공학 석사 학위논문

Vowel Duration and Fundamental Frequency Prediction for Automatic Transplantation of Native English Prosody onto Korean-accented Speech

자동 운율 복제를 위한 모음 길이와 기본 주파수 예측

2018 년 8 월

서울대학교 대학원

협동과정 인지과학전공

Matsvei Sabaleuski

Abstract

Vowel Duration and Fundamental Frequency Prediction for Automatic Transplantation of Native English Prosody onto Korean-accented Speech

Matsvei Sabaleuski

Interdisciplinary Program in Cognitive Science

The Graduate School

Seoul National University

The use of computers to help people improve their pronunciation skills of a foreign language has rapidly increased in the last decades. Majority of such Computer-Assisted Pronunciation Training (CAPT) systems have been focused on teaching correct pronunciation of segments only, however, while prosody received much less attention. One of the new approaches to prosody training is self-imitation learning. Prosodic features from a native utterance are transplanted onto learner's own speech, and given back as corrective feedback. The main drawback is that this technique requires two identical sets of native and non-native utterances, which makes its actual implementation cumbersome and inflexible.

As a preliminary research towards developing a new method of prosody transplantation, the first part of the study surveys previous related works and points out their advantages and drawbacks. We also compare prosodic

systems of Korean and English, point out major areas of mistakes that Korean learners of English tend to do, and then we analyze acoustic features that this mistakes are correlated with. We suggest that transplantation of vowel duration and fundamental frequency will be the most effective for self-imitation learning by Korean speakers of English.

The second part of this study introduces a new proposed model for prosody transplantation. Instead of transplanting acoustic values from a pre-recorded utterance, we suggest to use a deep neural network (DNN) based system to predict them instead. Three different models are built and described: baseline recurrent neural network (RNN), long short-term memory (LSTM) model and gated recurrent unit (GRU) model. The models were trained on Boston University Radio Speech Corpus, using a minimal set of relevant input features. The models were compared with each other and as well as with state-of-the-art prosody prediction systems from speech synthesis research.

Implementation of the proposed prediction model in automatic prosody transplantation is described and the results are analyzed. A perceptual evaluation by native speakers was carried out. Accentedness and comprehensibility ratings of modified and original non-native utterances were compared with each other. This study lays the groundwork for a fully automatic self-imitation prosody training system and its results can be used to help Korean learners master problematic areas of English prosody, such as sentence stress.

Keywords: Computer-assisted pronunciation training, Korean-accented English Prosody, Prosody transplantation, Prosody prediction, Deep Neural Networks (DNN)

Student Number: 2015-23283

Contents

List of Tables

List of Figures

Chapter 1. Introduction

1.1 Background

Computer-assisted pronunciation training (CAPT) has been attracting significant research over the past decades, with the growing number of people preferring to learn foreign languages online compared to a traditional classroom environment (Jilka and Mohler, 1998; Sundstrom, 1998; Murray, 1999; Probst et al., 2002; Eskenazi, 2009; Felps et al., 2009; De Meo et al., 2013; Pellegrino and Vigliano, 2015). CAPT systems benefit learners in various aspects: they are able to offer individualized one-on-one tutoring regardless of constraints in time and place, allowing students to learn at their own preferred pace. A study by Murray (1999) shows that users are more comfortable practicing pronunciation in a private setting, where they can avoid anxiety and embarrassment. Moreover, the feedback generated in CAPT system has an advantage in that it can provide feedback, specific to the native language of a learner, whereas the instructor in conventional classroom environment cannot address and is not necessarily aware of L1 diversities. This is an important advantage because learners' mother tongue influences the target language production in foreign language learning, indicating that different types of feedbacks are required for each L1; and addressing the L1 influence is a strength of a CAPT system.

