism (Bahdanau et al., 2014; Luong et al., 2015), which lets the decoded sequence to focus on relevant areas of the encoded sequence.

TransProse framework is based on a sequence-to-sequence network with attention mechanism, which was explained earlier in Chapter 2. For that reason, I will not go deep into the core of the architecture but I will explain more how it was extended to handle prosodic input and output.

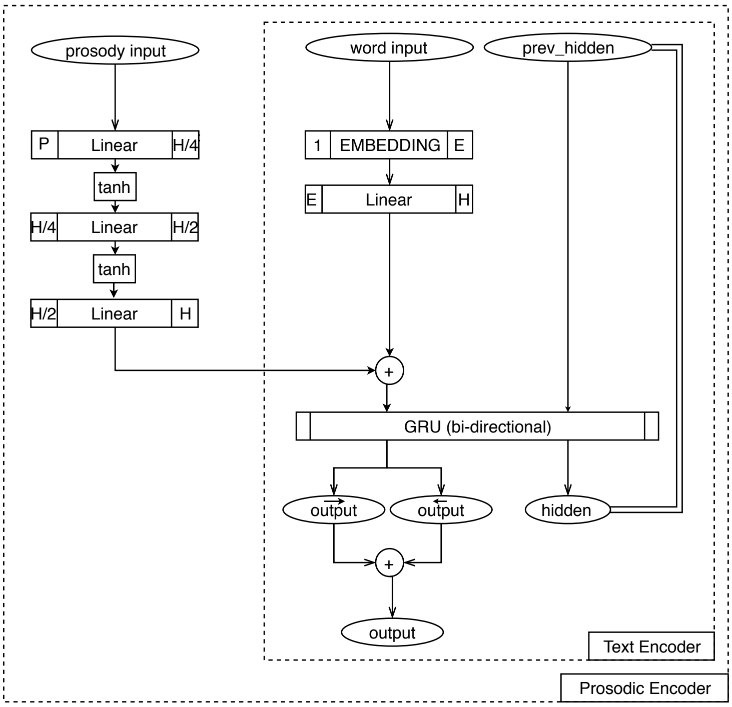


Figure 5.6: TransProse sequence-to-sequence translation encoder with prosodic input.

Encoding text tokens and prosody

The encoder of the system is illustrated in Figure 5.6. The text encoder part (inner box) takes word token indexes as input and passes them through an embedded layer then a linear layer to obtain word vectors of size H. Then, this vector is passed to a bidirectional GRU layer, outputting hidden and an output vectors in both directions at each step. The forward and backward output vectors are then summed in order to obtain an output of size H for each input token.

Encoding jointly with the added prosodic features is depicted in the outer box of the same figure. Note that prosody input vector carries any number of prosodic/acoustic features that belong to the word token at that timestep. This number is denoted with P. A separate encoding sequence is followed by the prosodic features. The input features are converted to a vector of size H in a gradual fashion where a linear layer is followed by a non-linearity at each step. Once it is the same size of the GRU input layer, it is summed with the encoded word input and introduced to the bidirectional layer together with the input word token representation. Output vectors at each timestep are then passed on through the decoder.

Decoding text tokens and prosody

As illustrated in Figure 5.7, the decoder is also designed to output either text tokens only or accompanied with their corresponding prosodic features. During training, target sequence tokens are input and passed through first, the embedding layer, then a linear layer followed by a dropout layer until it reaches the GRU layer. The output of the GRU layer is used to determine the attention weights according to each of the effect of the encoder output effect on that particular target token. The attention model is based on the global attention model in Luong et al. (2015). The weights vector for output at timestep t is calculated as in Equation 5.1, where h_t stands for GRU output in decoder side and h_s on the target side. General scoring function is used as the scoring function (Equation 5.2). A general overview of the implementation of the neural attention architecture is illustrated in Figure 5.8.

$$a_t(s) = \operatorname{align}\left(h_t, \overline{h}_s\right) = \frac{\exp\left(\operatorname{score}\left(h_t, \overline{h}_s\right)\right)}{\sum_s \exp\left(\operatorname{score}\left(h_t, \overline{h}_s\right)\right)} \tag{5.1}$$

$$\operatorname{score}\left(h_t, \overline{h}_s\right) = h_t^\top \mathbf{W}_a \overline{h}_s \tag{5.2}$$

After the attention weights are calculated, encoder outputs are multiplied with these weights and averaged to obtain the context vector. Context vector is then

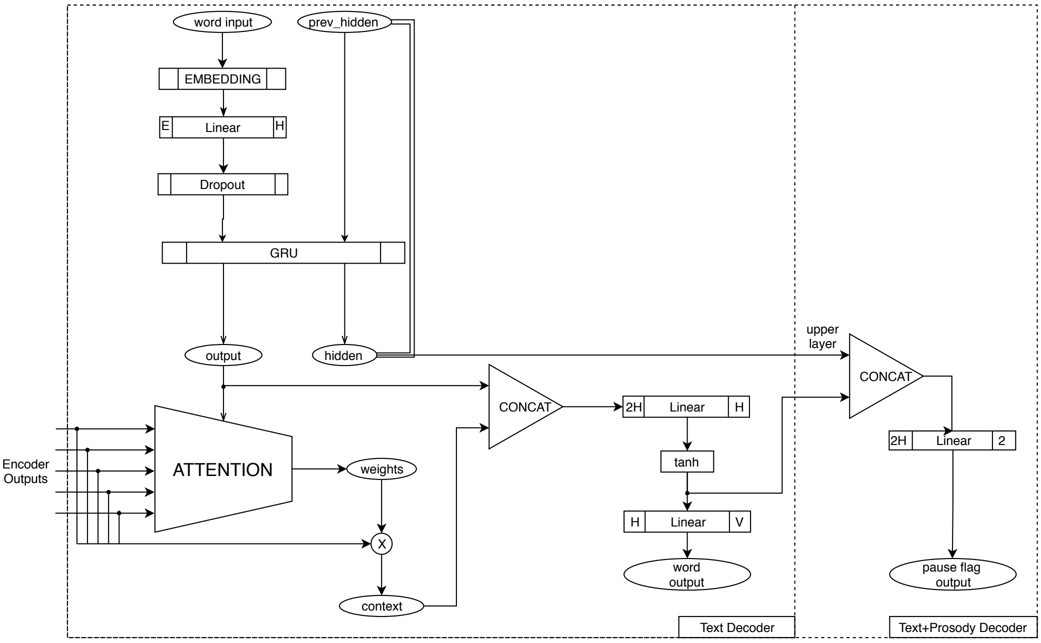


Figure 5.7: TransProse sequence-to-sequence translation decoder with prosodic output.

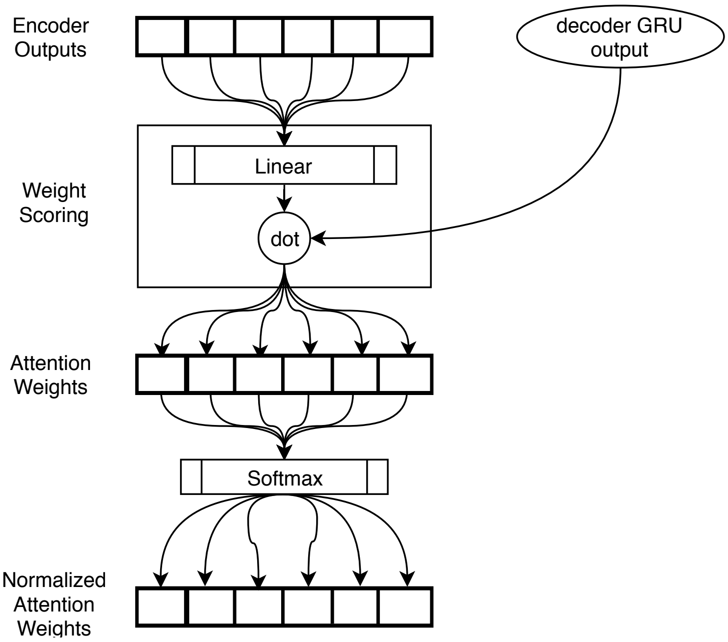


Figure 5.8: Attention mechanism in the TransProse decoder.

concatenated with the decoder GRU output and eventually used to calculate the vocabulary sized one-hot word token output.

