공학석사 학위논문

Automatic phrase break and sentence stress prediction in English with RNN

RNN 기반의 휴지부 및 문장 강세 자동 예측

August 2018

Graduate School of

Seoul National University

Interdisciplinary Program in Cognitive Science

Evgenia Nedelko

Abstract

Automatic phrase break and sentence stress prediction in English with RNN

Evgenia Nedelko
Interdisciplinary Program in Cognitive Science
The Graduate School
Seoul National University

Prosody has been widely researched and studied from different angles in recent years. In linguistics, **prosody** is concerned with those elements of speech that are not individual phonetic segments (vowels and consonants) but are properties of syllables and larger units of speech. These contribute to linguistic functions such as intonation, tone, stress, and rhythm. Prosody may reflect various features of the speaker or the utterance: the emotional state of the speaker; the form of the utterance (statement, question, or command); emphasis, contrast, and focus; or other elements of language that may not be encoded by grammar or by vocabulary.

In the present research we concentrated on such prosodic phenomena like phrase breaks and sentence stress. In English, pausing is more likely before a word carrying a high information content.

Defining pause is not easy. First of all, pausing is a very natural phenomenon related to breathing. Also it seems necessary to distinguish between silent pauses and "filled" pauses where a hesitation is perceived but the speaker continues to emit sound.

Prosodic stress, or sentence stress, refers to stress patterns that apply at a higher level than the individual word – namely within a prosodic unit. It may involve a certain natural stress pattern characteristic of a given language, but may also involve the placing of emphasis on particular words because of their relative importance.

The goal of the present work was to conduct a multi-class prediction task consisting of three classes: NB standing for not pause occurring, B standing for a minor pause occurring in the sentence and BB standing for a pause marking a sentence boundary.

The second goal was to conduct a sentence stress prediction task and demonstrate that the implementation of neural network models without additionally extracted features will allow to achieve a relatively high performance. Both prediction tasks were performed based on the word embedding model together with neural network architectures.

The main hypothesis was that the implementation of bidirectional neural networks will help increase the accuracy of pause prediction and drastically improve the overall performance. The second hypothesis was that a pre-trained word embedding model in combination with a neural network architecture will allow to achieve good performance on sentence stress prediction task.

The third and the last goal was to examine and compare the performance of different neural network models on the prediction tasks mentioned above.

Keywords: Prosody prediction, Phrase breaks, Sentence stress, Deep learning, Neural networks.

Student Number: 2015 - 23282

CONTENTS

List of Tables

List of Figures

Chapter 1

Introduction

1.1. Motivation & General Description

Prosody has been widely researched and studied from different angles in recent years. For example, Calhoun (2010) looked into the connection between informativeness and prosodic prominence studying how the latter influences informativeness in speech. Ferreira (1993) presents a model of prosodic production that describes the process of prosodic encoding and provides a quantitative specification of the relation between word lengthening and pausing.

People pause between words and sentences when they speak to emphasize content, or to make an utterance more understandable, or just to take a breath. The process of inserting prosodic breaks in an utterance is called **phrasing**. Ferreira (2007) suggests that some intonational phrases are planned top-down and generated from a discourse-semantic representation, but others are created on the fly, emerging when a speaker finds him- or herself needing to close off one processing chunk so that a new one can be initiated. Intonational phrases are then a byproduct of processing decisions.

Phrasing breaks long utterances into meaningful units of information and makes the speech more understandable. Thus, it is a crucial for a second language learner to be able to make appropriate use

of this prosodic cue so that his speech sounded intelligible and comprehensible. More importantly, when building a CALL system, similar to speech synthesis, phrase breaks should be treated as the first step for other models of prosody, such as intonation prediction and duration modeling, i.e. to learn intonation of a foreign language and place sentence stress appropriately, a learner first should understand and learn where to make pauses so that his speech sounded natural and native-like.

In English stress is cued not only by how we use the features of pitch, duration, loudness and vowel quality in the stressed syllables, but also by how these features are used, in a reduced manner, in background or unstressed syllables. It has been demonstrated that stressed syllables are detected and processed faster than unstressed ones (Cutler and Foss, 1977). Moreover, Cutler (1976) and Cutler and Fodor (1979) also found that reaction times are shorter when an acoustically identical phoneme is predicted to be stressed based on various aspects of the preceding context. Stressed syllables are thought to carry higher informational content than unstressed ones (Altman and Carter, 1989), and stress patterns appear to be strongly related to the information structure of a sentence (Calhoun, 2010).

Sentence stress emphasizes the portion of the utterance that is more important for the speaker or that the speaker wants the listener to concentrate on. Stress in a sentence has no fixed distribution. It is related to semantics. The function of stress in sentences is to highlight the information - bearing words in the utterance (Sole, 1991).

This thesis presents a neural network approach to phrase break and sentence stress prediction in English, that in future can be implemented in a CALL system or speech synthesis model.

1.2. Purpose of Research

The purpose of the study is threefold. The first goal of the present work was to conduct a multi-class prediction task consisting of three classes: NB standing for not pause occurring, B standing for a minor pause occurring in the sentence and BB standing for a pause marking a sentence boundary.

The second goal was to conduct a sentence stress prediction task and demonstrate that the implementation of neural network models without additionally extracted features will allow to achieve a relatively high performance. Both prediction tasks were performed based on the word embedding model together with neural network architectures.

The main hypothesis was that the implementation of bidirectional neural networks will help increase the accuracy of pause prediction and drastically improve the overall performance. The second hypothesis was that a pre-trained word embedding model in combination with a neural network architecture will allow to achieve good performance on sentence stress prediction task.

The third and the last goal was to examine and compare the performance of different neural network models on the prediction tasks mentioned above.

This research may further provide a background and become a starting point for building a CALL system that helps non-native speakers sound more native-like working on their intonation.

1.3. Scope of Research

The scope of the present research is limited to British English and the peculiarities of its prosodic features, as British English is considered to be an example for imitation for non-native learners of English. Moreover, as our training corpus consists of short pieces of news, we limit ourselves to the prediction of prosodic phenomena only in the genre of news.

The present study also investigates only neural models not looking into different approaches to the prosody prediction which are in more detail discussed in Chapter 2.

1.4. Structure of the Thesis

The present thesis is divided into 4 Chapters. In Chapter 1 we are giving the general outline of the present work. Chapter 2 contains a brief summary of related works as well as some theoretical background. We also briefly introduce neural networks and explain how they work. In Chapter 3 we explain the experimental setup and show the results of the conducted experiments. Chapter 4 contains general discussions and the conclusion.

Chapter 2

Overview and related works

The present chapter will look into how prosodic phenomena like pauses and sentence stress are explained in the linguistic theory and also provide a brief overview of related experimental works.

2.1. Prosody: linguistic theory

In linguistics, **prosody** is concerned with properties of syllables and larger units of speech, however, it does not handle individual phonetic segments (i.e. vowels and consonants) (Jurafsky, 2000). These contribute to linguistic functions such as intonation, tone, stress, and rhythm. Prosody may contain information about various properties of the speaker or the utterance itself: the emotional state of the orator; the form of the utterance (statement, question, or command); the presence of irony or sarcasm; emphasis, contrast, and focus; or other properties of spoken language that may not be reflected by grammar or by the choice of vocabulary.

Prosody can be divided into two main components: a metrical component and an intonational component (Selkirk, 1984; Warren, 1999; Zubizarreta, 1998). Metrical phonology is about sentence stress and duration – the sound features that cause an utterance to have a

distinct rhythm (Hayes, 1995; Liberman et al., 1977; Selkirk, 1984). Hayes (1995) states that the central claim of the metrical stress theory is that stress is the linguistic manifestation of rhythmic structure, and that the special phonological properties of stress can be explicated in this basis.

2.1.1. Pauses and impact on intonation

Phrasing can be defined as the mechanism that speakers turn to in order to break their speech into meaningful chunks of information. Although pausing is a natural and physiologically necessary phenomenon related to breathing and limited supply of air, it is claimed that pauses may also carry some contrastive linguistic information. Silverman et al. (1992) suggests that phrase breaks can be classified into multiple levels one of the most prominent and frequent levels being a pause. Goldman-Eisler (1961) suggests that pauses are an external reflection of certain cognitive activities involved in speech production. This activity could be on the part of the speaker (structuring some idea or a thought before delivering a clear message), or on part of the listener (giving listeners time to comprehend and assimilate what was just spoken).

Prahallad et al. (2010) have shown that prosodic phrase breaks are specific to a speaker. Parlikar et al. (2013) took a step further and studied how phrasing pattern can vary depending on the style examining such styles like parliament proceedings, radio broadcast, president Barack Obama public talks and audiobooks.

A question that has interested linguists for some time is related to the correlation between the syntactic phrases (ie. the syntax tree) and prosodic phrasing. According to Taylor (2009), factors which complicate the relationship between syntactic and prosodic phrases include:

1. Prosodic phrasing seems "flatter" in that while syntactic phrasing is inherently recursive and can exhibit a large degree of nesting, if levels in prosodic phrasing exist there are only a few, and there is no sense in which the phrases within an utterance seem to embed so strongly.
2. Prosodic phrasing is to some extent governed by purely phonological, phonetic or acoustic factors which can override the syntax. Chomsky (1968) commented on the fact that in

THIS IS THE DOG THAT WORRIED THE CAT THAT KILLED THE RAT THAT ATE THE MALT THAT LAY IN THE HOUSE THAT JACK BUILT

nearly every speaker says this with a flat prosodic phrasing like:

THIS IS THE DOG | THAT WORRIED THE CAT | THAT KILLED THE RAT | THAT ATE THE MALT | THAT LAY IN THE HOUSE | THAT JACK BUILT

whereas the syntactic phrasing is deeply recursive and embedded

(THIS IS (THE DOG (THAT WORRIED THE CAT (THAT KILLED THE RAT (THAT ATE THE MALT (THAT LAY IN THE HOUSE (THAT JACK BUILT)))))))

The speech patterns of the utterance seem to override the syntactic patterns.

3. Phrasing is particularly prone to speaker choice, and while "rules" exist for where phrase breaks may or may not occur, quite often it appears optional whether a phrase break will be found in a particular utterance.

So as we can see the relationship between syntactic and prosodic structure is complicated and a general theory which links syntax and prosody is yet to be developed. Some have even argued that the difficulty really lies with our models of syntax, and that if we developed syntactic models which took more account of the ways sentences were spoken some of these problems might be resolved (Taylor, 2009).

The ToBI standard introduced by Silverman et al. (1992) specifies two levels of prosodic phrase boundaries both of which correspond to two levels of perceived disjuncture. The ToBI standard uses a phrase hierarchy where each intonational phrase is composed of one or more intermediate phrases, each of which contains at least one accented word. Break indices in ToBi serve to indicate how strong the break between words is (Silverman, 1992):

- 0 = clitic boundary
- 1 = normal word boundary
- 2 = perceived juncture with mo intonation effect
- 3 = intermediate phrase
- 4 = full intonation phrase

For the purposes of this research, we narrow our prediction task to three classes: NB standing for not pause occurring, B standing for a minor pause occurring in the sentence and BB standing for a pause marking a sentence boundary.