Typically, during a CAPT session, a student listens to an utterance pronounced by a native speaker (teacher) and tries to imitate it as closely as possible. If the student makes a mistake, corrective feedback is given back, typically the same sentence, but recorded with a native speaker voice. Recent research has shown the importance of the student/teacher voice similarity on the effectiveness of such feedback (Watson and Kewley-Port, 1989; Jilka and

Mohler, 1998; Sundstrom, 1998). It was found that the closer teacher's voice resembles the student's voice in terms of fundamental frequency (F0) and articulation rate, the better result can be achieved (Probst et al., 2002; Felps et al., 2009). Such ideal feedback voice is often called 'golden voice', and is considered to be the voice of the student himself. That can be achieved by performing accent conversion. The rationale is that, by stripping away information that is only related to the teacher's voice quality, accent conversion makes it easier for students to perceive differences between their accented utterances and their ideal accent-free counterparts.

Pronunciation training, during which a learner receives feedback in his/her own voice, but with a native accent, is often called self-imitation learning. Studies have shown that self-imitation learning can be rather effective, especially when trying to improve accentedness, comprehensibility or intelligibility of speech (De Meo et al., 2013; Pellegrino and Vigliano, 2015). As such it has great potential when the aim is to improve the mastery of prosody of the foreign language. So far, the majority of CAPT systems have being focused on teaching correct pronunciation of segments. Suprasegmentals, on the other hand, have received much less attention. Here by suprasegmentals, or prosody, we understand a level of linguistic representation at which the acoustic-phonetic properties of an utterance vary independently of its lexical items. Prosody deals with suprasegmental features of speech, those that are bigger than a phoneme, which are typically syllables, words, phrases, etc. It encompasses a range of phenomena: emphasis, pitch accents, intonational breaks, rhythm, intonation and others. But intonation, speech rate and other elements of prosody play an important role in not only global accent overall, but in basic communication – comprehensibility and intelligibility of speech (Boula de Mareuil et al., 2004; Pellegrino, 2012; Rognoni and Grazia Busa, 2013; Sereno et al., 2016).

As such, self-imitation learning can be of great help to Korean learners of English. In English, every word has one or more lexical stress depending on the structure of the word and the number of syllables, but not all word stresses are phonetically realized in an utterance. Stress imposed on the utterance level has been traditionally called 'sentence stress'. Korean learners of English typically have great difficulties with rhythm and fluency of their speech (Kim and Flynn, 2004; Um, 2004; Kim, 2005; Lee et al., 2006; Jun, 2009; Kang et al., 2012; Yoo, 2012). Low proficiency learners tend to place sentence stress on most of the words in a sentence, even on function words. They tend to use strong vowels even in unstressed syllables, giving the impression of syllable-timed rhythm to native English listeners. Self-imitation prosody training could potentially help them to eliminate these problems.

1.2 Research Objective

To achieve 'golden voice' for prosody self-imitation training, a prosody transplantation technique must be applied. The most commonly used method is based on the pitch synchronous overlap and add (PSOLA) technique (Charpentier and Stella, 1986; Moulines and Charpentier, 1990; Valbret et at., 1992). It allows to manipulate and change segmental duration and F0 of the original utterance to match that pronounced by a native speaker.

This transplantation method has a number of restrictions, though. The major drawback is that a set of two identical recorded utterances (by learner and native speaker) is required; both must be annotated and must contain the exact same number of segments. This alone imposes two challenges for its real learning environment implementation. First, recording of teaching material with a native speaker can be cost-heavy and time-consuming. Second, a pronunciation mistake by learner (mispronunciation, omission, deletion, etc.) can prevent the system from working properly, due to the different number of

segments. Moreover, the results of such transplantation greatly depend on selection of a good native speaker voice. Additionally, voice of the same person can vary greatly between different recordings of the same utterance.

These problems explain why there has been little actual application of the prosody transplantation technique so far, and at the same time it motivates this study. Our hypothesis is that prosody transplantation can be carried out using predicted acoustic values (duration and F0), instead of copying them from pre-recorded utterances. The prediction model can have similar architecture to those, used for prosody prediction in speech synthesis. Recently, that topic has attracted considerable attention, with new deep learning-based methods (Fernandez et al., 2013; Ding et al., 2015; Su et al., 2016; Bernardi and Themistocleous, 2017). By doing that the entire prosody transplantation procedure can be potentially automated.

The research objectives are as follows.