The depiction of the decoder illustrated in Figure 5.7 has two types of prosodic outputs: flag and pause value output. Both are predicted with respect to the word output and hidden vector of the GRU layer. Flag outputs are of size 2 and real valued outputs are of size 1.

The two types of outputs are placed for illustration purposes. The model can be configured to output any number of flag or value type features.

Learning procedure

A translation model enhanced with prosodic input and/or output is obtained in two stages. First, training is performed on parallel text data updating only the parameters belonging to the text encoder and decoder. On a second stage, training is performed on prosodically annotated parallel data with the joint text+prosody encoder/decoder components. Before starting the second stage training, the extended text+prosodic architecture is initialized with the pre-trained first stage parameters. While training on prosodic data, all parameters are updated, where prosodic model components are trained from scratch.

In order to calculate gradients for the model to converge while training, loss functions has to be defined for the model outputs. The loss function compares the prediction of the model to the gold output and back-propagates to decide how the model parameters should be updated. For text token and flag-based outputs, masked cross entropy is used. Whereas for real valued outputs mean square error is employed. The total loss of the system is calculated after each batch by summing each individual loss with its respective weight, as in Equation 5.3:

$$L_{total} = W_{word} \cdot L_{word} + W_{pauseflag} \cdot L_{pauseflag} + W_{pausevalue} \cdot L_{pausevalue} \quad (5.3)$$

In text training, total loss function is only the loss coming from word token predictions. In audio training, W_{word} is set to 1.0 and prosodic feature losses ($W_{pausevalue}$ and $W_{pauseflag}$) are set to 10.0. In this work, only pause flag output is employed in the experiment reported in Section 5.6. For parameter optimization, Adam algorithm (Kingma and Ba (2014)) is used. After each training epoch, model is validated on a smaller validation set. Training is continued until no improvement is noted in terms of total loss in the validation set in the last three epochs.

5.3.2. Data and data preprocessing

Training is performed in two stages with two types of data, a parallel text corpus and a prosodically annotated parallel spoken audio corpus. *OpenSubtitles corpus* and *Heroes corpus* were used respectively for the two stages of the task.

Parallel text dataset

In order to keep consistent in the movie domain, text data are also obtained from movie based resources. *OpenSubtitles* collection[4] provides parallel text obtained from movie and series subtitles (Lison and Tiedemann (2016)). The *OpenSubtitles2018* release[5] contains 1,782 bilingual text pairs among 62 languages. For the English-Spanish pair more than 61 million sentence pairs are openly provided.

The text dataset to train TransProse models is gathered from this set. The dataset size was restricted to 5 million sentence pairs to accommodate training in reasonable amount of time. Sentence pairs for this set of 5 million sentence pairs, which we call the *opus5mm* set, is obtained by a simple set of filters selecting from the original corpus. These filters are:

1. Sentences shouldn't contain more than a certain number of tokens (40 in this case),
2. Sentences shouldn't contain any non-alphanumeric characters,
3. Sentence should only consist of tokens in a pre-determined vocabulary of most frequent 30,000 tokens in the whole corpus.

These filters were determined in order to ensure a training set as clean as possible. Since the corpus is derived automatically from subtitles registered in *opensubtitles.org*, it is likely to come across badly written sentences or misalignments. Also, subtitle segments where auditory or visual annotations are made had to be filtered out. These subtitle segments contain information on speaker, background music, voice characteristics and even signatures of the subtitle authors and are marked with usage of XML-style tags or other non-alphanumeric characters.

Another important characteristic of the movie subtitles is that translations are not necessarily literal. The differences are caused by the nature of subtitling, e.g. sentences are cut short to fit on the screen or some spoken remarks are omitted

[4] http://www.opensubtitles.org/

[5] http://opus.nlpl.eu/OpenSubtitles2018.php

to simplify reading. This feature makes movie subtitles sub-optimal for training translation models.

The *opus5mm* dataset consists of 5 million sentence pairs plus 10,000 pairs for validation and 10,000 for testing purposes. For tokenization, NLTK tokenizer is used with a modification on English enclitics[6].

Parallel speech dataset

For the second stage training involving prosodic parameters, *Heroes corpus* is used. This corpus is described with detail in Chapter 3. The experiments described in this chapter are performed on a pre-release version of the corpus that consisted of 7225 prosodically annotated English-Spanish parallel segments. Two training-test-validation partitionings generated from this dataset are described in Table 5.3. The first partitioning *heroes-v1* is generated by taking 80% of the shuffled segment pairs as training set and dividing the rest into two to be used as test and validation sets. The second partitioning *heroes-v2* is generated in a more manual fashion. First, 138 segment pairs were manually picked from *heroes-v1* test set, that ensured a translation well enough to be used in the prosodic prediction experiments. Secondly, after shuffling the rest of the segment pairs, 200 were chosen randomly for the validation set and the remaining 6887 segments were allocated as training set.

Dataset	**#Training samples**	**#Validation samples**	**#Testing samples**
heroes-v1	6141	542	541
heroes-v2	6887	200	138

Table 5.3: Heroes corpus partitioning versions and number of train, validation and testing set samples.

Punctuation handling and prosodic sequence representation

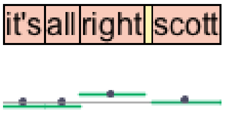
In the neural machine translation model described above, prosodic features are assumed to be parallel to the tokens that form the input and output sequences. That is, each token has its prosodic features associated to it. This goes parallel with how the movie segment data is stored in the speech corpus used. In *Heroes corpus*,

[6] Words were separated from apostrophes. For example the word “I’ll” consists of two tokens: “I” and “’ll”

prosodic features are calculated and stored for every word. These features include pause coming after that word (*pause after*, *mean f0* and *mean intensity*) associated with that word. Punctuation marks are also features of the words. Two features are associated with each word: *punctuation before* and *punctuation after*. These structures needed to be taken into account during the design of the translation pipeline.

In TransProse framework, word and punctuation tokens are considered as sequence tokens and prosodic features are associated to each of these tokens. Resulting from this design choice, punctuation tokens also need to carry prosodic features. Although this is logically unintuitive, it was the approximation that was made. Figure 5.11 shows an example of an utterance from the speech corpus and its representation as an input sequence to the neural network. The original segment consists of 4 tokens as can be seen in the *Prosograph* illustration. In order to convert it into a sequence, each word is tokenized and punctuation features of each word are placed as tokens. The input sequence results in 7 tokens (including the END token). F0 and intensity features are copied into the punctuation mark tokens belonging to the word. Pause after feature is kept only at the last word token.

Speech segment



Sequence representation

word token	it	's	all	right	,	scott	.	<END>
pause after	0.0	0.0	0.0	0.03	0.0	0.0	0.0	0.0
f0 mean	-1.29	-1.29	-1.29	2.13	2.13	0.25	0.25	0.0
intensity mean	0.02	0.02	0.02	2.34	2.34	0.83	0.83	0.0

Figure 5.9: Segment s3_1_0001_EN from the Heroes corpus and its sequence representation.

Prosodic parameters were normalized within -1 to +1 for f0 and intensity features, and 0 to +1 for pause features. F0 and intensity range is taken from the minimum and maximum values in the database. Pause range is taken as 0 to 10 seconds.

Further implementation details

The architecture described in this section is implemented with PyTorch[7]. Text models were trained on graphical processing units (GPU) and second stage prosodic models were trained on CPU. Other hyperparameters used while training are listed in Table 5.4.

Hyperparameter	**Value**
Encoder learning rate	0.0001
Decoder learning rate	0.0005
Batch size	64
Hidden layer size	512
Number of GRU hidden layers	2
Decoder dropout rate	0.1
Gradient norm clip rate	50.0
Vocabulary size (EN-ES)	30,000

Table 5.4: Hyperparameters used in the experiments with TransProse architecture.