2.1.2. Sentence stress and impact on intonation

In English, every word has one or more lexical stresses1 depending on the structure of the word and the number of syllables, but not all word stresses are phonetically realized in utterance. Content words, which deliver major semantic information and therefore require listeners' attention, normally receive stress on the utterance level whereas function words do not (Lee Gary Geunbae et al., 2017).

Prosodic stress, or sentence stress, refers to stress patterns that apply at a higher level than the individual word – namely within a prosodic unit. It may involve a certain natural stress pattern characteristic of a given language, but may also involve the placing of emphasis on particular words because of their relative importance.

The major function of sentence stress is to highlight semantically important words and to form the rhythmic pattern of the utterance. It has been known that sentence stress occurs at regular intervals and unstressed syllables between consecutive stressed syllables are reduced, causing the impression of 'stress-timed rhythm' (Lee Gary Geunbae et al., 2017).

The following pattern has been claimed for English: the traditional distinction between (lexical) primary and secondary stress is replaced partly by a prosodic rule stating that the final stressed syllable in a phrase is given additional stress. (A word spoken alone becomes such a phrase, hence such prosodic stress may appear to be lexical if the pronunciation of words is analyzed in a standalone context rather than within phrases.)

Prosodic stress is also often used pragmatically to emphasize (focus attention on) particular words or the ideas associated with them. Doing this can change or clarify the meaning of a sentence.

In English, stress is most dramatically realized on focused or accented words. The main stress within a sentence, often found on the last stressed word, is called the *nuclear stress*.

In English, stress is cued not only by how we use the features of pitch, duration, loudness and vowel quality in the stressed syllables, but also by how these features are used, in a reduced manner, in background or unstressed syllables (Sole, 1991).

Sentence stress emphasizes the portion of the utterance that is more important for the speaker or that the speaker wants the listener to concentrate on. Stress in a sentence has no fixed distribution. It is related to semantics. The function of stress in sentences is to **highlight the information** - bearing words in the utterance. The general rule – content words are stressed, grammatical words unstressed – applies to «normal» default stressing. It does not apply when contrastive or emphatic meaning is intended. In fact, any word or syllable might be

stressed (in fact, bear the intonational nucleus) when used contrastively or emphatically.

The relative prominence of a syllable within a sentence has been attributed to three different factors (Selkirk, 1995):

1) The presence or absence of a pitch accent on the syllable (the accent factor)

2) The position of the syllable within a constituent structure (the phrasing factor)

3) The presence or absence of other prominent syllables in the immediate vicinity of the syllable (the rhythm factor)

2.2. Prosody prediction

Prosody prediction is the task of generating the prosodic form from the text.

Generally, prosody prediction is divided into three different prediction tasks that are tackled separately:

- Phrasing prediction
- Prominence prediction
- Intonational tune prediction

2.2.1. Computational Approaches to Prosody Prediction

As far as the phrasing prediction is concerned, all approaches can be roughly divided into deterministic, classifier, HMM (Hidden-Markov

model) and neural network approaches.

Deterministic approaches are:

1. deterministic punctuation (DP): Place a phrase break at every punctuation mark

2. deterministic content, function (DCF): Place a phrase break every time a function word follows a content word.

These can be combined into a single algorithm, deterministic content function punctuation (DCFP), which places a break after every content word that precedes a function word and also at punctuation (Taylor, 2009).

A number of more sophisticated deterministic systems have been proposed which make of rules for specific cases. For example, the verb balancing rule of Bachenko and Fitzpatrick (1990) works through a sentence left to right and compares the number of words in a potential phrase formed with the verb and the syntactic constituents to the left, and the number of words in the constituent to the right. The potential phrase with the shortest number of words is chosen as the correct one.

As for classifier approaches, Wang and Hirschberg (1992) introduced the idea of using decision trees for phrase break prediction. Decision trees allow a wide variety of heterogeneous features, examples of which are given below (Taylor, 2009):

• total seconds in the utterance

• total words in the utterance

• speaking rate

• time from start of sentence to current word

• time to end of sentence from current word

• is the word before the juncture accented?

• is the word after the juncture accented?

• Part-of-speech (POS)

• syntactic category of constituent immediately dominating the word before the juncture

• syntactic category of constituent immediately dominating the word after the juncture

Neural network approaches will be examined in more details in the next subsection.

2.2.2. Phrase break prediction task

Phrasing is crucial to speech synthesis which is why it became the center to many research projects. For example, Watts O. et al (2011) used usupervised continuous-valued word features for phrase break prediction without a part-of-speech tagger and applied CART as the classification method for the task getting F-measure of 77.7 on breaks. The proposed method showed a little less good perfomance compared to the top line system with full POS that achieved F-measure of 79.8.

In the other paper, Watts et al. (2013) used neural net word representations achieving with the proposed unsupervised model F-score of 78.35.

Vadapalli et al. (2014) proposed a neural network dictionary learning architecture to induce task specific word representations, i.e.,

to derive word representations specific to phrase break prediction, and achieved F-score of 0.72 on the task. Proposed architecture uses a **multilayer perceptron (MLP)** setup as a discriminative classifier.

In the other paper, Vadapalli et al. (2016) built a recurrent neural network architecture used word embeddings for phrase break prediction and were able to achieve F-score of 92 with a simple RNN.

Rosenberg et al. (2015) also tried to apply RNN architectures to the phrasing detection task and achieved accuracy of almost 93% building a bi-directional RNN architecture.

2.2.3. Sentence stress prediction task

Sentence stress has been studied less when compared to phrasing, however, we list most important results achieved in the course of various research below.

Gary Geunbae Lee et al. (2017) proposed a sentence stress prediction model using the contexual information containing two words preceding a target word and three words following it and the following features:

Word_class + pos_tag,

Word_identity + pos_tag,

Vowel_number + pos_tag

Syllable_number + pos_tag

Linear-chain CRF was applied to the task and F-measure of 97.2 on the prediction task was achieved.

2.3. Neural Network Language Models

A **language model** is a function, or an algorithm for learning such a function, that captures the salient statistical characteristics of the distribution of sequences of words in a natural language, typically allowing one to make probabilistic predictions of the next word given preceding ones.

A **neural network language model** is a language model based on Neural Networks, exploiting their ability to learn distributed representations to reduce the impact of the so-called *curse of dimensionality* (Bengio et al., 2003).

In the context of learning algorithms, **the curse of dimensionality** refers to the need for huge numbers of training examples when learning highly complex functions. When the number of input variables increases, the number of required examples can grow exponentially. In the context of language models, the problem comes from the huge number of possible sequences of words, e.g., with a sequence of 10 words taken from a vocabulary of 100,000 there are 1050 possible sequences.

A **distributed representation** of a symbol is a vector of features which characterize the meaning of the symbol, and are not mutually exclusive. If a human were to choose the features of a word, he might pick grammatical features like gender or plurality, as well as semantic features like *animate* or *invisible*. With a neural network language model, one relies on the learning algorithm to discover these

features, and the features are continuous-valued (what makes the optimization problem involved in the process of learning much simpler).

The basic idea is to learn to associate each word in the dictionary with a continuous-valued vector representation. Each word corresponds to a point in a feature space. One can imagine that each dimension of that space corresponds to a semantic or grammatical characteristic of words. The hope is that functionally similar words get to be closer to each other in that space, at least along some directions. A sequence of words can thus be transformed into a sequence of these learned feature vectors. The neural network learns to map that sequence of feature vectors to a prediction that we are interested in, such as the probability distribution over the next word in the given sequence (Bengio et al., 2003).

The advantage of this distributed representation approach is that it allows the model to generalize well to sequences that are not in the set of training word sequences, but that are similar in terms of their features, i.e., their distributed representation. Because neural networks tend to map nearby inputs to nearby outputs, the predictions corresponding to word sequences with similar features are mapped to similar predictions. Because many different combinations of feature values are possible, a very large set of possible meanings can be represented compactly, allowing a model with a comparatively small number of parameters to fit a large training set (Bengio et al., 2003).

2.4. Word Embeddings

Word embeddings are a type of word representation that allows words with similar meaning to have a similar representation.

They are a distributed representation for text that is perhaps one of the key breakthroughs for the impressive performance of deep learning methods on challenging natural language processing problems.

For the present task, it has been decided to apply GloVe (Global Vectors for word representations) proposed by the group of researchers from Stanford in 2014 as there is an extensive pre-trained model containing nearly 2M of vocabulary. Another approach would be to train a word-2-vec model ourselves, however, given the small size of dataset, this option would far less effective.

GloVe is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed in aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructure of the word vector space.

Fig. 2.1 illustrates an example of vectors that quantify the relatedness of two words. You can see that the model perfectly captures the gender relation placing *man* and *woman*, *sister* and *brother* etc. on different ends of the vector field.

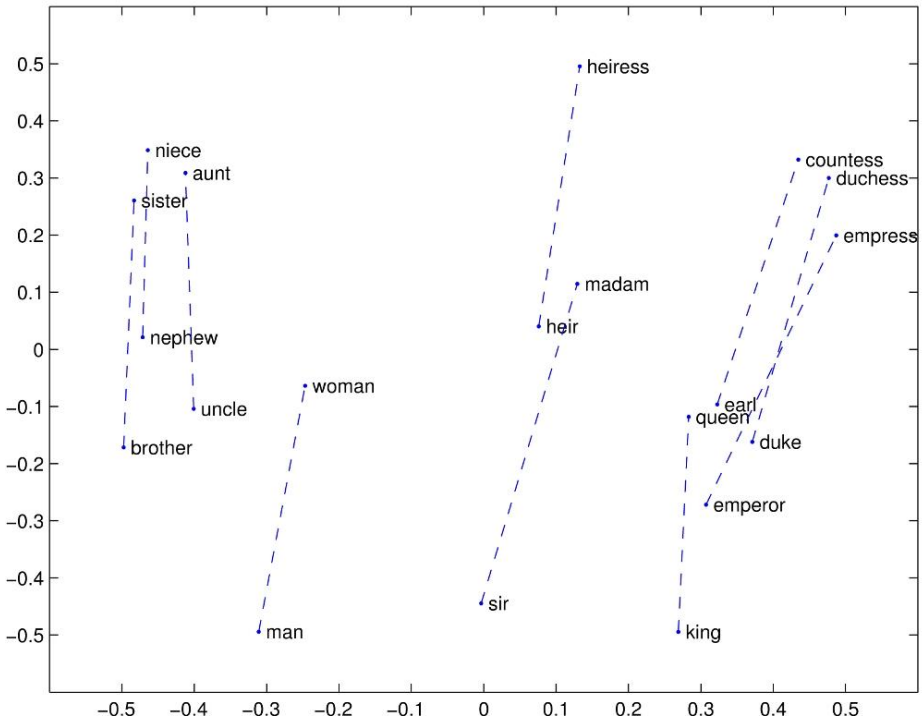


Figure 2.1 Example of word embedding vectors (source: https://nlp.stanford.edu/projects/glove/)

The underlying concept that distinguishes man from woman, i.e. sex or gender, may be as well specified by other word pairs, such as *sir – madam*, *king – queen*. To explain this observation from the mathematical point of view, we might expect that the vector differences man – woman, sir – madam and king – queen might be roughly equal and doing vector operations like **king – man + woman** will give us a **queen**.