Research objective 1. So far, prosody transplantation research concentrated on universal transplantation of prosodic parameters to all the segments within the utterance. But such approach does not really tell the learner what kind of mistake and where was made. As such we aim to investigate the typical prosodic mistakes of target learners (L1 Korean speakers of English), and propose a selective transplantation strategy that should give them a more valuable feedback.

Research objective 2. The next objective is to determine what kind of input features will be most effective for a potential prosody prediction model. More features is not always better, as not all of them might be contributing to the results. Additionally, automatic application of prosody transplantation imposes certain constraints: only automatically obtainable features are preferable.

Research objective 3. The main objective is to introduce a prosody

prediction model for automatic prosody transplantation. We will build a number of model, that will primarily rely on input that can be obtained automatically and evaluate its performance.

Research objective 4. The final objective is to show how such prediction model can be applied to automatic prosody transplantation and evaluate its performance.

To achieve these objectives, the following will be undertaken:

- We will build vowel duration and fundamental frequency prediction system based on recurrent neural network (RNN) architectures in Python[1] programming language, using TensorFlow[2] library for deep learning.
- The models will be trained on data from a native American English corpus – the Boston University Radio Speech Corpus[3].
- Predicted values will be used in a prosody transplantation experiment on L1 Korean English data, using Praat[4] and Python's Promo[5] library.
- The results of the prosody transplantation will be evaluated in a perceptual listening experiment.

The results can be used to help Korean learners master such problematic areas of English prosody for them, like sentence stress and rhythm. The findings can be also extended to other language pairs. Potentially, the proposed model can be integrated into any existing prosody oriented CAPT

[1] Python software is available from https://www.python.org/ ..
[2] Tensorflow software is available from https://www.tensorflow.org/ and is introduced in Abadi et al. (2016).
[3] BURSC is available from https://catalog.ldc.upenn.edu/LDC96S36 and is described in Ostendorf et al. (1996).
[4] Praat software is available from http://www.praat.org/ and described in Boersma (2011).
[5] Prosody-Morphing (ProMo) library is available from https://github.com/timmahrt/promo .

system. One area of research that requires further study is perceptual evaluation of transplantation results.

1.3 Research Outline

This study focused on the answering the before mentioned research questions. The results of the following research question will be defined in Chapter 4 after being discussed in each chapter.

Chapter 2 surveys previous studies on self-imitation learning and prosody transplantation. We compare prosodic systems of English and Korean and point out typical mistakes, that Korean learners are prone to do. It also introduces deep learning and its application in prosody prediction research.

In Chapter 3 we describe the proposed models' architecture and input data, build two separate models for vowel duration and fundamental frequency prediction respectively, and evaluate the results.

In Chapter 4 we also show how predicted values from the proposed model can be implemented in an automatic prosody transplantation experiment. We also investigate the results of such transplantation using perceptual listening evaluation.

Chapter 5 concludes the thesis, summarizes the results, points out its limitations as well as areas for future study.

Chapter 2. Related Works

2.1 Self-imitation Prosody Training

During self-imitation learning, feedback is given back to the student in his/her own voice, but with the prosodic features of a native speaker. It is claimed that such type of learning can provide more motivation that any of the more traditional approaches (Probst et al., 2002; Yoon, 2007; Felps et al., 2009). A typical self-imitation training session can consist of the following: self-imitation learning "software plays the target sentence uttered by a native speaker, records what the language learner repeats, imposes only the prosodic features of the native speaker onto the learner's utterance, and plays back the learner's utterance with the native speaker's prosody, demonstrating to the second language learners that they could "speak" like the native speaker" (Yoon, 2007).

Various research has recently appeared in the field of L2 learning that employ transforming foreign-accented speech into its native-accented counterpart. Either segmental, prosodic or both transformations cab be done, although prosodic transplantation is more common. Research so far has been done for a number of different foreign accents of English: Spanish, Italian, French, Indian, Korean, Chinese, and Japanese accents (Felpts et al., 2009; Zhao et al., 2012; Aryal et al., 2013; Park, 2013; Rognoni and Grazia Busa, 2013; Sereno et al., 2016). Other investigates language pairs are: Spanish, Chinese and Japanese-accented Italian (De Meo et al, 2013; Pellegrino and Vigliano, 2015; Sereno et al., 2016), Italian-accented Spanish (Boula de Mareuil et al., 2004), French-accented German (Jugler et al., 2016), and Polish-accented French (Kaglik and Boula de Mareuil, 2010).