Word vectors were created using the *gensim* library (Řehůřek and Sojka (2010)). These vectors were trained from English and Spanish segments in the complete OpenSubtitles corpus. An optimal vocabulary was created with a combination of the most frequent 30,000 tokens from this corpus with the tokens in the Heroes corpus. Also, tokens that are needed for the network to function were added as special tokens. These tokens are: switch token (SOS), end token (END), unknown token (UNK) and empty token (EMP).

5.4. How does prosodic punctuation restoration affect translation?

In previous chapter, it was stated that punctuation restoration of transcriptions has an important role for subsequent processing steps such as machine translation. This section focuses on this very statement and explores the effect of punctuation restoration in transcripts on translation. Principal functionality of punctuation in a machine translation system is that it segments source input into meaningful units through sentence structure, which in turn gives cues on the output structure. Most state-of-the-art translation systems take sentences as units to translate. The type of punctuation that ends the sentence signifies if it is a statement, interrogation or

[7] https://pytorch.org/

exclamation. This does not only affect what punctuation mark should be placed at the end of the target sentence but also the translation itself. Moreover, intra-sentence segmentations through usage of commas signal which types of word groupings (e.g. clauses) should be carried to the target translation.

The first question that this section explores is: To what extend source input punctuation affects machine translation performance? Secondly, assuming to have unpunctuated transcriptions of an audiovisual content, e.g. coming from ASR, how can we recover from this loss with punctuation restoration as a preliminary process step to machine translation? Thirdly and mainly, the effect of using prosody and domain-adapted punctuation and translation models is explored.

Experimental Setup

The experiments are based on the movie domain focusing on the use case of translation of TV series. Translation models were trained on OPUS subtitle corpus and then adapted to the Heroes corpus. Test set consists of 542 parallel segments (*heroes-v1* set).

In order to quantify the difference in performance caused by punctuation, simply the source sentences are sent to translation with and without the punctuation marks already present in the dataset. These marks are the annotated punctuation in the original English subtitles of the TV series. Segments with the original punctuation are called *subtitle punctuation* and *no punctuation* with punctuation marks removed.

Punctuation restorations are performed over English segments with models obtained using the *punkProse* framework presented in Chapter 4. Four models that were trained specifically for this experiment are listed in Table 5.5. As the Heroes corpus is not sufficiently big to train a punctuation model, all models are principally trained on the TED corpus[8]. Adapted models were obtained through training over English training segments of the Heroes corpus dataset (version 1). Two types of feature sets are used for training the models: 1. Lexical-only where words are the only features, 2. Lexical-prosodic where words and two prosodic features are used (pause and mean-f0).

Similarly for text translation models, a model that was first trained over the OPUS dataset (5 million subtitle segment pairs) was then adapted to the TV series by re-training on 6142 segment pairs from the Heroes corpus. Translation direction is English to Spanish. Two types of models were created with respect to source language punctuation. The standard model was trained with punctuation

[8] Presented in Chapter 3

Punctuation model	**Base training dataset**	**Adaptation dataset**	**Features**
ted-w	TED Corpus	-	word
ted-wpmf	TED Corpus	-	word, pause, mean-f0
tedheroes-w	TED Corpus	Heroes corpus	word
tedheroes-wpmf	TED Corpus	Heroes corpus	word, pause, mean-f0

Table 5.5: Punctuation restoration models used for punctuating raw English segments.

presence in both source and target language segments (model $p \rightarrow p$). A side model was created by removing punctuation in English segments but keeping in Target (model $u \rightarrow p$). This model was created to test if a translation model is able to guess the punctuation on the target side even though it is not present in the source language, emulating ASR output.

Results

Table 5.6 shows the translation performance of various settings in this experiment. The baseline, which translates from manually punctuated English transcriptions, gives a BLEU score of 20.15%. A significant fall of almost 8% in BLEU is observed when the punctuation marks are removed from the translation input when the same translation model is used. Although, through using the translation model that was trained on unpunctuated input, this fall is largely recovered (17.44% BLEU).

The rest of the rows on Table 5.6 are results from translation of English segments with recovered punctuation. BLEU scores obtained with 4 source input types, each one resulting from using a different punctuation model, are reported. It can be seen that BLEU scores improve generally compared to the unpunctuated input. However, punctuation models trained from a different domain does not seem to reach the performance of the translation model that predicts from unpunctuated input. This threshold is only surpassed by the restored input that is adapted to the dataset and uses prosodic features as input (18.08% BLEU).

It has to be taken note that the restoration models only predict period (.), comma (,) and question mark (?). Other punctuation marks such as colon (:) and quotation marks (") have an important role in defining the meaning thus needs to be included during translation if general domain translation is considered. However, in movie/series domain these punctuation marks are seldom used.

Punctuation in source phrase	**Punctuation model**	**Translation model**	**BLEU (%)**
subtitle (baseline)	-	$p \to p$	20.15
none	-	$p \to p$	12.17
none	-	$u \to p$	17.44
restored	*ted-w*	$p \to p$	16.73
restored	*ted-wpmf*	$p \to p$	17.22
restored	*tedheroes-w*	$p \to p$	16.94
restored	*tedheroes-wpmf*	$p \to p$	18.08

Table 5.6: BLEU scores obtained from translating English subtitle segments with restored punctuation.

5.5. Does pause encoding improve translation?

In the previous section I reported the improvement in translation quality through punctuation restoration on the input sentence to the system. Results showed that using prosodic modelling on the punctuation restoration process benefits translation quality in terms of BLEU scores. In this section, I further explore the introduction of prosodic features directly on the translation system and its eventual effect on text translation quality. I particularly focus on the inclusion of intra-word pausings as an additional feature on the encoder side of the sequence-to-sequence translation architecture.

Motivation for this question comes from the observations made from the dubbed scripts of the Heroes corpus which was presented in Section 5.2. It has been observed that many times pausing in the English segments were reflected in the Spanish translations in terms of phrasing. These examples suggest that pauses residing in the input sentence might be a feature that needs to be taken in an automated translation setting.

Experimental Setup

Two models trained with the TransProse framework is compared. The baseline model is the one presented in the previous section. This model encodes and decodes only word tokens. The prosodic model is trained with pauses as the only prosodic feature on the encoding side while only word tokens are outputted on the decoder side. Both models are trained on punctuated input sentences. On a real-world setting the input sentences would lack punctuation since they would be the output of an ASR system. However, in this case, having an access to a punctuation restoration system remedies this deficit. Punctuation marks are kept in input sentences for two main reasons: (1) for their effect on translation quality (as proved in previous section, and (2) for their high correlation with pauses in speech.

As training and testing sets, *heroes-v1* dataset is used. Two versions of the test set are created: one with the original subtitle punctuation annotations and another one with recovered punctuation using the prosodic punctuation recovery model *tedheroes-wpmf* presented in previous section. Two versions of the testing sets are identical in terms of the word tokens but show differences in punctuation due to the errors made during recovery.

Results

		translation encoder type	
		text	text+pauses
2* punctuation in input	subtitle	20.15	21.46
	recovered	18.08	19.15

Table 5.7: BLEU scores on the *heroes-v1* testing set with and without pause encoding.

Table 5.7 lists the BLEU scores obtained by the baseline and the prosodically enhanced encoder models on the two testing sets. With perfect punctuation assumed on the input sentences, there is an improvement of 1.31% in terms of BLEU scoring. With punctuation recovery preprocessing on the raw transcripts, translation quality still increases by a 1.07%. These improvements prove the hypothesis that prosodic encoding can help improve quality of neural machine translation.

5.6. Can we transfer pausing in speech-to-speech translation?

In previous experiments, I dealt with the input of prosodic features –mainly pause– to a translation system in order to improve the quality of the output text. This section further expands on this framework and explores outputting of prosodic features as well in order to be used as cues in synthesis applications. The motivation for this task is to approach more e.g. the process of automatic spoken translation where the aim is to synthesize the translation with a transfer of prosodic features from source to target language.