2.5. DNN, RNN, LSTM and GRU

A deep neural network (DNN) is a feed-forward, artificial neural

network with multiple layers between the input and output layers. It can be defined as a conventional multilayer perceptron (MLP) with many (often more than 2) hidden layers. Each hidden unit typically uses the logistic function to map its total input from the layer below to the scalar state that it sends to the layer above (Hinton, G., 2012). The DNN finds the correct mathematical manipulation to turn the input into the output, whether it be a linear relationship or a non-linear relationship. The network moves through the layers calculating the probability of each output.

DNNs are typically feedforward networks in which data flows from the input layer to the output layer without looping back. At first, the DNN creates a map of virtual neurons and assigns random numerical values, or "weights", to connections between them. The weights and inputs are multiplied and return an output between 0 and 1. If the network didn't accurately recognize a particular pattern, an algorithm would adjust the weights. That way the algorithm can make certain parameters more influential, until it determines the correct mathematical manipulation to fully process the data.

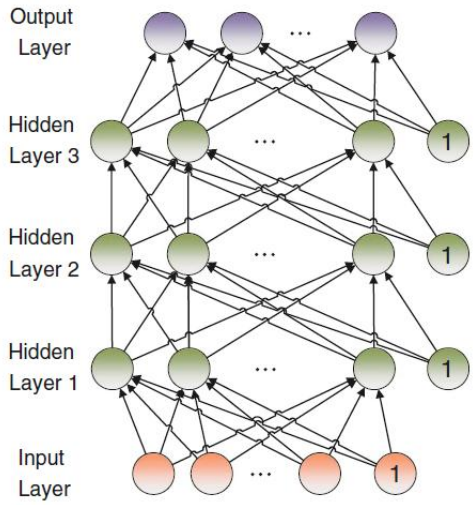


Figure 2.2 An example DNN with an input layer, three hidden layers, and an output layer (taken from Dong Yu, et al. 2014)

A **recurrent neural network** (**RNN**) is a class of artificial neural network where connections between nodes form a directed graph along a sequence. In a traditional neural network we assume that all inputs (and outputs) are independent of each other. But for many tasks, especially for language, it is not a very good idea. The RNN is a class of neural network models where connections among many of its units form a directed cycle, hence the term *recurrent*. Such a cycle or recurrence is associated with the time-delay operation. The use of time-delayed recurrence over the temporal dimension gives rise to the *memory* structure, expressed as internal states, in the RNN, permitting it to exhibit the type of dynamic temporal behavior (Dong Yu et. al, 2014).

In theory RNNs can make use of information in arbitrarily long

sequences, but in practice they are limited to looking back only a few steps. Here is what a typical RNN looks like:

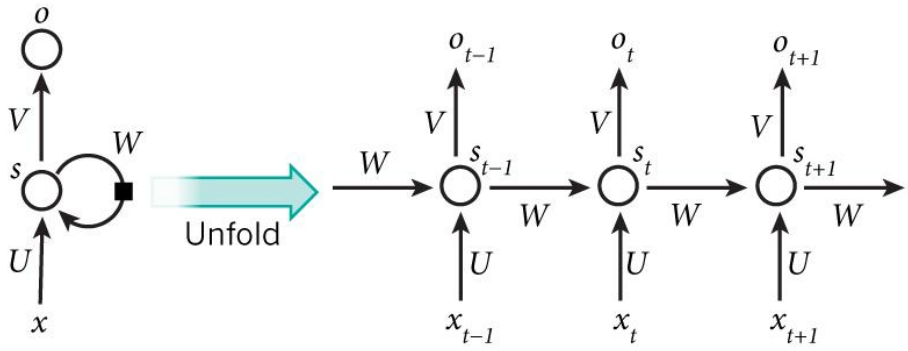


Figure 2.2 (RNN architecture)

The above diagram shows a RNN being **unrolled** (or unfolded) into a full network. By unrolling we simply mean that we write out the network for the complete sequence. For example, if the sequence we care about is a sentence of 5 words, the network would be unrolled into a 5-layer neural network, one layer for each word. The formulas that govern the computation happening in a RNN are as follows:

- x_t is the input at time step t. For example, x_1 could be a one-hot vector corresponding to the second word of a sentence.
- s_t is the hidden state at time step t. It's the "memory" of the network. s_t is calculated based on the previous hidden state and the input at the current step: $s_t = f(Ux_t + Ws_{t-1})$. The function f usually is a nonlinearity such as tanh or ReLU. s_{-1}, which is required to calculate the first hidden state, is typically initialized to all zeroes.

- o_t is the output at step t. For example, if we wanted to predict the next word in a sentence it would be a vector of probabilities across our vocabulary. $o_t = \text{softmax}(Vs_t)$.

Bidirectional RNNs are based on the idea that the output at time t may not only depend on the previous elements in the sequence, but also future elements. For example, to predict a missing word in a sequence you want to look at both the left and the right context. Bidirectional RNNs are quite simple. They are just two RNNs stacked on top of each other. The output is then computed based on the hidden state of both RNNs.

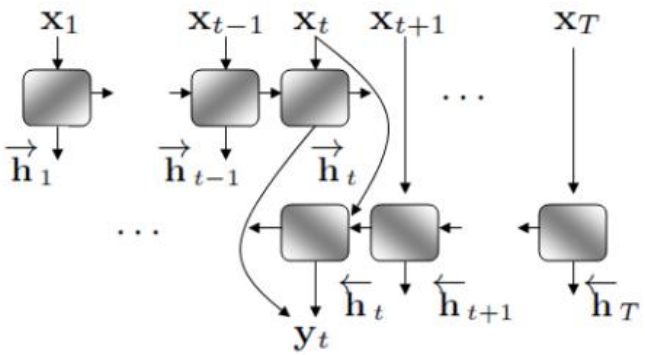


Figure 2.4. Bidirectional RNN (source: Fernandez et al. 2014)

Long short-term memory (**LSTM**) units are a building unit for layers of a recurrent neural network (RNN). A RNN composed of LSTM units is often called an LSTM network. A common LSTM unit is composed of a **cell**, an **input gate**, an **output gate** and a **forget gate**.

The cell is responsible for "remembering" values over arbitrary time intervals; hence the word "memory" in LSTM.

LSTMs contain information outside the normal flow of the recurrent network in a gated cell. Information can be stored in, written to, or read from a cell, much like data in a computer's memory. The cell makes decisions about what to store, and when to allow reads, writes and erasures, via gates that open and close. Like the basic RNNs, the LSTM version can be shown to be a universal computing machine in the sense that given enough network units, it can compute anything a conventional computer can compute and if it has proper weight matrices. But unlike the basic RNNs, the LSTM version is better-suited to learn from input sequence data to classify, process, and predict time series when there are very long time lags of unknown lengths between important events (Dong Yu et. al, 2014).

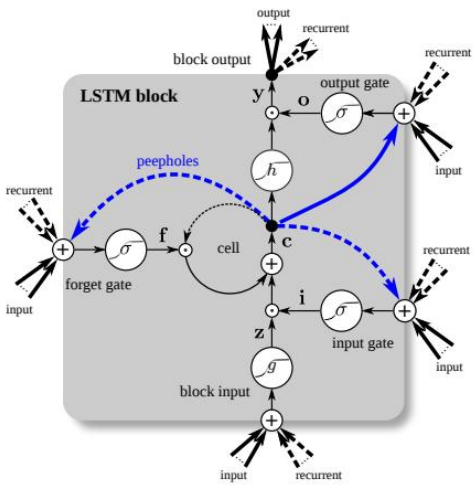


Figure 2.5 LSTM architecture (source: Greff et. al, 2015)

A gated recurrent unit (GRU) is basically an LSTM without an **output gate**, which therefore fully writes the contents from its memory cell to the larger net at each time step.

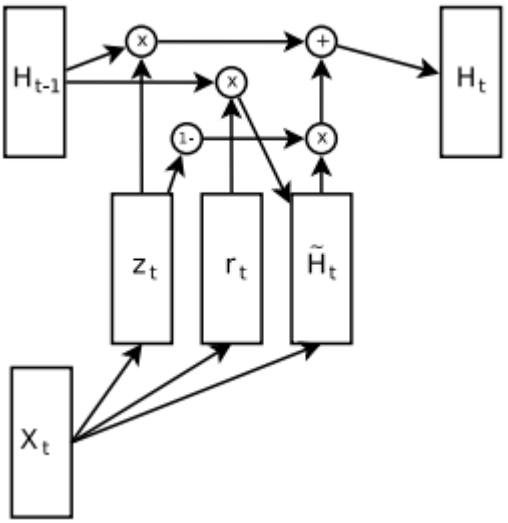
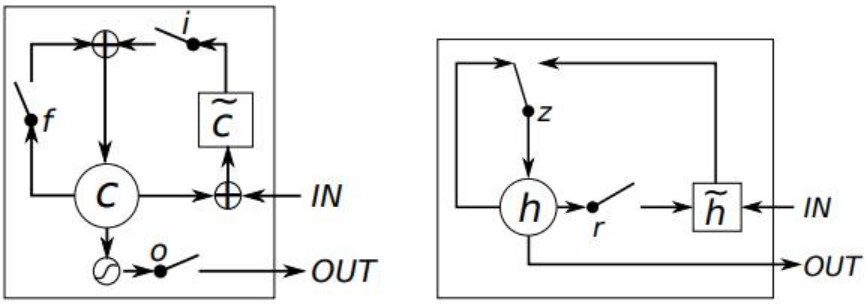


Figure 2.6 Gated Recurrent Unit (source: Józefowicz, R. et. al, 2015)



(a) Long Short-Term Memory (b) Gated Recurrent Unit

Figure 2.7 LSTM and Gated Recurrent Unit (source: Chung, J., Gülçehre, Ç., Cho, K., & Bengio, Y., 2014)

Chapter 3

Experiment setup and Results

3.1. Corpus

The Aix-MARSEC database consists of over five hours of natural-sounding British English speech data collected from 53 different speakers (17 males and 36 females). The corpus includes approximately 55,000 orthographically transcribed words. The annotations currently include: phonemes, syllables, syllable constituents, rhythm units, stress feet, words, intonation units**,** together with the output of the automatic MOMEL modelling and INTSINT symbolic coding algorithms.