Table 2.1. Review of previous research on self-imitation prosody learning.

Author	Language	Method	Features	Data	Evaluation
Kaglik, et al. (2010).	Polish L2 French	PSOLA, text-to-speech synthesis	Duration, F0	Recorded and synthesized speech	Perceptual evaluation
Bonneau, et al. (2011)	French L2 English	TD-PSOLA	Duration, F0	Recorded speech	Training session and parametric comparison
Felps, et al. (2009)	Indian L2 English	FD-PSOLA	Duration, F0	Recorded speech	Perceptual evaluation
De Meo, et al. (2013)	Chinese L2 Italian	PSOLA	Duration, F0, intensity, articulation rate	Recorded speech	Training session and perceptual evaluation
Pettorino, et al. (2012)	Italian L2 English	PSOLA	Duration, F0, intensity	Recorded speech	Spectrographic comparison
Yoon (2007)	Korean L2 English	TD-PSOLA	Duration, F0, intensity	Recorded speech	Spectrographic comparison
Pellegrino, et al. (2015)	Japanese L2 Italian	PSOLA	Duration, F0, intensity	Recorded speech	Training session and perceptual evaluation
Zhao, et al. (2012)	Chinese L2 English	LP-PSOLA	Duration, F0	Recorded speech	Speech recognition score
Sereno, et al. (2014)	Korean L2 English	PSOLA	Duration, F0	Recorded speech	Perceptual evaluation
Boula de Mareul, et al. (2004)	Spanish L2 Italian, Italian L2 Spanish	TD-PSOLA	Duration, F0	Recorded and synthesized speech	Perceptual evaluation
Jugler, et al. (2016)	French L2 German	TD-PSOLA	Duration, F0	Recorded speech	Perceptual evaluation
Park (2012)	Korean, Japanese and Chinese L2 English	TD-PSOLA	Duration, F0	Recorded speech	Perceptual evaluation
Rognoni, et al (2014)	Italian L2 English	TD-PSOLA	Duration, F0	Recorded speech	Perceptual evaluation

In Table 2.1 we summarize and analyse prosody transplantation studies done so far. For the comparison we used the following relevant criteria: language pair used for transplantation, transplanted prosodic features, method of transplantation, what kind of data was used, and how the results were evaluated. Overall, there are two common sub-areas of research: perceptual effects of prosody transplantation on accentedness (or similar ratings) and effects of self-imitation learning on language proficiency.

2.1.1 Prosody Transplantation Methods

Prosody transplantation can be achieved by employing a speech modification technique. The intonation and duration manipulation of speech signal have been a subject of great interest and several methods have been proposed to address this problem, with pitch synchronous overlap-add (PSOLA) being the most popular one. PSOLA allows to change the duration or shift the pitch of speech signals. Its main advantages are simplicity of the algorithm itself, rather good quality and fast execution speed.

The PSOLA algorithm involves the following three steps: “an analysis of the original speech waveform in order to produce an intermediate non-parametric representation of the signal, modifications brought to this intermediate representation, and finally the synthesis of the modified signal from the modified intermediate representation” (Moulines and Charpentier, 1990, p.2).