The particular task I define in this section is the transfer of pauses. Previously on Section 5.2, I have given some examples of direct and indirect transfer of pauses in the dubbed movie segments of the Heroes corpus. It has been shown that in majority of the times a pause in the English segment is reflected in the dubbed Spanish segments. I delve into the question of whether its possible to incorporate the modelling of transferring of pauses in a neural machine translation framework.

Experimental Setup

In order to carry out this task, TransProse framework is set to input and output pause features. As explained in Section 5.3.1, the encoder-decoder architecture accepts prosodic input for each input word token. It can also be set to output binary or real-numbered features for each output word token. In this experiment, each input word token to the encoder is accompanied with the duration of the pause coming after that word token. On the decoder side, for every output word token a binary flag is outputted determining presence of a pausing coming after that word token. To keep the model simple, duration of the pauses are not predicted.

Heroes-v2 set is chosen as dataset for its selection of testing samples that consists of hand-selected simpler translations. In this particular setting, the translation quality is an important factor in terms of evaluation. If the text translation is not above a certain quality threshold, it is hard to determine whether it is right or wrong where the model predicts a pause at a certain point.

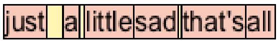
Prosodic translation models were trained by adaptation to the models trained on the *opus5mm* set on the *heroes-v2* training set consisting of 6887 segments.

Results

The task of predicting labels for each predicted word poses a particular challenge in terms of evaluation. The reason is that the predicted text translations are generally different than the gold standard translations. If the word with a pause after in the gold standard is not present in the predicted translation, then there is no way to evaluate the pause prediction performance. Also, as the data are not created in laboratory conditions pausing in the input language segments are not necessarily reflected in the target language segments in 100% of the cases. For these reasons, we carried out manual inspection on the relatively small test set to see how much the model predicts meaningful pauses that reflect the pauses in the input sentences.

On manual inspection, it is seen that in a minority of the cases input pauses were reflected on the predicted prosodic translations. Out of 138 segments in the testing set, 64 of them have a pause event of more than 0.05 seconds in the input English segments. Out of prosodic translations of this 64 segments, in only 16 of them a pause flag is output (25%). Also, in 9 segment translations a pausing is predicted even though there is none in the original input sentence. Statistically it can be said that the model performs poorly in reflecting pauses in translations. Even though a small portion of the pauses in input sentences are reflected in the prosodic translations, it is seen in some of the translations that model is able to convey the input pausing correctly to the translation. See Figures 5.10 and 5.11 for some of the examples that can be deemed as successful prosodic translations.

Input segment

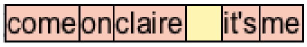


Prosodic translation

Es un poco [P] triste, eso es todo.

Figure 5.10: Prosodic translation of segment s3_16_0113 from the Heroes corpus.

Input segment



Prosodic translation

venga, claire. [P] soy yo.

Figure 5.11: Prosodic translation of segment s3_1_0001 from the Heroes corpus.

Examples like these show that the model does learn how to predict pauses in translations to some extend. However, the size of the training set shows to be too small to obtain useful generalizations for this problem.

Perception tests with text-to-speech synthesis

A perception test was prepared to test to what extend the pausing cues outputted by the prosodic translation actually help. This test involved participants listening to a batch of original segments from the Heroes corpus and then listening to two types of synthesized translations (dubbings): (1) synthesis of the "classical" text translation output, and (2) synthesis of the prosodic translation together with the prosodic cues. Two comparisons were made for each sample: (1)

Which one of the Spanish dubbings is a better translation of the original English segment? and (2) Which one of the dubbings better reflects the prosody of the original speech?

Selecting the samples It was challenging to select the samples to be used in such a test for two reasons: firstly, the quality of translations was, in general, considerably low. If a fair comparison has to be made, both outputs of the text translation model and the prosodic translation model had to be with an acceptable quality. This was assured by manually picking samples from the testing set which had acceptable translations for both models. Secondly, if a prosodic comparison was to be made, there had to be a prosodic cue output on the prosodic model. As reported earlier, only one quarter of the prosodic translations of the testing set actually had a pausing output given that there was a pausing in the source English sentence. 15 sentences were selected respecting these requirements.

Synthesizing the translations IBM Watson TTS demo[9] is provided free of charge online for demo usage with one male, one female Spanish speaker. Also, it is possible to add prosodic cues to the synthesis using SSML tags[10]. In this case, it was only needed to add breaks after the words with a pause after on the prosodic translations. Lengths for the breaks were selected regarding the average lengths of breaks in the Spanish segments (presented in Section 5.2).

Results 32 people participated in the test. The results of the perception test show that in 76.5% of the cases the translation made by the prosodic model was preferred. However, in prosodic assessment, synthesized samples with the prosodic cues were preferred in only 27% of the cases. In 32.4% of the cases, participants stated that they heard no difference between the synthesized samples in terms of prosody. The majority 40.6% preferred synthesized version of the text translation.

The lack of agreement on synthesized samples can be explained by two reasons: firstly, both synthesized samples were greatly far from the original segments from the series. First hearing the original segment from the TV series and then presented with an automated output, many participants found the dubbings highly "robotic" compared to acted speech samples. Second reason is that in many samples the added pauses contributed even more to the unnaturalness of the synthesized samples. This shows that the pauses cannot be taken in isolation from other prosodic cues. Appearance and duration of the pauses are directly affected by the speech rate. In turn, a pause between two words affects the general intonation of the sentence. If the pause for example is placed for emphasis, the emphasized word should be marked with a high pitch or intensity as well. Placement of a

[9] https://www.ibm.com/watson/services/text-to-speech/

[10] Previously explained in Chapter 2

pause without taking account the general prosodic structure does not contribute in terms of expressivity and even might harm it in terms of naturalness.

5.7. Conclusion

In this chapter, I have discussed about the reasons and ways to include prosodic features in a spoken language translation pipeline. I formed the basis of our study and experiments through the use-case of automatic translation and dubbing of media material such as movies or TV shows. Experiments were performed using a proposed prosodically enhanced translation system and a parallel corpus compiled from the original and dubbed spoken segments of a recent TV show (Heroes corpus).

The empirical study performed on the segments of the Heroes corpus indicate that pausings have an effect both on the translation and dubbings made by professionals. In majority of cases both original and dubbed speech segments agree on containing an intra-word pause. I argued that if an automated system was to be built to translate and dub spoken segments in a TV show, it has to heed certain prosodic characteristics of the actors' speeches just as dubbing artists do. I further demonstrated how the classic speech-to-speech translation pipeline would fail to do a proper translation when prosody of the source sentence is ignored.

Motivated by the shortcomings of this classic translation pipeline, a new framework has been introduced that takes prosodic features into account and outputs prosodic cues for the synthesis of the translated segments. This framework, which I call *TransProse*, is designed to take speech transcriptions together with their word-level prosodic features and output translations with word-level prosodic cues. Joint prosodic-textual translation models were trained in two stages, where in first stage translation of word tokens is learned from a large corpus from movie domain and later transfer of prosodic features are learned on a second stage using the prosodically annotated Heroes corpus.

My experiments involving the incorporation of prosody to the movie-domain translation pipeline were built around these three questions: (1) How does prosodic punctuation restoration affect translation?, (2) Does pause encoding improve translation? and (3) Can we transfer pausing in spoken translation? Through these three questions I have employed prosody into the TransProse translation pipeline in three steps. In the first step prosody is incorporated on the standard text-to-text translation setting by punctuating the source sentences using prosodic cues. In the second step, intra-word pausings as the sole prosodic features is introduced on the encoder side to improve the translations. And finally on the third step, I introduced both prosodic input and output where the output tokens were accompanied with flags that signal if a pause should be placed after that token or not.

As my initial study suggested, improvements over the translation quality were achieved with incorporation of prosody onto the input side of the translation framework. Related to my first question, I reported an improvement over usage of prosodic features in a preliminary punctuation restoration step. This was demonstrated with punctuating the input phrases to the translation system first with a lexical-based model and then with a prosodically enhanced model. While the lexically-based model recorded an improvement in terms of 4.77% BLEU over the unpunctuated input, with incorporation of prosody this improvement further rose to 5.91%. This showed the importance of prosody in a translation pipeline even as a feature of a preliminary step such as punctuation recovery.