The prosodic annotations in Aix-Marsec corpus were inherited from the original SEC (Spoken English Corpus) which was manually transcribed with prosodic marks. The difference between stressed and accented syllables is as follows:

An accented syllable has an independent pitch movement associated with it, known as the tone. Tones are marked with iconic symbols representing the pitch movement. Syllables which are felt to be stressed but not accented (i.e. they are prominent but have no independent pitch movement) are marked with a circle. Unstressed syllables are left unmarked.

The pitch of all unaccented syllables is predictable from the

tone marks on neighbouring accented syllables.

The rhythm unit annotation is based on Jassem's (1952) notion of Anacrusis (ANA) and Narrow Rhythm Unit (NRU) and the stress foot annotation on Abercrombie's (1964) notion of Foot (F) (Hirst et al., 2009).

Narrow Rhythm Unit consists of a stressed syllable followed by a sequence of unstressed syllables and ends at the following word boundary. **Foot** begins with a stressed syllable or an intonation boundary and ends before the following stressed syllable or at the next intonation boundary. Hence the right boundary of the Narrow Rhythm Unit coincides with the word boundary whereas that of the Foot is often placed inside a word before the next stressed syllable. Hirst et al. gives the following example of a foot with "|" corresponding to foot boundaries:

a) They preDICted his eLECtion.

would be analysed as

b) They pre- | DICted his e- | LECtion. |

Anacrusis consists of a sequence of proclitic unstressed syllables. Anacrusis and Narrow Rhythm Unit combine to constitute the Total Rhythm Unit. Jassem claimed in particular that the rhythmic organisation of these two types of constituents is completely different: unstressed syllables in the Anacrusis tend to be pronounced "extremely rapidly" whereas the duration of each syllable in a Narrow Rhythm Unit tends to be inversely proportional to the number of syllables in that unit.

In Abercrombie (1964), utterance-initial unstressed syllables or words preceding the first stressed syllable form a separate Foot.

Following the approach introduced in the study by Lee Gary Gaeunbae et. al (2017), if we also regard the first syllable in NRU as a stressed syllable, we can easily extract words where sentence stress is imposed. Hence we used Jassem's rhythm unit annotations to implement our sentence stress prediction model.

3.2. Approach

For the present task we use word embeddings and Recurrent Neural Networks that have been shown to achieve the best performance when dealing with language. As described in Chapter 2, various studies have also applied RNNs and its extensions.

However, to illustrate the difference, we also conducted experiments with simple DNNs and tried out LSTM, bidirectional LSTM (reads the data forward and backward that allow it to remember left-right and right-left dependencies) and GRU also adding and removing additional information like POS-tags and phrase break annotations.

The data is structured as follows. We have text data (sequences of words) with each word having a label. Text data and labels are stored separately. Below is the example of a piece of labeled sentence stress data. Phrase break data has the same structure.

[‘The’, ‘U’] [‘**Miller's’, ‘S’]** [‘**Reel’, ‘S’]** [‘**takes’**, **‘S’**] [‘the’, ‘U’]

[‘**form’, ‘S’]** [‘of’, ‘U’] [‘a’, ‘U’] [‘**love’, ‘S’]** [‘**story’**, **‘S’**] [‘**woven’, ‘S’]** [‘from’, ‘U’] [‘the’, ‘U’] [‘**letters’, ‘S’]** [‘**poems’, ‘S’]** [‘and’, ‘U’] [‘**songs’, ‘S’]** [‘of’, ‘U’] [**‘Robert’, ‘S’**] [‘**Burns’, ‘S’]** [‘and’] [**‘features’, ‘S’**] [‘the’, ‘U’] [‘**singing’**, ‘S’] [‘of’, ‘U’] [**‘Jean’, ‘S’**] [‘**Redpath’** , ‘S’] [‘and’, ‘U’] [**‘Rod’, ‘S’**] [‘**Patterson’**, ‘S’]

Next, text data is input into the GLoVe model that builds word embedding model. The model has 300 dimensions, meaning that each word will be represented as a 300-dimensional vector.

Then we input these vectors together with the labels into the corresponding neural network model that is further trained to give us the correct prediction.

3.2.1. POS-tagger

The process of classifying words into their parts of speech and labeling them accordingly is known as part-of-speech tagging, POS-tagging, or simply tagging. Parts of speech are also known as word classes or lexical categories. The collection of tags used for a particular task is known as a tagset (Bird et al., 2009).

Aix-Marsec corpus is not annotated with POS-tags. For the purposes of the present research we implement a POS-tagger provided by nltk package in Python (Bird et al., 2009). All in all there are 34 tags, with ADJ staying for adjective, N for noun and NP for proper noun. A table with detailed information can be found in the Appendix 1.

3.2.2. Activation Functions

In artificial neural networks, the activation function of a node defines the output of that node given an input or set of inputs. In other words, the activation function is used to transform the activation level of a neuron into an output signal (Karlik et al.)

If we consider a neuron (see the equation below), the value of its output (Y) can be any number ranging from –inf to +inf because the neuron itself doesn't really know the bounds. If the value can be anything, i.e. if we don't have any threshold determining the bounds, the output of the neuron can take on very large values and the neuron doesn't know whether it should fire or not. The activation function in its turn, maps the output of a neuron to the established bounds (e.g., between 0 and 1) and takes the decision of whether or not to pass the signal. In other words, the activation function allows us to check the output value produced by a neuron and decide whether outside connections should consider this neuron as activated or not.

$$Y = \sum (\text{weight} * \text{input}) + \text{bias}$$

Activation functions can be of two types:

1. Linear activation function
2. Non-linear activation functions

Linear activation function is a simple function of the form f(x) = x, that basically passes the output without any modifications.

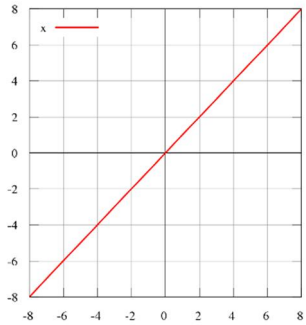


Fig. 3.1 Linear activation function

Non-linear activation functions allow neural networks to compute nontrivial problems using only a small number of nodes. They are used to separate the linearly inseparable data. Few examples of different non-linear activation functions include sigmoid, tanh, ReLU etc.

Karlik et al. analyses the performance of generalized multi-layer-perceptron (MLP) architectures using various different activation functions for the neurons of hidden and output layers. For experimental comparison they use different types of sigmoid, tanh and other widely-used activation functions.

Below we give a brief summary of some popular non-liner activation functions that we also implemented in the course of our experiment:

1. Sigmoid function is also known as logistic activation function and is used when our final goal is to predict probability in the output layer as it squashes real-valued numbers into a range between 0 and 1. Mathematically it is represented as:

$$f(x) = \sigma(x) = \frac{1}{1 + e^{-x}}$$

Graphically it can be represented as:

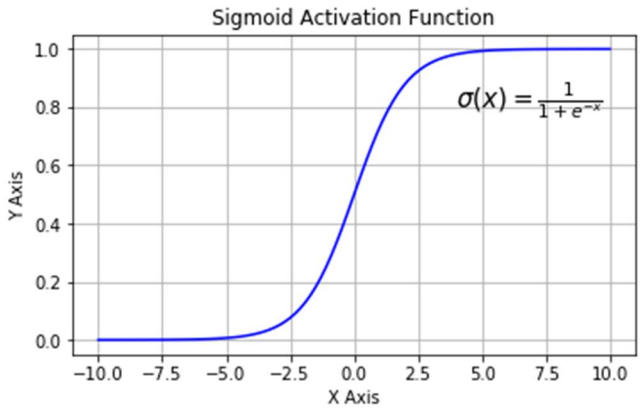


Fig. 3.2 Sigmoid activation function

2. Tahn activation function is known as the hyperbolic tangent activation function and is commonly used in Long-short term memory neural networks. Tanh also takes a real-valued number but limits in into a range between -1 and 1.

Mathematically it is represented as:

$$f(x) = \tanh(x) = \frac{(e^{x} - e^{-x})}{(e^{x} + e^{-x})}$$

Graphical representation is the following:

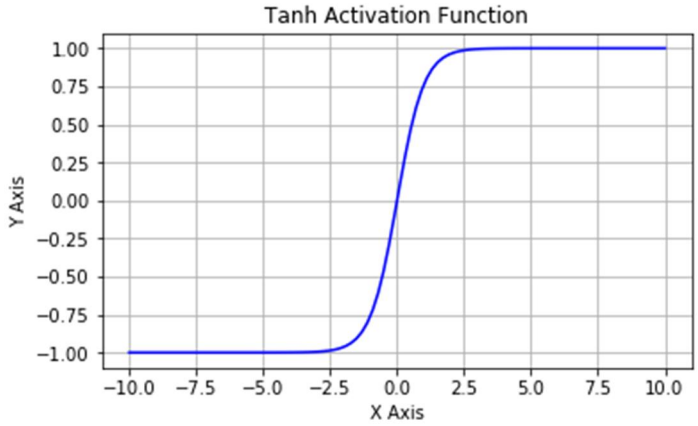


Fig. 3.3 Tahn activation function

3. ReLU (Rectified Linear Unit) is mathematically represented by the following simple equation:

$$f(x) = \max(0, x)$$

This means that when the input x < 0, the output is 0 and if x > 0, the output is x. When represented graphically, it looks as follows:

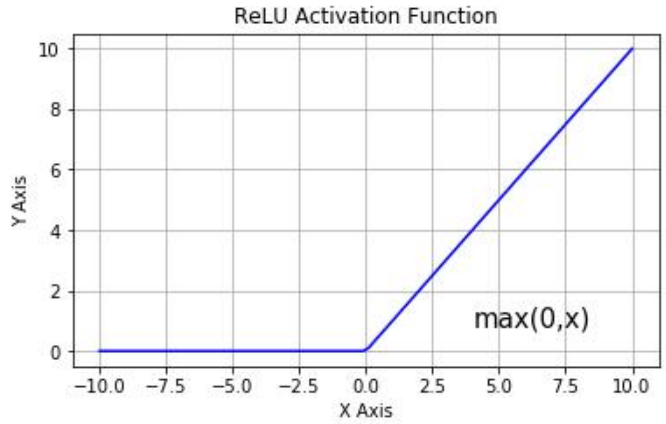
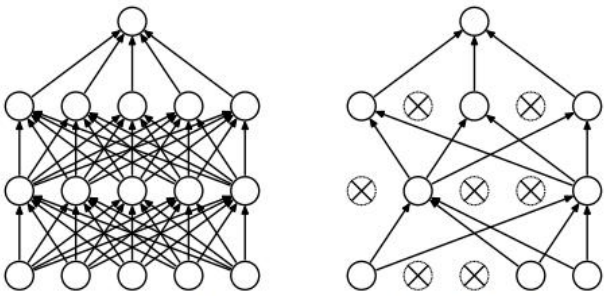


Fig. 3.4 ReLU activation function

Although sigmoidal and tanh functions are the most commonly used nonlinear functions, their limitations are well known (Deng, 2014). Both suffer from the so-called vanishing gradient problem when the gradient of neuron's output is so vanishingly small (close to 0) that it prevents the weight from changing its value. The neuron becomes saturated and in the worst case this may stop the neural network from further training. ReLU is used to address this problem. Jaitly and Hinton (2011) were the first to implement ReLU in the DNNs to the speech recognition task to overcome the drawbacks of the sigmoid function. Mass et al. also successfully applied rectified linear units to LVCSR (large vocabulary speech recognition), with the best accuracy

obtained when combining ReLU with the dropout regularization technique that is the topic of our further discussion.