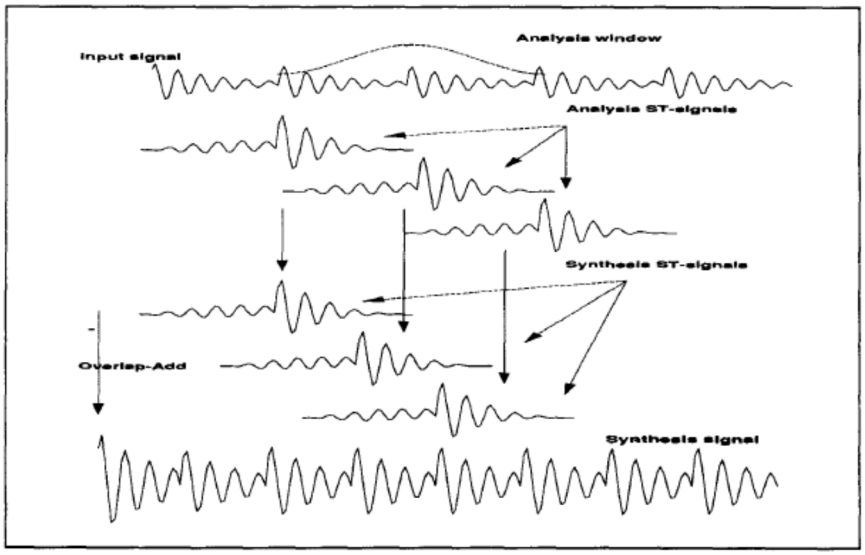


Figure 2.2. PSOLA prosodic modification framework. Pitch-scale modification. It is achieved by modifying the time delay between pitch-marks (Valbert et al., 1992).

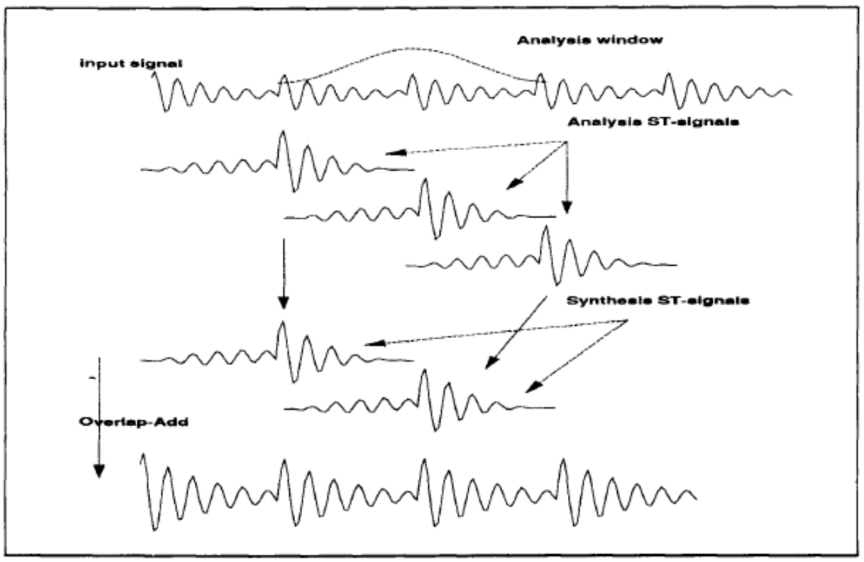


Figure 2.1. PSOLA prosodic modification framework. Time-scaling operation, aimed at speeding up the speech signal. It is achieved by selective elimination of the analysis ST-signals (Valbert et al., 1992)

The first step (analysis) consists in decomposing the excitation waveform $x(n)$ into a sequence of short-term (ST) signals $x_m(n)$, synchronized with the local pitch-period. These ST-signals are obtained by multiplying the signal by

a sequence of pitch-synchronous analysis windows $h_m\ (n)$, usually of the

$$x_m(n) = h_m\ (t_m - n)\ x(n).$$

Hanning type:

The windows are centred around successive instants called pitch-marks t_m, which are set at a pitch-synchronous rate on the voiced portions of the signal and at a constant rate on the unvoiced portions. The windows are always longer than one single pitch period, so that neighbouring ST signals overlap with each other. Their length is proportional to the local pitch period, with the proportionality factor μ lying between 2 and 4 (2 for low-pitch male voices, 3 for high-pitch female voices).

The second step is modification. The sequence of analysis ST signals $x_m(n)$ is converted into a modified stream of synthesis ST-signals, synchronized on a new set of synthesis pitch-marks. These synthesis pitch-marks are determined in order to comply with the desired prosodic modifications. Such a conversion involves three basic operations: a modification of the number of ST-signals, a modification of the delays between the ST-signals, and possibly, a modification of the waveform of each individual ST-signal. An excitation signal with modified pitch-scale and time-scale is then obtained by overlap-adding the stream of synthesis short-term signals.