For answering my second question, I have incorporated intra-word pauses as a prosodic feature on the input side of the translation and assessed the translation quality. Comparing with standard text translation an improvement of 1.31% BLEU was achieved. To demonstrate this increase in a setting closer to a real speech-to-text translation setting, punctuation marks in the input phrases were removed (which would be missing in ASR output) and recovered again using the prosodic punctuation models and a similar improvement has been recorded (1.07%). The results clearly show the usefulness of including prosodic features in a spoken translation pipeline. The results I demonstrated were performed over one prosodic feature, intra-word pauses. Further prosodic features should be assessed and incorporated in future research.

The final experiment presented in this section was the most experimental as it delved into the area of spoken output. I have demonstrated that through the proposed framework it is possible to obtain some meaningful output to be used as cues in a text-to-speech system. However, even when assessed on a clean testing set, the output given by the system failed to satisfy in terms of achieving transfer of prosodic features. Also, the perception tests I performed did not show any improvement in terms of reflection of prosodic features of the source phrases. Even though a state-of-the-art TTS system has been employed, synthesized translations hint that pauses cannot be taken as an isolated feature to achieve prosodic transfer in speech-to-speech translation. In order to achieve that, many aspects should be considered as a whole such as transfer of spectral characteristics, speech rate, intonation, etc.

All in all, the proposed methodology paves the way for research for inclusion prosody in both speech-to-text and speech-to-speech translation pipelines. Even with a simple model and a limited sized audio data it is possible to achieve improvements on the spoken language translation domain through incorporation of prosody.

Chapter 6

CONCLUSIONS AND FUTURE WORK

In this dissertation, I have addressed the motivation for the inclusion of prosodic modelling in systems with spoken input. Initially, I have tried to keep a general view on the issue while explaining my motivation for this work. As it is not only the linguistic content but also the para-linguistic content that counts in human communication, machines should also pay attention to it in processing natural languages. By machines, I mean the constantly increasing computational methods that help humans in many tasks that involve understanding our own ways of communication. This intelligence can already be seen in interactive voice based assistants, automatic transcription or translation systems. Moreover, its growth promises that it will be ubiquitous in near future to have systems understanding and interacting as humans do.

The trend in the development of these machines is that focus is given into the extraction of the linguistic content when human speech input is being processed. Prosody, which has an important linguistic and para-linguistic role in communication is generally not given the attention it deserves. Recent work that showed great progress in speech recognition and machine translation through the rise of deep neural network based methods, focus merely on the recognition and transfer of the linguistic content. There is relatively a smaller number of studies exploring approaches including para-linguistic content that speech prosody carries.

Defined within this broad perspective, I have argued and demonstrated my point specifically on two main applications that processes spoken language: automatic speech transcription and spoken language translation. In the former one, I focused my attention on the effect prosody has on the punctuation of the resulting transcription. In the latter one, I experimented on the inclusion of pause features on both input and output of a sequence-to-sequence translation pipeline where movie domain data was employed. To accommodate both of these data-driven processes, a number of toolkits were developed for the creation, processing, handling and visualization of speech data annotated with acoustic/prosodic features.

This chapter is organized as follows: I will give final conclusions regarding the developments and experiments conducted within the framework of this dissertation in Section 6.1. Next, I will sketch a roadmap for future work in Section 6.2. Finally, the contributed artifacts that are the results of the work in this dissertation is given in three sections: publications in Section 6.3, datasets in Section 6.4 and open source software resources in Section 6.5.

6.1. Conclusions

The motivation to enhance speech related methodologies with prosodic modellings comes with a cost and that mostly resides in the field of data harvesting. It is notable the portion of the work given in this dissertation only involved in development of methodologies to collect naturally expressive speech data and shape them to be processed in machine learning based applications. A complete pipeline for collecting, handling, annotation, storage and visualization of prosodic data has been developed. *Proscript* library, which accommodates the process of acoustic/prosodic annotation of transcripted speech data, served a basis for many of the experimentation presented. Regarding collection of large datasets, *Prosograph* was developed as a spin-off framework which served greatly in manual examination of this type of data. It was seen that there was a lot to learn from the data itself during the design of machine-learning based systems that process prosody. Nature of prosody shows that it is worth its own set of toolkit for examination and studying. Prosograph addressed this with its easily programmable interface that helped visualization and interaction of speech related characteristics in huge chunks of data.

I furthermore addressed the lack of available naturally expressive parallel spoken data to the scientific community by presentation of a novel methodology for extracting parallel corpora from dubbed movies. Although this methodology comes with its own challenges depending on the source of data, it shows the valuable resource for parallel expressive speech that lies in field of dubbing. Using this methodology, an English-Spanish parallel speech dataset was gathered, annotated with acoustic-prosodic features and published openly for the speech research community. This is to my knowledge first example of a corpus containing a rich variety of prosodic characteristics and is bilingually structured.

A part of this dissertation focused on the topic of punctuation as it was seen as the closest form of symbology in written text that is influenced by prosody. Output of speech recognition interfaces lack this form of symbology. Punctuation manifests intonational patterns that are influenced by modalities (question, affirmation, exclamation etc.) as the type of the punctuation mark at the end of a sentence, or breaks and phrasing manifest as commas. These symbols prove to be essential for both humans and machines during the reading of the transcript. Although these theme is explored and proven that prosodic features do improve accuracy of the punctuation marks generated for raw transcripts, there was no past work that studied the effect of various features to the detection of different types of punctuation marks.

An application oriented approach was followed by taking a state-of-the-art RNN-based punctuation recovery system as basis and expanding it to handle more types of prosodic features that were available through a corpus based on conference talks. Experiments that were conducted on purely spoken data showed that combination of the features word, POS, intra-word pause and pitch worked best for accurate detection of punctuation marks in general. The proposed architecture that processes lexical and prosodic features in parallel and trains on one pass achieved a significant improvement compared to the baseline approach when only spoken data was used. An F-score of 83%, 71.8% and 55.2% was obtained with various feature combinations in accurate restoration of periods, question marks and commas. The methodology proved to give promising results also when employed within a framework where an automatic speech recognition (ASR) system was employed.

Effect of prosody in punctuation restoration modelling was tested on two subsequent processes: dependency parsing and machine translation. The former evaluation was conducted on a small dataset and showed that accounting of prosody does imply an improvement in correct paring of the sentences. This shows that commas which are detected more accurately with prosodic models does improve parsing despite its low F-score. Within a spoken language machine translation framework, punctuation restoration that was applied as a preprocessing step before machine translation showed better results with the inclusion of prosodic modelling.

A spoken language machine translation framework was built in order to experiment with prosodic input and output. While previous work on speech-to-speech translation focused on transfer of intonational patterns and emphasis, the framework introduced in this dissertation was based on involvement of raw prosodic features into a sequence-to-sequence MT setup. Basing on the movie domain, my motivation was to build an automatic dubbing pipeline that took prosodic features into the effect. Prosodic modelling through punctuation and as pause features on the input side showed improvement with respect to translation quality. Although some meaningful output has been recorded in decoding pause features, results showed to be poor in transferring pause features. I presume this can be overcame with better quality text data and a larger parallel audio corpus.

6.2. Future Work

prosograph ile gelismeler movie corpus ustune

6.2.1. Punctuation restoration for ASR

Catalan TV broadcast **?** commas are poorly detected. linguistic modelling needed

6.2.2. Machine Translation enhanced with Prosody

prosody embeddings

6.3. List of Publications

- Öktem, A., Farrús, M., Bonafonte, A. *Bilingual Prosodic Dataset Compilation for Spoken Language Translation.* Proc. IberSPEECH 2018, October 25-29 2018, Barcelona, Spain.

- Külebi, B., Öktem, A., *Building an Open Source Automatic Speech Recognition System for Catalan*. Proc. IberSPEECH 2018, October 25-29 2018, Barcelona, Spain.