3.2.3. Dropout



(a) Standard Neural Net (b) After applying dropout.

Figure 1: Dropout Neural Net Model. **Left**: A standard neural net with 2 hidden layers. **Right**: An example of a thinned net produced by applying dropout to the network on the left. Crossed units have been dropped.

Fig. 3.5 Dropout Neural Net model (source: Srivatsava, 2014)

The basic idea of dropout is to randomly omit a certain percentage (e.g., α) of the neurons in each hidden layer for each presentation of the samples during training. This means during the training each random combination of the (1−α) remaining hidden neurons needs to perform well even in the absence of the omitted neurons. This requires each neuron to depend less on other neurons to detect patterns. Alternatively, dropout can be considered a technique that adds random noise to the training data. This is because each higher-layer neuron gets input from a random collection of the lower-layer neurons. The excitation received by each neuron is different even if the same input is fed into the DNN.

With dropout, DNNs need to waste some of the weights to

remove the effect of the random noise introduced. As such, dropout essentially reduces the capacity of the DNN, and thus can improve generalization of the resulting model. When a hidden neuron is dropped out, its activation is set to 0 and so no error signal will pass through it. This means that other than the random dropout operation, no other changes to the training algorithm are needed to implement this feature. At the test time, however, instead of using a random combination of the neurons at each hidden layer, we use the average of all the possible combinations.

3.2.4. Evaluation Metrics

One measure of accuracy is the number of correct predictions to the total number of input samples, i.e. to the total number of predictions made.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions\ made}$$

In their research, Taylor and Black (1998) measured the percentage of breaks and non-breaks predicted correctly, and the total word boundaries predicted correctly. However, breaks and non-breaks typically have a very skewed distribution in a corpus what makes the data very imbalanced. The number of NBs usually turns out to be relatively high. Thus, a model that predicts all boundaries as NB is likely to get a very high total accuracy. Looking at the individual

accuracies on breaks and non-breaks is not ideal, because comparing two models that way is fairly difficult.

This problem can be resolved by calculating the accuracy in terms of precision and recall scores, and then combining them into a F-measure (van Rijsbergen, 1979), which is a harmonic mean between precision and recall. It tells us how precise our classifier is (how many predictions it makes correctly), as well as how robust it is.

Precision tells how many of the predicted breaks are correct. Recall tells how many of the actual breaks were predicted (Parlikar, 2013). The combined harmonic mean value is thus a good indicator of overall quality of a model.

High precision but lower recall, is considered extremely accurate, but it then misses a large number of instances that are difficult to classify. The greater the F1 Score, the better is the performance of our model. Mathematically, it can be expressed as:

$$F - measure = 2 * \frac{1}{\frac{1}{precision} + \frac{1}{recall}}$$

Precision is the number of correct positive results divided by the number of positive results predicted by the classifier.

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives}$$

Recall is the number of correct positive results divided by the total amount of given relevant samples (all instances that should have been identified as positive).

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives}$$

3.3. Automatic phrase break prediction

Phrase break prediction task can be treated as simple classification task. We decided not to restrict the scope of the present work to the prediction of only two classes (Breaks and Non-Breaks). Instead, we differentiate between final pauses coming at the end of the sentence (labeled BB) and intermediate pauses (labeled B) that appear in the sentence itself. The absence of pauses is marked as NB. Below is the example.

[‘Good’, ‘morning’]

['NB', 'BB']

As have been mentioned above, we use RNN together with pre-trained glove word embedding model.

3.4. Automatic sentence stress prediction

Similarly to phrase break prediction task sentence stress prediction is also a classification task where the model has to assign each word to a class depending on whether it defines it as stressed (labeled S) or

unstressed (labeled U).

Here we present three different approaches, training only a sentence stress model and sentence stress model that uses the decisions (output) of phrase break prediction model. The results are illustrated in the next section.

Furthermore, we also make use of data balancing adding more weights to classes that appear less frequently in the data.

3.5. Experimental Results

The results are shown in the table below. All the results are represented as **mean accuracy over the test set**.

3.5.1 Phrase break experiment

As we mentioned before, we distinguish two types of pauses labeled B and BB correspondingly in the experimental setup. BB signals a major intonation boundary while B marks a minor one.

Below we illustrate the approach on the example of our deep model with several layers.

The first step is data preprocessing (Fig. 3.6). We take our dataset, divide it into train (90%) and test (10%) sets, extract labels and input text data into the word embedding model containing 300 vectors. Additionally, we add POS-data acquired with the help of nltk pos-tagger what adds 34 more vectors to our total number of vectors.

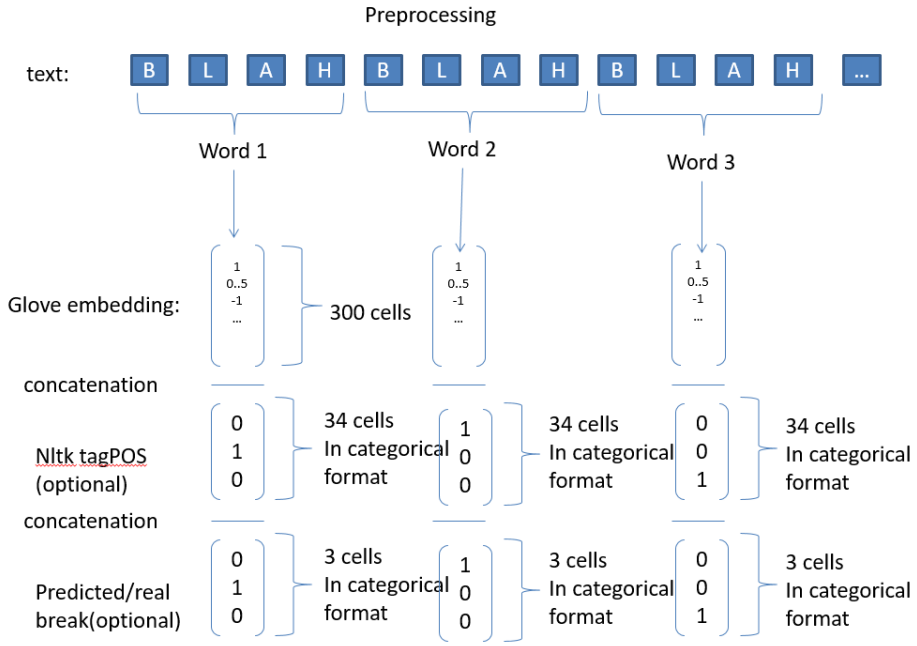


Fig. 3.6 Data preprocessing stage and features used

The next step is training. The first layer of the architecture is a simple DNN layer with 200 nodes. The second layer is a bidirectional LSTM layer (Fig. 3.7). The third layer is a LSTM layer with 100 nodes which is used to combine features extracted during reading data from left to right and from right to left. The next layer is again a DNN layer used to extract deeper features considering feature-vectors independently (Fig. 3.8).

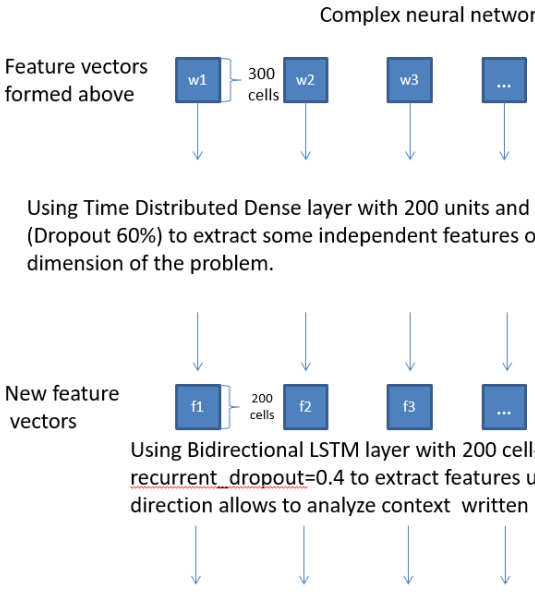


Fig. 3.7 First layer of the complex neural model

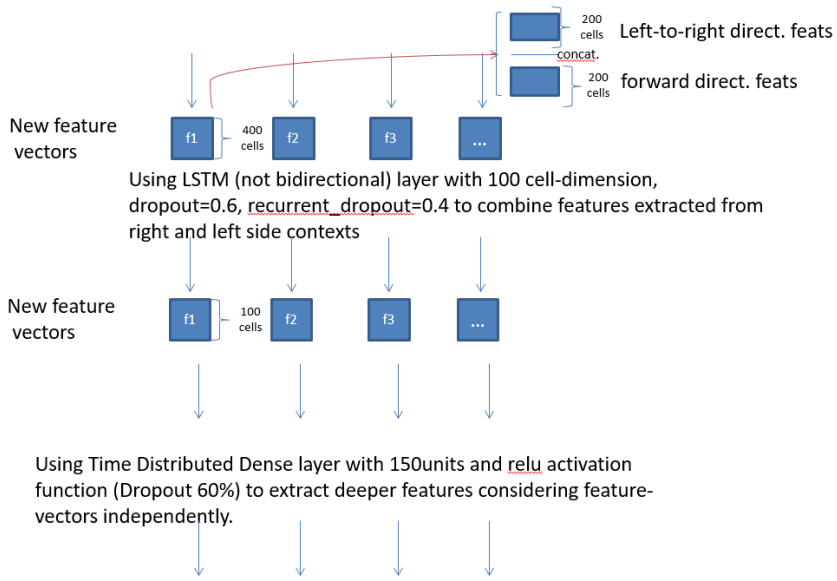


Fig. 3.8 Second layer of the complex neural model

The last layer is a DNN layer with a softmax function used to get probabilities of our classes (B, BB or NB / S or U) in a categorical format.