In the Time-Domain PSOLA (TD-PSOLA) approach, the synthesis ST-signals are obtained by simply copying a version of the corresponding analysis signal, so that the algorithm consists in selecting a certain number of analysis ST-signals $x_m(n)$ and translating them by the sequence of delays $\delta_q = \tilde{t}_q - t_m$:

$$\tilde{x}_q(n) = x_m(n - \delta_q)$$

In the Frequency-Domain PSOLA (FD-PSOLA) approach, the synthesis ST-signals are obtained by a frequency-domain transformation of the translated signal.

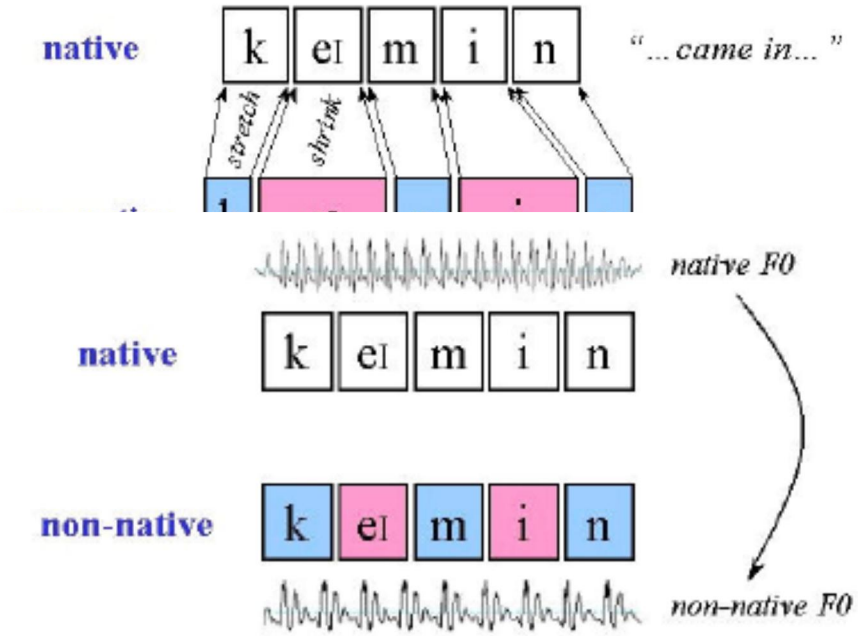


Figure 2.4. Native F0 imposition. Taken from Yoon (2007).

The last stage is to synthesize the signal: this is done by filtering the modified excitation signal by synthesis filters synchronized with the synthesis pitch-marks, and derived from the analysis filters through a simple interpolation procedure. The process is illustrated in Figure 2.1. and Figure 2.2., showing the two simple cases of time-scaling and pitch-scaling.

The most commonly used PSOLA method is time-domain modification (TD-PSOLA), followed by frequency domain modification (FD-PSOLA), and linear prediction modification (LP-PSOLA). Time domain approach is known to be more advantageous, because it requires much lower computational effort and provides good quality over a moderate transformation scale and is relatively fast (Valbret et al., 1992). The main drawback is that time-scaling by up by factor greater than two (twice longer or shorter) can cause noticeable

degradation in quality. If corresponding segments in native and original non-native utterance differ from each other more than that, duration distortion can be severe. As a result, might need to ask native speakers to decrease their rate of speech.

Yoon was one of the first to suggest using TD-PSOLA for self-imitation learning prosody transplantation (Yoon, 2007). The transplantation was done in the following stages. First, segment alignment for both native and non-native utterances was performed (see Figure 2.3). After that, by implementing PSOLA in Praat, non-native segment's length was adjusted to resemble that of the native ones. Then intensity contour was transplanted using a Praat script: original contour was mathematically 'neutralized' and native intensity contour was imposed instead. And finally F0 contour of the non-native utterance was replaced with F0 contour of a the native one (see Figure 2.4). Additionally, utterances with only one or two acoustic features were transplanted as well. By conducting a spectrographic comparison, it was determined that duration and F0 modification, as well as F0 only modification yield results, resembling the original native prosody the most.