- Öktem A, Farrús M, Bonafonte A., *Visualizing Punctuation Restoration in Speech Transcripts with Prosograph*. Proc. Interspeech 2018, p. 1493-4, Sep 2-6 2018, Hyderabad, India.

- Öktem A, Farrús M, Wanner L. *Punctuating Transcribed Speech Using Lexical and Prosodic Cues via Attentional Parallel RNNs.* (Under review) Computer Speech and Language. Elsevier.

- Öktem A, Farrús M, Wanner L., *Attentional Parallel RNNs for Generating Punctuation in Transcribed Speech*. In: Camelin N, Estève Y, Martín-Vide

C. Statistical Language and Speech Processing. 5th International Conference SLSP 2017; 2017 Oct 23-25; Le Mans, France. Cham: Springer, 2017. p. 131-42. (LNCS; no. 10583). DOI: 10.1007/978-3-319-68456-7_11

- Burga A, Öktem A, Wanner L., *Revising the METU-Sabancı Turkish treebank: an Exercise in Surface-syntactic Annotation of Agglutinative Languages.* In: Montemagni S, Nivre J, editors. Proceedings of the Fourth International Conference on Dependency Linguistics (Depling 2017); 2017 Sept 18-20; Pisa, Italy. ACL; 2017. p. 32-41.

- Öktem A, Farrús M, Wanner L., *Prosograph: A Tool for Prosody Visualisation of Large Speech Corpora.* In: Proceedings of the 18th Annual Conference of the International Speech Communication Association (INTERSPEECH 2017),p. 809-10, Stockholm, Sweden: ISCA; 2017.

- Öktem A, Farrús M, Wanner L., *Automatic Extraction of Parallel Speech Corpora from Dubbed Movies.* In Proceedings of the 10th Workshop on Building and Using Comparable Corpora (BUCC), p. 31-35, Vancouver, Canada: ACL, 2017.

6.4. List of Datasets

- **TED Talks Corpus** - TED talks are a set of conference talks lasting in average 15 minutes each that have been held worldwide in more than 100 languages. They include a large variety of topics, from technology and design to science, culture and academia. The corpus consists of 1038 talks by 877 English speakers, uttering a total amount of 155174 sentences. The corresponding transcripts, as well as audio and video files, are available on TED's website[1]. LINK: http://hdl.handle.net/10230/33981

- **Heroes Corpus** - Heroes corpus contains mapped bilingual (English and Spanish) speech segments from the TV series Heroes. It contains 7000 single speaker speech segments extracted from the original and Spanish dubbed version of 21 episodes. Audio segments are accompanied with subtitle transcriptions and word-level prosodic/paralinguistic information. LINK: http://hdl.handle.net/10230/35572

[1] http://www.ted.com

6.5. List of Software Resources

- **movie2parallelDB** - Automatic parallel speech database extractor from dubbed movies. LINK: https://github.com/TalnUPF/movie2parallelDB

- **Prosograph** - A Visualizer for prosodically annotated speech corpora written with Processing. LINK: https://github.com/TalnUPF/Prosograph

- **PunkProse** - A library for punctuation generation for speech transcripts using lexical and prosodic features. LINK: https://github.com/TalnUPF/punkProse

- **Proscript** - A Python package to help create proscript files. Proscript helps represent speech with annotated prosody. The library carries automatic annotation scripts that are based on Praat. LINK: https://github.com/alpoktem/proscript

- **TED corpus preprocessing scripts** - A library for creating a trainable corpus from the prosodically annotated TED corpus prepared by Mireia Farrús and Catherine Lai. LINK: https://github.com/alpoktem/ted_preprocess

- **Prosodic punctuation generation demo on ASR** - This is a demo software that contains scripts to punctuate audio recordings using punkProse library. It is intended to use for demonstration purposes. LINK: https://github.com/alpoktem/punkProse_ASR-demo

- **TransProse** - A framework based on sequence-to-sequence neural networks for translation with prosodic features. LINK: https://github.com/alpoktem/TransProse

Bibliography

Agüero, P. D., Adell, J., and Bonafonte, A. (2006). Prosody generation for speech-to-speech translation. In International Conference on Acoustics, Speech, and Signal Processing (ICASSP), volume 1, pages 557–560. IEEE.

Agüero, P. D., Tulli, J. C., and Bonafonte, A. (2018). Pause transfer in the speech-to-speech translation domain. pages 87–90.

Algeo, J. (1987). A comprehensive grammar of the english language. by randolph quirk, sidney greenbaum, geoffrey leech, and jan svartvik. london: Longman. 1985. x + 1779. Journal of English Linguistics, 20(1):122–136.

Almeman, K., Lee, M., and Almiman, A. A. (2013). Multi dialect Arabic speech parallel corpora. In 1st International Conference on Communications, Signal Processing, and their Applications (ICCSPA), pages 1–6. IEEE.

Anumanchipalli, G. K., Oliveira, L. C., and Black, A. W. (2012). Intent transfer in speech-to-speech machine translation. In Spoken Language Technology (SLT) Workshop, pages 153–158. IEEE.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. CoRR, abs/1409.0473.

Ballesteros, M. and Wanner, L. (2016). A neural network architecture for multilingual punctuation generation. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing.

Baron, D., Shriberg, E., and Stolcke, A. (2002). Automatic punctuation and disfluency detection in multi-party meetings using prosodic and lexical cues. Channels, 20(61):41.

Batista, F., Moniz, H., Trancoso, I., and Mamede, N. (2012). Bilingual experiments on automatic recovery of capitalization and punctuation of automatic speech transcripts. IEEE Transactions on Audio, Speech, and Language Processing, 20(2):474–485.

Bendazzoli, C. and Sandrelli, A. (2005). An approach to corpus-based interpreting studies: developing EPIC (European Parliament Interpreting Corpus). In MuTra 2005–Challenges of Multidimensional Translation, pages 1–12.

Bird, S., Klein, E., and Loper, E. (2009). Natural Language Processing with Python. O'Reilly Media Inc.

Boersma, P. (2001). Praat, a system for doing phonetics by computer. Glot International, 5(9/10):341–345.

Bohnet, B. and Kuhn, J. (2012). The best of both worlds: A graph-based completion model for transition-based parsers. In Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, EACL '12, pages 77–87, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bohnet, B. and Nivre, J. (2012). A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL '12, pages 1455–1465, Stroudsburg, PA, USA. Association for Computational Linguistics.

Buchholz, S. and Marsi, E. (2006). Conll-x shared task on multilingual dependency parsing. In Proceedings of the Tenth Conference on Computational Natural Language Learning, CoNLL-X '06, pages 149–164, Stroudsburg, PA, USA. Association for Computational Linguistics.

Cettolo, M., Girardi, C., and Federico, M. (2012). Wit^3: Web inventory of transcribed and translated talks. In Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT), pages 261–268, Trento, Italy.

Chafe, W. (1988). Punctuation and the prosody of written language. Written Communication, 5(4):395–426.

Che, X., Wang, C., Yang, H., and Meinel, C. (2016). Punctuation prediction for unsegmented transcript based on word vector. In LREC.

Cho, E., Niehues, J., and Waibel, A. H. (2017). Nmt-based segmentation and punctuation insertion for real-time spoken language translation. In Proceedings of Interspeech 2017.

Cho, K., van Merrienboer, B., Gülçehre, Ç., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. CoRR, abs/1406.1078.

Christensen, H., Gotoh, Y., and Renals, S. (2001). Punctuation annotation using statistical prosody models. In in Proc. ISCA Workshop on Prosody in Speech Recognition and Understanding, pages 35–40.

Dahl, G. E., Yu, D., Deng, L., and Acero, A. (2012). Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. IEEE Transactions on Audio, Speech, and Language Processing, 20(1):30–42.

Do, Q. T., Sakti, S., and Nakamura, S. (2017a). Toward expressive speech translation: A unified sequence-to-sequence lstms approach for translating words and emphasis. In Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017, pages 2640–2644.

Do, Q. T., Sakti, S., and Nakamura, S. (2018). Sequence-to-sequence models for emphasis speech translation. IEEE/ACM Trans. Audio, Speech & Language Processing, 26(10):1873–1883.