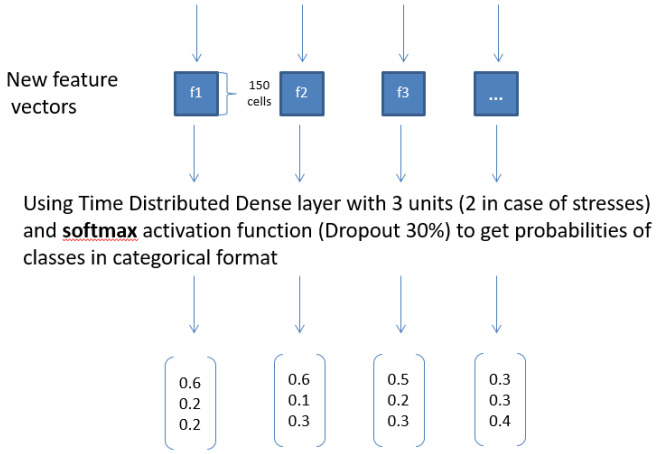


Fig. 3.9 Last layer of the model with probabilities on categorical format

The data is structured as follows:

Table 3.1 Size of phrase breaks data

NB	B	BB
4618	915	326

Model output is the following:

1) ['NB', 'NB', 'B', 'NB', 'NB', 'B', 'NB', 'NB', 'NB', 'NB', 'B', 'NB', 'NB', 'NB', 'B', 'NB', 'B', 'NB', 'NB', 'B', 'NB', 'NB', 'NB', 'B', 'NB', 'NB', 'B', 'NB', 'NB', 'BB']

The Miller's Reel | takes the form of a love story | woven | from the letters | poems | and songs of Robert Burns | and features the singing of Jean Redpath and Rod Patterson ||

2) ['NB', 'NB', 'NB', 'B', 'B', 'NB', 'NB', 'B', 'B', 'NB', 'NB', 'BB']
that's The Miller's Reel | especially for Burns night | here on Radio ||

The results of the experiment are illustrated in the tables below.

Table 3.2 Type of Neural Network

Architecture	Mean accuracy	Precision	Recall	F1-score
DNN	0.783	NB: 0.84 B: 0.43 BB: 0.16 Avg.: 0.74	NB: 0.93 B: 0.29 BB: 0.05 Avg.: 0.78	NB: 0.88 B: 0.35 BB: 0.08 Avg.: 0.75
RNN	0.783	NB: 0.80 B: 0.45 BB: 0.07 Avg.: 0.71	NB: 0.98 B: 0.08 BB: 0.01 Avg.: 0.78	NB: 0.88 B: 0.13 BB: 0.02 Avg.: 0.72
GRU	0.787	NB: 0.81 B: 0.49 BB: 0.22 Avg.: 0.72	NB: 0.98 B: 0.09 BB: 0.06 Avg.: 0.79	NB: 0.88 B: 0.15 BB: 0.10 Avg.: 0.73
LSTM	0.782	NB: 0.81 B: 0.44 BB: 0.19 Avg.: 0.72	NB: 0.96 B: 0.14 BB: 0.08 Avg.: 0.78	NB: 0.88 B: 0.21 BB: 0.11 Avg.: 0.73

Table 3.3 Bidirectional Neural Networks

Architecture	Mean accuracy	Precision	Recall	F1-score
biRNN	0.886708	NB: 0.90 B: 0.57 BB: 1.00 Avg.: 0.87	NB: 0.96 B: 0.32 BB: 1.00 Avg.: 0.89	NB: 0.93 B: 0.41 BB: 1.00 Avg.: 0.87
biGRU	**0.891002**	NB: 0.91 B: 0.60 BB: 0.60 Avg.: 0.88	NB: 0.96 B: 0.35 BB: 0.99 Avg.: 0.89	NB: 0.93 B: 0.44 BB: 1.00 Avg.: 0.88
biLSTM	0.804	NB: 0.88 B: 0.46 BB: 1.00 Avg.: 0.82	NB: 0.92 B: 0.51 BB: 0.00 Avg.: 0.80	NB: 0.90 B: 0.48 BB: 0.01 Avg.: 0.78
Bidirectional complex	0.887117	NB: 0.91 B: 0.57 BB: 1.00 Avg.: 0.87	NB: 0.96 B: 0.36 BB: 1.00 Avg.: 0.89	NB: 0.93 B: 0.44 BB: 1.00 Avg.: 0.88

Table 3.4 POS-tag information added

Architecture	Mean accuracy	Precision	Recall	F1-score
DNN + POS	0.783	NB: 0.85 B: 0.43 BB: 0.19 Avg.: 0.75	NB: 0.93 B: 0.31 BB: 0.08 Avg.: 0.78	NB: 0.89 B: 0.36 BB: 0.12 Avg.: 0.76
RNN + POS	0.780	NB: 0.80 B: 0.43 BB: 0.17 Avg.: 0.71	NB: 0.97 B: 0.05 BB: 0.09 Avg.: 0.78	NB: 0.88 B: 0.10 BB: 0.11 Avg.: 0.72
GRU + POS	0.776	NB: 0.83 B: 0.42 BB: 0.18 Avg.: 0.73	NB: 0.94 B: 0.21 BB: 0.13 Avg.: 0.78	NB: 0.88 B: 0.28 BB: 0.15 Avg.: 0.75
LSTM + POS	0.785	NB: 0.82 B: 0.46 BB: 0.19 Avg.: 0.73	NB: 0.96 B: 0.16 BB: 0.09 Avg.: 0.79	NB: 0.88 B: 0.23 BB: 0.11 Avg.: 0.74

Table 3.5 POS-tags + bidirectional network

Architecture	Mean accuracy	Precision	Recall	F1-score
biRNN + POS	0.888344	NB: 0.90 B: 0.59 BB: 1.00 Avg.: 0.87	NB: 0.97 B: 0.31 BB: 1.00 Avg.: 0.89	NB: 0.93 B: 0.41 BB: 1.00 Avg.: 0.87
biGRU + POS	0.897751	NB: 0.92 B: 0.63 BB: 1.00 Avg.: 0.89	NB: 0.96 B: 0.42 BB: 1.00 Avg.: 0.90	NB: 0.94 B: 0.50 BB: 1.00 Avg.: 0.89
biLSTM+POS	0.807134	NB: 0.90 B: 0.46 BB: 1.00 Avg.: 0.84	NB: 0.90 B: 0.62 BB: 0.00 Avg.: 0.81	NB: 0.90 B: 0.52 BB: 0.01 Avg.: 0.79
Bidirectional Complex+POS	0.885890	NB: 0.91 B: 0.55 BB: 1.00 Avg.: 0.87	NB: 0.95 B: 0.42 BB: 1.00 Avg.: 0.89	NB: 0.93 B: 0.47 BB: 1.00 Avg.: 0.88

The best performance with the accuracy equaling 0.897751 (89,7%) was achieved by biGRU with POS-tag information added as a feature.

The results are discussed in more detail in Section 3.6.

3.5.2 Sentence stress experiment

Table 3.6 Sentence stress experiment results

Model	Mean acc.	Precision	Recall	F1-Score
DNN	0.929857	S: 0.94 U: 0.91 Avg: 0.93	S: 0.92 U: 0.94 Avg: 0.93	S: 0.93 U: 0.92 Avg: 0.93

RNN	0.921063	S: 0.92 U: 0.92 Avg: 0.92	S: 0.93 U: 0.91 Avg: 0.92	S: 0.93 U: 0.91 Avg: 0.92
GRU	0.927607	S: 0.94 U: 0.92 Avg: 0.93	S: 0.93 U: 0.93 Avg: 0.93	S: 0.93 U: 0.92 Avg: 0.93
LSTM	0.926380	S: 0.93 U: 0.92 Avg: 0.93	S: 0.93 U: 0.92 Avg: 0.93	S: 0.93 U: 0.92 Avg: 0.93
Neural architectures with POS				
DNN with POS	0.928630	S: 0.95 U: 0.90 Avg: 0.93	S: 0.91 U: 0.95 Avg: 0.93	S: 0.93 U: 0.92 Avg: 0.93
RNN with POS	0.922086	S: 0.93 U: 0.91 Avg: 0.92	S: 0.92 U: 0.92 Avg: 0.92	S: 0.93 U: 0.92 Avg: 0.92
GRU with POS	0.928425	S: 0.94 U: 0.92 Avg: 0.93	S: 0.93 U: 0.93 Avg: 0.93	S: 0.93 U: 0.92 Avg: 0.93
LSTM with POS	0.931493	S: 0.94 U: 0.92 Avg: 0.93	S: 0.93 U: 0.93 Avg: 0.93	S: 0.94 U: 0.93 Avg: 0.93
Annotated phrase break data used				
DNN with annotated PB	0.927812	S: 0.93 U: 0.92 Avg: 0.93	S: 0.93 U: 0.92 Avg: 0.93	S: 0.93 U: 0.92 Avg: 0.93
RNN with annotated PB	0.919018	S: 0.92 U: 0.92 Avg: 0.92	S: 0.93 U: 0.90 Avg: 0.92	S: 0.93 U: 0.91 Avg: 0.92
GRU with annotated PB	0.929857	S: 0.94 U: 0.92 Avg: 0.93	S: 0.93 U: 0.93 Avg: 0.93	S: 0.93 U: 0.92 Avg: 0.93
LSTM with annotated PB	0.937832	S: 0.95 U: 0.93 Avg: 0.94	S: 0.94 U: 0.94 Avg: 0.94	S: 0.94 U: 0.93 Avg: 0.94
Bidirectional architectures				

biRNN	0.819427	S: 0.84 U: 0.80 Avg: 0.82	S: 0.83 U: 0.81 Avg: 0.82	S: 0.83 U: 0.80 Avg: 0.82
biGRU	0.927403	S: 0.93 U: 0.92 Avg: 0.93	S: 0.93 U: 0.92 Avg: 0.93	S: 0.93 U: 0.92 Avg: 0.93
biLSTM	0.930675	S: 0.94 U: 0.92 Avg: 0.93	S: 0.93 U: 0.93 Avg: 0.93	S: 0.94 U: 0.92 Avg: 0.93
Bidirectional architectures with POS				
biRNN with POS	0.838037	S: 0.85 U: 0.82 Avg: 0.84	S: 0.84 U: 0.83 Avg: 0.84	S: 0.85 U: 0.83 Avg: 0.84
biGRU with POS	0.927198	S: 0.93 U: 0.92 Avg: 0.93	S: 0.93 U: 0.92 Avg: 0.93	S: 0.93 U: 0.92 Avg: 0.93
biLSTM with POS	**0.972629**	S: 0.98 U: 0.97 Avg: 0.97	S: 0.95 U: 0.99 Avg: 0.97	S: 0.97 U: 0.98 Avg: 0.97
Bidirectional architectures with annotated phrase break data				
biRNN with annotated PB	0.811656	S: 0.78 U: 0.86 Avg: 0.82	S: 0.90 U: 0.71 Avg: 0.81	S: 0.84 U: 0.78 Avg: 0.81
biGRU with annotated PB	0.926994	S: 0.93 U: 0.93 Avg: 0.93	S: 0.94 U: 0.91 Avg: 0.93	S: 0.93 U: 0.92 Avg: 0.93
biLSTM with annotated PB	0.932311	S: 0.94 U: 0.92 Avg: 0.93	S: 0.93 U: 0.94 Avg: 0.93	S: 0.94 U: 0.93 Avg: 0.93

Examples of the model output (in bold are the words that are stressed):