The obtained modified utterances were compared spectrographically. After application of all 3 techniques two utterances became almost identical prosodically, although there were present some sub-segmental variations. To determine how each of these features contributes to accent reduction, the author also compared utterances in which only 1 or 2 of these features were cloned. F0 contour only and duration and F0 contour modifications seemed to resemble native prosody the most.

Although the results of the transplantation were deemed satisfactory, the experiment has a number of limitations. Only spectrographically evaluation was performed, though perceptual evaluation by native speakers might be considered a more reliable method. Additionally, transplantation was applied

only on segmental level, using syllables or phoneme sub-segments might give rather different results. Finally, transplantation can be applied selectively, when only predetermined target segments will receive modification.

Nevertheless, PSOLA is one of the easiest methods to implement, and has been widely used in prosody transplantation research. TD-PSOLA implemented in Praat is the most widely used method, probably due to its relatively easy implementation and good speed. Interestingly, only Zhao et al. tried to experiment with selective prosodic transplantation (Zhao et al., 2012). In their research both native and foreign-accented utterances were labelled according to ToBI[6] system, and only prosodic parameters in mismatched segments were replaced. The potential advantage of selective modification, is that it allows the learner to see exactly which area was mispronounced, thus leading to a greater educational effect. We consider the research in selective transplantation so far rather lacking and requiring additional investigation.

Few researchers tried to employ other techniques. Aryal et al. (2015) proposed to use articulatory synthesis, while Felps et al. (2012) used STRAIGT vocoder to synthesize native-like prosody and impose it onto segmental information. Both approaches are much more computationally heavy. Do not allow to transplant different prosodic features in different combinations.

2.1.2 Effects of Prosody Transplantation on Accentedness Rating

[6] Tones and break indices (ToBI) is a set of conventions for transcribing and annotating the prosody of speech (Silverman et al., 1992).

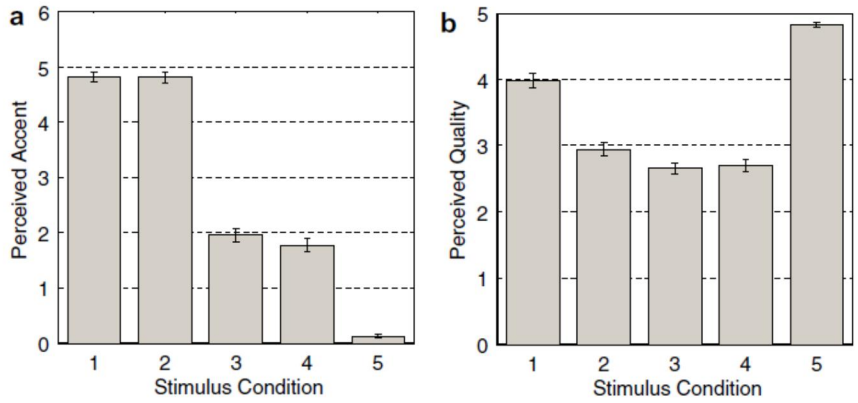


Figure 2.5. Accent and speech quality ratings for each transplantation type. (1 = foreign speaker, 2 = prosodic transformation, 3 = segmental transformation, 4 = prosodic and segmental transformations, 5 = native speaker). Taken from Felps et al. (2009).

One of the main areas of research is concentrated on investigating the role of prosody in accentedness, intelligibility, comprehensibility or similar proficiency-related ratings. A typical experiment would consist of transplanting native prosody onto non-native speech, and then comparing it with the original native and non-native utterances, or other types of modifications (e.g. when segmental information is transplanted instead). The comparison is commonly done by asking native speakers of the target foreign language to do perceptual evaluation of the utterances. The results of such experiments are rather mixed.

Jugler et al. (2016) transplanted fundamental frequency and phone duration from native German speakers and compared modified utterances with the original French-accented and native ones. Although prosodic transplantation did lead to the reduction in accentedness ratings, the introduced distortions also made the utterance to sound more unnatural.

Felps et al. (2009) compared the effects of prosody with that of segments and found opposite results. They transplanted native F0 and phone duration, as well as segmental information onto Indian-accented English speech. To test

the influence of prosody onto accentedness ratings, different transplantation models were implemented: prosodic only, segmental only, or both. Modified, as well as original foreign and native utterances were evaluated by native speakers. They found that prosodic transplantation lowers accentedness rating only slightly, while segmental and combined transplantation give the best result. At the same time, they also reduce the perceived quality of speech much to a greater extent (see Figure 2.5).