Do, Q. T., Sakti, S., Neubig, G., and Nakamura, S. (2016). Transferring emphasis in speech translation using hard-attentional neural network models. In Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016, pages 2533–2537.

Do, Q. T., Sakti, S., Neubig, G., Toda, T., and Nakamura, S. (2015). Improving translation of emphasis with pause prediction in speech-to-speech translation systems. In IWSLT.

Do, Q. T., Toda, T., Neubig, G., Sakti, S., and Nakamura, S. (2017b). Preserving word-level emphasis in speech-to-speech translation. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 25(3):544–556.

Domínguez, M., Farrús, M., and Wanner, L. (2016). An automatic prosody tagger for spontaneous speech. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 377–386. The COLING 2016 Organizing Committee.

Domínguez, M., Latorre, I., Farrús, M., Codina-Filba, J., and Wanner, L. (2016). Praat on the web: An upgrade of praat for semi-automatic speech annotation. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations, pages 218–222. The COLING 2016 Organizing Committee.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res., 12:2121–2159.

Dyer, C., Ballesteros, M., Ling, W., Matthews, A., and Smith, N. A. (2015). Transition-based dependency parsing with stack long short-term memory. CoRR, abs/1505.08075.

Farrús, M., Lai, C., and Moore, J. D. (2016). Paragraph-based Prosodic Cues for Speech Synthesis Applications. In Proceedings of the 8th International Conference on Speech Prosody.

Favre, B., Grishman, R., Hillard, D., Ji, H., Hakkani-Tur, D., and Ostendorf, M. (2008). Punctuating speech for information extraction. In Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on, pages 5013–5016. IEEE.

Federmann, C. and Lewis, W. D. (2016). Microsoft Speech Language Translation (MSLT) corpus: The IWSLT 2016 release for English, French and German. In International Workshop on Spoken Language Translation.

Fung, J. G., Hakkani-Tür, D., Magimai-Doss, M., Shriberg, E., Cuendet, S., and Mirghafori, N. (2007). Cross-linguistic analysis of prosodic features for sentence segmentation. In Eighth Annual Conference of the International Speech Communication Association.

Gao, Y., Zhou, B., Gu, L., Sarikaya, R., kwang Kuo, H., Rosti, A. . I., Afify, M., and Zhu, W. (2006). Ibm mastor: Multilingual automatic speech-to-speech translator. In 2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings, volume 5, pages V–V.

Garner, P. N., Clark, R., Goldman, J.-P., Honnet, P.-E., Ivanova, M., Lazaridis, A., Liang, H., Pfister, B., Ribeiro, M. S., Wehrli, E., and Yamagishi, J. (2014). Translation and prosody in swiss languages.

Garrido, J.-M., Codina, M., and Fodge, K. (2018). TransDic, a public domain tool for the generation of phonetic dictionaries in standard and dialectal Spanish and Catalan. In Proc. IberSPEECH 2018, pages 291–295.

Goldberg, Y. (2016). A primer on neural network models for natural language processing. J. Artif. Int. Res., 57(1):345–420.

Goldman, J.-P., Honnet, P.-E., Clark, R., Garner, P. N., Ivanova, M., Lazaridis, A., Liang, H., Macedo, T., Pfister, B., Ribeiro, M. S., Wehrli, E., and Yamagishi, J. (2016). The siwis database: a multilingual speech database with acted emphasis. Idiap-RR Idiap-RR-13-2016, Idiap.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). Deep Learning. The MIT Press.

Gravano, A., Jansche, M., and Bacchiani, M. (2009). Restoring punctuation and capitalization in transcribed speech. In IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pages 4741–4744. doi:10.1109/ICASSP.2009.4960690.

Green, N. (2011). Dependency parsing. page 137–142.

Guo, P., Huang, H., Jian, P., and Guo, Y. (2016). Prosodic annotation enriched statistical machine translation. In 2016 10th International Symposium on Chinese Spoken Language Processing (ISCSLP), pages 1–5.

Hermann, K. M. and Blunsom, P. (2014). Multilingual Models for Compositional Distributional Semantics. In Proceedings of ACL.

Hernandez, F., Nguyen, V., Ghannay, S., Tomashenko, N. A., and Estève, Y. (2018). TED-LIUM 3: twice as much data and corpus repartition for experiments on speaker adaptation. CoRR, abs/1805.04699.

Hillard, D., Huang, Z., Ji, H., Grishman, R., Hakkani-Tur, D., Harper, M., Ostendorf, M., and Wang, W. (2006). Impact of automatic comma prediction on pos/name tagging of speech. In Spoken Language Technology Workshop, 2006. IEEE, pages 58–61. IEEE.

Huang, X., Acero, A., and Hon, H.-W. (2001). Spoken Language Processing: A Guide to Theory, Algorithm, and System Development. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition.

Jakubıcek, M. and Horák, A. (2010). Punctuation detection with full syntactic parsing. Special issue: Natural Language Processing and its Applications, page 335.

Jones, B. E. M. (1994). Exploring the role of punctuation in parsing natural text. In Proceedings of the 15th Conference on Computational Linguistics - Volume 1, COLING '94, pages 421–425, Stroudsburg, PA, USA. Association for Computational Linguistics.

Katagiri, S. (2000). Handbook of Neural Networks for Speech Processing. Artech House signal processing library. Artech House.

Khomitsevich, O., Chistikov, P., Krivosheeva, T., Epimakhova, N., and Chernykh, I. (2015). Combining prosodic and lexical classifiers for two-pass punctuation detection in a Russian ASR system. In International Conference on Speech and Computer, pages 161–169. Springer.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. CoRR, abs/1412.6980.

Klejch, O., Bell, P., and Renals, S. (2017). Sequence-to-sequence models for punctuated transcription combining lexical and acoustic features. In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 5700–5704.

Kolář, J. and Lamel, L. (2012). Development and evaluation of automatic punctuation for french and english speech-to-text. In Proceedings of Interspeech, pages 1376–1379.

Kolář, J., Švec, J., and Psutka, J. (2004). Automatic punctuation annotation in Czech broadcast news speech. In in Proceedings of the SPECOM.

Kurimo, M., Byrne, W., Dines, J., Garner, P. N., Gibson, M., Guan, Y., Hirsimäki, T., Karhila, R., King, S., Liang, H., Oura, K., Saheer, L., Shannon, M., Shiota, S., Tian, J., Tokuda, K., Wester, M., Wu, Y.-J., and Yamagishi, J. (2010). Personalising speech-to-speech translation in the emime project. In Proceedings of the ACL 2010 System Demonstrations, ACLDemos '10, pages 48–53, Stroudsburg, PA, USA. Association for Computational Linguistics.

LeCun, Y., Bottou, L., Orr, G. B., and Müller, K.-R. (1998). Efficient backprop. In Neural Networks: Tricks of the Trade, This Book is an Outgrowth of a 1996 NIPS Workshop, pages 9–50, London, UK, UK. Springer-Verlag.

Levy, T., Silber-Varod, V., and Moyal, A. (2012). The effect of pitch, intensity and pause duration in punctuation detection. In Electrical & Electronics Engineers in Israel (IEEEI), 2012 IEEE 27th Convention of, pages 1–4. IEEE.

Lison, P. and Tiedemann, J. (2016). Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles. In LREC.

Liu, Y., Chawla, N. V., Harper, M. P., Shriberg, E., and Stolcke, A. (2006). A study in machine learning from imbalanced data for sentence boundary detection in speech. Computer Speech & Language, 20(4):468–494.

Lowth, R. (1762). A short introduction to English grammar. Scolar P Menston (Yorks.).

Lu, W. and Ng, H. T. (2010). Better punctuation prediction with dynamic conditional random fields. In Proceedings of the 2010 conference on empirical methods in natural language processing, pages 177–186. Association for Computational Linguistics.

Luong, M., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. CoRR, abs/1508.04025.

Ma, J., Zhang, Y., and Zhu, J. (2014). Punctuation processing for projective dependency parsing. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 791–796. Association for Computational Linguistics.