Example 1:

Data as annotated:

The **Miller's Reel** | **takes** the **form** of a **love story** | **woven** | from the **letters** | **poems** | and **songs** of **Robert Burns** | and **features** the **singing**

of **Jean Redpath** and **Rod Patterson**

Model output:

['U', 'S', 'S', 'S', 'U', 'S', 'U', 'U', 'S', 'S', 'S', 'U', 'U', 'S', 'S', 'U', 'S', 'U', 'S', 'S', 'U', 'S', 'U', 'S', 'U', 'S', 'U', 'S', 'S']

Predicted by the bidirectional LSTM model:

The **Miller's Reel** | **takes** the **form** of a **love story** | **woven** | from the **letters** | **poems** | and **songs** of **Robert Burns** | and **features** the **singing** of **Jean Redpath** and **Rod Patterson**

Example 2:

Data as annotated:

that's The **Miller's Reel** | **especially** for **Burns night** | **here** on **Radio 4**

Model output:

['U', 'U', 'S', 'S', 'S', 'U', 'S', 'S', 'S', 'U', 'S', 'S']

Predicted by the LSTM model:

that's The **Miller's Reel especially** for **Burns night here** on **Radio 4**

3.6. Error analysis and Discussion

If we look at the PB results (Table 3.2), we can see that varying the type of network didn't give a drastic improvement in the performance. These results are consistent with the findings made by Vadapalli et al., 2016, where they implemented RNN and LSTM, also varying the

deepness of the networks adding more layers and also increasing dimensions of word embeddings. Vapapalli et al. managed to achieve almost 93% of accuracy comparing to our 89%, however, this difference can be explained by the following factors:

- Number of predicted classes (2 in the work by Vadapalli and 3 in the present research). Table 3.1 shows that data distribution is very unbalanced, NB class being the largest and class BB taking the smallest part. However, we managed to achieve a 100% precision when predicting BB class by implementing bidirectional networks. This can also be a topic of the further research.
- The consistence of data. Vadapalli et al. used 3 different audiobooks training all three models separately. That means that their models were trained on only one speaker, i.e. they are speaker-dependent models. Our data is very inconsistent including several speakers and even interviews with several people talking.

Watts et al. 2011 also present their results using different evaluation metrics including precision, recall and F-score what makes it is for us to make a comparison. The difference in the precision can be again explained by the number of PB classes. However, by implementing bidirectional networks we managed to achieve 100% precision on major BB break.

Sentence stress models without any additional features show quite good performance ranging from 92% to 93% of accuracy. This

proves our hypothesis that unlike pause prediction task (multiclass prediction) sentence stress prediction task (binary classification) is not as complicated and can be successfully completed by a simple implementation of neural network models. The results show that a simple introduction of a DNN model allows to achieve 93% of accuracy. However, we managed to achieve 97% of accuracy on the sentence stress prediction task by implementing a bidirectional LSTM model with POS-tag information added as a feature.

Nevertheless, if we try to track the general tendency, POS-tags don't improve the performance of the models that much. We suppose, the reason lies in the using of word embeddings which to some extent contain additional information grouping different parts of speech closer to each other.

If we compare the accuracy of RNN model and biRNN on the pause prediction task, we will see that reading the data backwards gives an improvement of around 8-9% in model accuracy. That means that when speaking we think a couple of words ahead before saying them. But is it true for learners of foreign languages? Moreover, a bidirectional RNN shows surprisingly low results on sentence stress prediction tasks achieving only 81% of accuracy.

Below is the table presenting the results achieved by other researches interested in phrase break prediction task.

Table 3.7 Phrase break prediction comparison table

Paper	Corpus	Method	Result
Watts O., Yamagishi J., King S. Unsupervised continuous-valued word features for phrase-break prediction without a part-of-speech tagger. Interspeech, 2011.	Wall Street Journal section of the Penn Treebank	supervised prediction of PBs using classification trees (CART); the VSM features that have been acquired in an unsupervised fashion are used as predictor features.	Precision on the test set: B - 83.0, NB - 91.8; Recall on the test set: B – 73.0, NB – 95.3. F-score on the test set: B – 77.7, NB – 93.5.
O. Watts, S. Gangireddy, J. Yamagishi, S. King. Neural net word representation for phrase-break prediction without a part of speech tagger.		Shared projection feed-forward NN (unsupervised distributional pre-training): adding the word representations pre-trained on the LM task and keeping them fixed gives the best performance	Mean and standard deviation of F-scores (test set) of final systems is 78.36.
Vadapalli A., Prahallad K. Learning continuous-valued word representations for phrase break prediction. Interspeech, 2014.		Paper proposes a neural network dictionary learning architecture to induce task specific word representations, i.e., to derive word representations specific to phrase break prediction. Proposed architecture uses a **multilayer perceptron (MLP)** setup as a discriminative classifier. This allows to induce word features that are specific to the phrase break classification task.	Neural network learning with LSA dictionary + POS achieves 0.72 F-measure on the PB prediction task.
Vadapalli A., Gangashetty V S. An investigation of recurrent neural	Audiobooks by J. Austen: Emma Pride and	Use two different RNN architectures: (1) Elman RNN and (2) Long short term	F-measure: RNN – 92.55 on PAP

network architectures using word embeddings for phrase break prediction. Interspeech 2016, San Francisco, USA.	Prejudice Mansfield Park	memory (LSTM).	LSTM – 92.82 on PAP

Chapter 4

Conclusion

4.1. Summary

In the present work, we built two prediction models that carries out two different tasks:

- prediction of minor and major pauses in the text
- prediction of sentence stress

The experiment showed that implementation of different linear neural network models (DNN and simple RNN) didn't improve the performance that drastically showing only a slight improvement. However, the implementation of bidirectional networks on the pause prediction task allows us to predict sentence boundaries with 100% accuracy.

Sentence stress models without any additional features show quite good performance ranging from 92% to 93% of accuracy. This proves our hypothesis that unlike pause prediction task (multiclass prediction) sentence stress prediction task (binary classification) is not as complicated and can be successfully completed by a simple implementation of neural network models. The results show that a simple introduction of a DNN model allows to achieve 93% of accuracy.

When considering words independently of each other, in other words, when using DNN models, adding information about pauses is advisable, because it gives the model some additional information about the context. However, when using any kind of RNN, the context is already explored by the network, and to a much deeper extent. Therefore, even if information about annotated pauses contributes to some improvement, it is very insignificant.

The results of the experiment allow us to make a conclusion that the data itself is insufficient (5 hours of speech), unbalanced and includes different speakers each having some peculiarities.

Implementation of the pre-trained GloVe model allowed models to reduce the number of network parameters reducing the possibility of model overtraining and secondly it allowed to improve the quality of classification by using an effective model for embedding.

Bidirectional models proved to be effective providing data about right and left context of the target word. It is proved by the results of the experiment on both phrasing and sentence stress prediction tasks. The best performance on the phrase break prediction task with the accuracy equaling 0.89 (89%) was achieved by a bidirectional GRU with POS-tag information added as a feature. And the best performance on the sentence stress prediction task with the accuracy equaling **0.97** (97%) was achieved by the **bidirectional LSTM trained with the POS-tag information added as a feature.**

In other words, reading the data forwards and backwards at the same time improves the model accuracy. That means that when

speaking we think a couple of words ahead before saying them. The question is whether it is also true for learners of foreign languages. The mentioned issue may become the topic of further research.

4.2. Contribution

The main contribution of the present research may be seen in the comparison of the performance of neural network models on two different prosody prediction tasks: intonation phrases and sentence stress. According to the results of this comparison we can conclude that although both phenomena are related to prosody, sentence stress prediction can be seen as an easier task while pause prediction still remains quite challenging.

The purpose of the study was threefold. The first goal of the present work was to conduct a multi-class prediction task on intonation boundaries consisting of three classes: intermediate boundary, sentence boundary and the absence of boundary.

The second goal was to conduct a sentence stress prediction task and demonstrate that the implementation of neural network models without additionally extracted features will allow to achieve a relatively high performance. Both prediction tasks were performed based on the word embedding model together with neural network architectures.

The main hypothesis was that the implementation of bidirectional neural networks will help increase the accuracy of pause prediction and drastically improve the overall performance. The second hypothesis was that a pre-trained word embedding model in

combination with a neural network architecture will allow to achieve good performance on sentence stress prediction task.

The third and the last goal was to examine and compare the performance of different neural network models on the prediction tasks mentioned above.

4.3. Future work

The present research has several limitations. The first one is quite inconsistent data containing recordings made by different speakers what may have influenced the experimental results as every person has their own speaking style. Moreover, some of the recordings were in the form of a radio interview.

The second limitation is the size of the data. The corpus consists of only 5 hours of speech. Because of the small size, we had to implement different regularization techniques like dropout to prevent our model from overtraining and saturation.

The third one is lack of different levels of annotations. To get POS-tags for our data, we had to implement a NLTK POS-tagger which is far from being 100% accurate what also may have influenced the overall performance of the models.

However, this research may further provide a background and become a starting point for building a CALL system that helps non-native speakers sound more native-like working on their intonation.

References

1. Abercrombie, D. (1964) Syllable quantity and enclitics in English, In Honour of Daniel Jones, (Abercrombie, D., Fry, D. F., McCarthy, P .A. D., Scott, N. C. & Trim, J. M. L., eds.). London: Longmans.
2. Altman, G., and Carter, D. (1989). Lexical stress and lexical discriminability: stressed syllables are more informative, but why? Comput. Speech Lang. 3, 265–275. doi: 10.1016/0885-2308(89)90022-3
3. Bachenko, J., & Fitzpatrick, E. (1990). A Computational Grammar of Discourse-Neutral Prosodic Phrasing in English. Computational Linguistics, 16, 155-170.
4. Bengio, Y., Ducharme, R., Vincent, P. and Jauvin, C. (2001, 2003) A Neural Probabilistic Language Model. NIPS'2000 13:933-938, and revised in J. Machine Learning Research (2003) 3:1137-1155.
5. Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python.
6. Calhoun, S. (2010). How does informativeness affect prosodic prominence? Lang. Cogn. Process. 25, 1099–1140. doi: 10.1080/01690965.2010.491682
7. Chomsky, N. Aspects of the Theory of Syntax. MIT University Press, 1965.

8. Chung, J., Gülçehre, Ç., Cho, K., & Bengio, Y. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *CoRR, abs/1412.3555*.
9. Clifton, C. Jr., Carlson, K., and Frazier, L. (2002). Informative prosodic boundaries. Lang. Speech 45, 87–114. doi: 10.1177/00238309020450020101
10. Cummins, F., and Port, R. F. (1998). Rhythmic constraints on stress timing in English. J. Phon. 26, 145–171. doi: 10.1006/jpho.1998.0070
11. Cutler, A. (1976). Phoneme-monitoring reaction time as a function of intonation contour. Percept. Psychophys. 20, 55–60. doi: 10.3758/BF03198706
12. Cutler, A., and Fodor, J. A. (1979). Semantic focus and sentence comprehension. Cognition 7, 49–59. doi: 10.1016/0010-0277(79)90010-6
13. Cutler, A., and Foss, D. J. (1977). On the role of sentence stress in sentence processing. Lang. Speech 20, 1–10. doi: 10.1177/002383097702000101
14. Deng, L., Hinton, G., and Kingsbury, B. (2013). New types of deep neural network learning for speech recognition and related applications: An overview. In Proceedings of International Conference on Acoustics Speech and Signal Processing (ICASSP).
15. Deng, L., and Yu, D. (2014). Deep Learning: Methods and Applications. Now Publishers Inc., Hanover, MA, USA.

16. Dong Yu, Li Deng (2014). Automatic Speech Recognition: A Deep Learning Approach, Springer Publishing Company, Incorporated.
17. Ferreira, F. (1993). The creation of prosody during sentence production. Psychol. Rev. 100, 233–253. doi: 10.1037/0033-295X.100.2.233
18. Ferreira, F. (2007). Prosody and performance in language production. Lang. Cogn. Process. 22, 1151–1177. doi: 10.1080/01690960701461293
19. Józefowicz, R., Zaremba, W., & Sutskever, I. (2015). An Empirical Exploration of Recurrent Network Architectures. *ICML*.
20. Goldman-Eisler, F. (1961). The distribution of pause durations in speech. Language and Speech, 4(4):232–237.
21. Goodfellow, I., Bengio, Y., and Courville, A. (2016). Deep Learning. The MIT Press.
22. Greff, K., Srivastava, R. K., Koutn´ık, J., Steunebrink, B. R., Schmidhuber, J. (2015). "LSTM: A search space odyssey," CoRR, vol. abs/1503.04069, [Online]. Available: http://arxiv.org/abs/1503.04069
23. Hayes, B. 1995. Metrical stress theory, Chicago, IL: University of Chicago Press.
24. Hinton, G., Osindero, S., and The, Y. (2006). A fast learning algorithm for deep belief nets. Neural Computation, 18(7), pp. 1527–1554.

25. Hinton, G., and Salakhutdinov, R. (2006). Reducing the dimensionality of data with neural networks. Science, 313(5786), pp. 504–507.
26. Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T., and Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition. IEEE Signal Processing Magazine, 29(6), pp. 82–97.
27. Hochreiter, S., and Schmidhuber, J. (1997). Long Short-Term Memory. Neural Comput., 9(8), pp. 1735-1780, DOI=http://dx.doi.org/10.1162/neco.1997.9.8.1735
28. Jaitly, N., Hinton, G. (2011). Learning a better representation of speech sound waves using restricted Boltzmann machines. In Proceedings of International Conference on Acoustics Speech and Signal Processing (ICASSP).
29. Jassem, W. (1952). Stress in Modern English. Bulletin de la Société Linguistique Polonaise XII: 189-194.
30. Jurafsky, D., Martin, J. H. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Prentice-Hall, 2000.
31. Karlik, B., Vehbi Olgac, A. (2011). Performance Analysis of Various Activation Functions in Generalized MLP Architectures of Neural Networks. International Journal of Artificial Intelligence and Expert Systems. 1. 111-122.

32. Lee Gary Geunbae et al. (2017). Automatic sentence stress feedback for non-native English learners. Computer Speech and Language 41 (2017) 29–42.
33. Liberman Mark, Prince Alan (1977). On Stress and Linguistic Rhythm. Linguistic Inquiry, Vol. 8, No. 2 (Spring, 1977), pp. 249-336.
34. Lu, J., Wang, R. & De Silva, L.C (2012). Automatic stress exaggeration by prosody modification to assist language learners perceive sentence stress. Int J Speech Technol 15: 87. https://doi.org/10.1007/s10772-011-9124-2
35. Mass, A., Hannun, A., Ng., A. (2013). Rectifier nonlinearities improve neural network acoustic models. International Conference on Machine Learning (ICML) Workshop on Deep Learning for Audiom, Speech, and Language Processing, 2013.
36. Rosenberg et al. Modeling phrasing and prominence using deep recurrent learning, 2015.
37. Parlikar Alok (2013). Style-Specific Phrasing in Speech Synthesis. Carnegie Mellon University PhD Thesis.
38. Selkirk, E. 1995. Sentence prosody: Intonation, stress and phrasing. In The handbook of phonological theory, Edited by: Goldsmith, J. 550–569. Cambridge, MA and Oxford: Blackwell.
39. Selkirk, E.O. 1984. Phonology and syntax: The relation between sound and structure, Cambridge, MA: MIT Press.

40. Silverman, K.E., Beckman, M.E., Pitrelli, J.F., Ostendorf, M., Wightman, C.W., Price, P., Pierrehumbert, J.B., & Hirschberg, J. (1992). TOBI: a standard for labeling English prosody. ICSLP.
41. Solé Sabater, M. J. (1991). Stress and Rhythm in English. Revista Alicantina de Estudios Ingleses, 4, 145–162.
42. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural network from overfitting. Journal of Machine Learning Research, 15, 1929 – 1958.
43. Taylor, P. (2009). Text-to-Speech Synthesis. Cambridge: Cambridge University Press. doi:10.1017/CBO9780511816338
44. Vadapalli et al. (2014). Learning continuous-valued word representations for phrase break prediction. Interspeech.
45. Vadapalli et al. (2016). An investigation of RNN architectures using word embeddings for phrase break prediction. Interspeech.
46. Wang, Q., Hirschberg, J. (1992). Automatic classification of intonational phrase boundaries. Computer Speech & Language. 6. 175-196. 10.1016/0885-2308(92)90025-Y.
47. Warren, P. 1999. Prosody and language processing. In Language processing, Edited by: Garrod, S. and Pickering, M. 155–188. Hove, UK: Psychology Press Ltd.
48. Watts O. et al. Unsupervised continuous-valued word features for phrase-break prediction without a POS tagger. Interspeech, 2011.
49. Watts O. et al. Neural network representation for phrase-break prediction without a part of speech tagger, 2014.

50. Zubizarreta, M. L. 1998. Prosody, focus, and word order, Cambridge, MA: MIT Press.

APPENDIX 1.

Table 1. A part-of-speech tagset

Tag	Meaning	Examples
ADJ	adjective	*new, good, high, special, big, local*
ADV	adverb	*really, already, still, early, now*
CNJ	conjunction	*and, or, but, if, while, although*
DET	determiner	*the, a, some, most, every, no*
EX	existential	*there, there's*
FW	foreign word	*dolce, ersatz, esprit, quo, maitre*
MOD	modal verb	*will, can, would, may, must, should*
N	noun	*year, home, costs, time, education*
NP	proper noun	*Alison, Africa, April, Washington*
NUM	number	*twenty-four, fourth, 1991, 14:24*
PRO	pronoun	*he, their, her, its, my, I, us*
P	preposition	*on, of, at, with, by, into, under*
TO	the word *to*	*to*
UH	interjection	*ah, bang, ha, whee, hmpf, oops*
V	verb	*is, has, get, do, make, see, run*
VD	past tense	*said, took, told, made, asked*
VG	present participle	*making, going, playing, working*
VN	past participle	*given, taken, begun, sung*
WH	*wh* determiner	*who, which, when, what, where, how*

APPENDIX 2.

Table 2. Confusion matrix (sentence stress prediction)

Model			Actual class	
			S	U
DNN	Predicted class	S	2439	198
		U	145	2108
RNN	Predicted class	S	2452	185
		U	201	2052
GRU	Predicted class	S	2449	188
		U	166	2087
LSTM	Predicted class	S	2451	186
		U	174	2079
DNN with annotated PB	Predicted class	S	2462	175
		U	178	2075
RNN with annotated PB	Predicted class	S	2464	173
		U	223	2030
GRU with annotated PB	Predicted class	S	2452	185
		U	158	2095
LSTM with annotated PB	Predicted class	S	2468	169
		U	135	2118
biRNN	Predicted class	S	2185	452
		U	431	1822
biGRU	Predicted class	S	2454	183
		U	172	2081
biLSTM	Predicted class	S	2462	175
		U	164	2089
biRNN with annotated PB	Predicted class	S	2380	257
		U	664	1589
biGRU with annotated PB	Predicted class	S	2478	159
		U	198	2055
biLSTM with annotated PB	Predicted class	S	2451	186
		U	145	2108

요약 (국문 초록)

운율은 최근 몇년 사이에 광범위하게 연구되었다. 언어학에서, 운율은 개별 음성 세그먼트 (모음 및 자음) 가 아니라 음절의 특성과 더 큰 음성 단위인 말의 요소를 다룬다. 이러한 요소들은 억양, 성조, 강세 및 리듬과 같은 언어학적 기능에 기여한다. 운율은 화자 또는 발화의 다양한 특징, 즉 화자의 감정상태, 발언의 형태 (서술, 질문 또는 명령) , 또는 문법이나 어휘에 의해 인코딩될 수 없는 언어의 다른 요소들을 반영한다.

본 연구는 운율의 요소인 휴지부 및 문장강세에 집중한다. 휴지뷰를 정의하는 것이 쉽지 않다. 우선 휴지부는 호흡과 관련된 매우 자연스러운 현상이다. 또한 무음휴지부와 화자가 주저하지만 지속적으로 소리를 낼 때 발생하는 " 채워진" 휴지부를 구별해야 한다.

문장강세는 운율단위 내에서 적용되는 강세패턴을 나타낸다. 본 현상은 특정 언어의 강세패턴을 포함할 수도 있지만, 상대적 중요성을 가진 특정 단어에 중점을 두는 것도 포함할 수도 있다.

본 연구의 목표는 다음과 같다. 다중 분류 작업 수행함으로써 주 휴지부, 보조 휴지부 및 휴지부가 발생하지 않는 구간을 예측하는 모델을 만드는 것이다.

두 번째 목표는 문장 강세 예측 모델을 구성해서 언어학적 정보가 포한되어 있는 추가 피처를 반영 안 하는 신경망 모델의 구현이 비교적 높은 성능을 달성할 수 있음을 입증하는 것이다. 모든 예측 모델들은 단어 임베딩 모델

(word embedding)과 다양한 신경망 구조(neural networks) 기반으로 학습된 것이다.

주 가설은 bidirectional neural networks 구현이 휴지부 예측 모델의 정확도를 높이고 전반적인 성능을 향상시키는데 도움이 된다는 것이다. 두 번째 가설은 사전 훈련된 (pre-trained) 단어 임베딩 모델을 신경망 구조와 결합하여 문장 강세 예측 작업에서 좋은 성능을 얻을 수 있다는 것이다.

세 번째와 마지막 목표는 앞에서 언급된 예측 작업에 대한 다양한 신경망 모델의 성능을 검사하고 비교하는 것이다.

키워드 : 운율예측, 휴지부, 문장강세, 딥러닝, 신경망.
학번 : 2015 - 23282