Matusov, E., Hillard, D., Magimai-doss, M., Hakkani-tur, D., Ostendorf, M., and Ney, H. (2007). Improving speech translation with automatic boundary prediction. In In Proc. Interspeech, pages 2449–2452.

Matusov, E., Mauser, A., and Ney, H. (2006). Automatic sentence segmentation and punctuation prediction for spoken language translation. In International Workshop on Spoken Language Translation (IWSLT) 2006.

McAuliffe, M., Socolof, M., Mihuc, S., Wagner, M., and Sonderegger, M. (2017). Montreal Forced Aligner: Trainable text-speech alignment using Kaldi. In Proc. Interspeech, pages 498–502.

Melvin, R. S., May, W., Narayanan, S. S., Georgiou, P. G., and Ganjavi, S. (2004). Creation of a doctor-patient dialogue corpus using standardized patients. In 4th International Conference on Language Resources and Evaluation (LREC).

Mertens, P. (2004). The prosogram: Semi-automatic transcription of prosody based on a tonal perception model. In Proceedings of the Speech Prosody, Nara, Japan.

Moore, N. (2016). What's the point? the role of punctuation in realising information structure in written english. Functional Linguistics, 3(1):6.

Nivre, J. and Fang, C.-T. (2017). Universal dependency evaluation. In Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017), page 86–95, Gothenburg, Sweden.

Noth, E., Batliner, A., Kiessling, A., Kompe, R., and Niemann, H. (2000). 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.

Nunberg, G. (1990). The Linguistics of Punctuation.

Öktem, A., Farrús, M., and Wanner, L. (2017). Automatic extraction of parallel speech corpora from dubbed movies. In Proceedings of the 10th Workshop on Building and Using Comparable Corpora (BUCC), pages 31–35, Vancouver, Canada.

Paulik, M., Rao, S., Lane, I., Vogel, S., and Schultz, T. (2008). Sentence segmentation and punctuation recovery for spoken language translation. In 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, pages 5105–5108.

Peitz, S., Freitag, M., Mauser, A., and Ney, H. (2011). Modeling punctuation prediction as machine translation. In International Workshop on Spoken Language Translation (IWSLT) 2011.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE, 77(2):257–286.

Rangarajan Sridhar, V. K., Bangalore, S., and Narayanan, S. (2013). Enriching machine-mediated speech-to-speech translation using contextual information. Comput. Speech Lang., 27(2):492–508.

Řehůřek, R. and Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, pages 45–50, Valletta, Malta. ELRA. http://is.muni.cz/publication/884893/en.

Rosenberg, A. (2010). AutoBI - A tool for automatic ToBI annotation. In Proceedings of Interspeech, pages 146–149, Makuhari, Japan.

Rosenberg, A. (2018). Speech, prosody, and machines: Nine challenges for prosody research.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, pages 65–386.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1988). Neurocomputing: Foundations of research. chapter Learning Representations by Backpropagating Errors, pages 696–699. MIT Press, Cambridge, MA, USA.

Schlangen, D. and Hough, J. (2017). Joint, incremental disfluency detection and utterance segmentation from speech. In EACL.

Schulz, H., Ruiz, M., and Fonollosa, J. A. R. (2008). TECNOPARLA - Speech technologies for Catalan and its application to speech-to-speech translation. Procesamiento del lenguaje natural, 41:319–320.

Schuster, M., Paliwal, K. K., and General, A. (1997). Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing.

Sečujski, M., Gerazov, B., Csapó, T. G., Delić, V., Garner, P. N., Gjoreski, A., Guennec, D., Ivanovski, Z., Melov, A., Németh, G., Stojkovic, A., and Szaszák, G. (2016). Design of a speech corpus for research on cross-lingual prosody transfer. In International Conference on Speech and Computer, pages 199–206. Springer.

Silverman, K., Beckman, M., Pitrelli, J., Ostendorf, M., Wightman, C., Price, P., Pierrehumbert, J., and Hirschberg, J. (1992). TOBI: A Standard for Labeling English Prosody. In 2nd International Conference on Spoken Language Processing (ICSLP 92), pages 867–870, Banff, Canada.

Spitkovsky, V. I., Alshawi, H., and Jurafsky, D. (2011). Punctuation: Making a point in unsupervised dependency parsing. In Proceedings of the Fifteenth Conference on Computational Natural Language Learning, CoNLL '11, pages 19–28, Stroudsburg, PA, USA. Association for Computational Linguistics.

Szaszak, G., Gabor Csapo, T., Garner, P. N., Gerazov, B., Ivanovski, Z., Nemeth, G., Toth, B., Secujski, M., and Delic, V. (2014). The sp2 scopes project on speech prosody.

Theano Development Team (2016). Theano: A Python framework for fast computation of mathematical expressions. arXiv e-prints, abs/1605.02688.

Tilk, O. and Alumäe, T. (2015). LSTM for punctuation restoration in speech transcripts. In Proceedings of Interspeech, pages 683–687.

Tilk, O. and Alumäe, T. (2016). Bidirectional recurrent neural network with attention mechanism for punctuation restoration. In Proceedings of Interspeech, pages 3047–3051.

Treviso, M. V., Shulby, C. D., and Aluísio, S. M. (2017). Evaluating word embeddings for sentence boundary detection in speech transcripts. CoRR, abs/1708.04704.

Truong, D. Q., Neubig, G., Sakti, S., Toda, T., and Nakamura, S. (2014). Collection and analysis of a japanese-english emphasized speech corpora. In 2014 17th Oriental Chapter of the International Committee for the Co-ordination and Standardization of Speech Databases and Assessment Techniques (COCOSDA), pages 1–5.

Tsiartas, A., Ghosh, P., Georgiou, P. G., and Narayanan, S. (2011). Bilingual audio-subtitle extraction using automatic segmentation of movie audio. In International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 5624–5627. IEEE.

Tündik, M. [U+FFFD], Szaszák, G., Gosztolya, G., and Beke, A. (2018). User-centric evaluation of automatic punctuation in asr closed captioning. In Proceedings of Interspeech 2018, pages 2628–2632, Hyderabad, India.

Ueffing, N., Bisani, M., and Vozila, P. (2013). Improved models for automatic punctuation prediction for spoken and written text. In INTERSPEECH, pages 3097–3101.

Vandeghinste, V., Verwimp, L., Pelemans, J., and Wambacq, P. (2018). A comparison of different punctuation prediction approaches in a translation context.

Wahlster, W. (2013). VerbMobil: Foundations of speech-to-speech translation. Springer Science & Business Media.

Wang, T. and Cho, K. (2015). Larger-context language modelling. CoRR, abs/1511.03729.

Wester, M. (2010). The EMIME Bilingual Database. Technical report, The University of Edinburgh.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. CoRR, abs/1609.08144.

Xu, C., Xie, L., and Xiao, X. (2017). A bidirectional lstm approach with word embeddings for sentence boundary detection. Journal of Signal Processing Systems.

Xu, Y. (2013). ProsodyPro — A tool for large-scale systematic prosody analysis. In Proceedings of Tools and Resources for the Analysis of Speech Proso, pages 7–10, Aix-en-Provence, France.

Öktem, A., Farrús, M., and Bonafonte, A. (2018a). Visualizing punctuation restoration in speech transcripts with prosograph. In Interspeech 2018, Hyderabad, India.

Öktem, A., Farrús, M., and Wanner, L. (2017a). Attentional parallel rnns for generating punctuation in transcribed speech. In 5th International Conference on Statistical Language and Speech Processing SLSP 2017, Le Mans, France.

Öktem, A., Farrús, M., and Wanner, L. (2017b). Prosograph: a tool for prosody visualisation of large speech corpora. In Proceedings of the 18th Annual Conference of the International Speech Communication Association (INTERSPEECH), Stockholm, Sweden.

Öktem, A., Farrús, M., and Bonafonte, A. (2018b). Bilingual prosodic dataset compilation for spoken language translation. In Proc. IberSPEECH 2018, pages 20–24.