Similar results were obtained by Rognoni and Grazia Busa (2013). It was found out that segmental information has the strongest effect in accentedness perception. Although prosodic parameters seem to play a role as well: F0 and duration do have effect on the ratings, not only when transplanted together, but also when transplanted separately. In their experiment, duration showed a slightly greater effect, compared to F0.

Boula de Mareuil et al. (2004) investigated whether prosody transplantation affects natural and synthesized speech differently. For synthesized speech, prosody transplantation resulted in better accentedness ratings, while in case of natural speech, the role of segmental and prosodic information seems to be more balances.

A few experiments were also conducted for the Korean-accented English. Sereno et al. (2016) examined the relative impact of segments and intonation on accentedness, comprehensibility, and intelligibility. The research showed that segments have a significant effect on accentedness, comprehensibility, and intelligibility, but prosody only has an effect on intelligibility. Park (2013) conducted a similar study, with the main difference that they investigated the effects of prosody and segments across speakers of different L1 background, as well as evaluated by raters of different L1 background. Both L1 Korean and native American English speakers considered the segments to contribute more to the accentedness. Japanese raters, on the other hand, gave more

weight to the influence of prosodic parameters. These findings are significant in that they imply that, for native speakers of some languages, the difference in prosody could have a greater influence on the foreign-accentedness than the difference in segments, while for native speakers of other languages it might be the other way around.

Overall, we can conclude that prosody does influence language-proficiency rating, although to a much lesser degree, then segments. Intelligibility seems to be affected the most. Additionally, the background of raters might play a role as well, that is people of certain L1 backgrounds might give more weight to prosody than others.

2.1.3 Effects of Self-Imitation Learning on Proficiency Rating

The other major area of research was the effects of self-imitation learning on language proficiency. Pellegrino and Vigliano (2015) investigated the effects of self-imitation learning on Japanese students of Italian. All participants were recorded before the training, then during the session, each learner trained to mimic their utterances with native accent as many times as they need to approximate the model, and recording were made again after that. The results indicate that self-imitation promotes an improvement in learners' performances in terms of communicative effectiveness (ability to convey correct pragmatic function), but average rate of accentedness isn't affected significantly.

De Meo et al. (2013) conducted a similar research, but compared the results from two groups: one did self-imitation learning, while the other underwent conventional imitation training. Degree of foreign accent, improvements in intelligibility, and effectiveness of communication were measured to determine the success of each technique. Raters were asked to identify the speech act type of each utterance. Both teaching strategies promoted an

Table 2.2. Mean and standard deviation of prosodic elements. The score is the Euclidian distance between target values of Korean learners and a native English speaker (Yoon, 2011).

Mean (SD)	Before training			After training		
	Intonation	Intensity	Duration	Intonation	Intensity	Duration
Control group	517 (151)	213 (23)	329 (70)	504 (203)	223 (25)	323 (96)
Experimental group	524 (191)	223 (31)	419 (140)	420 (160)	233 (28)	343 (82)

improvement in performances; however, the self-imitation training proved to result in more accurate prosodic realizations. Both training sessions resulted in better intelligibility ratings, while self-imitation training resulted in slightly more utterances judged to sound native-like. These results suggest, that if the aim is to learn prosodic patterns of a foreign language, prosody transplantation might significantly improve the results.

Bonneau and Colotte (2011), instead of doing perceptual evaluation, took a more direct approach and investigate whether self-imitation learning can help French student master English lexical stress: F0 and duration ratio of stressed and unstressed vowels were compared between control and target group, that underwent self-imitation training and received visual feedback (spectrogram) as well. Results showed that the various kinds of feedback provided by the system enable French learners with a low production level to improve their realisations of English lexical accents more than (simple) auditory feedback. One limitation of this study, however, was that it is not clear whether positive effect was due to self-imitation learning or due to visually given feedback. As such, an additional experiment, whether these two types of feedback are separated, is required.

The effects of self-imitation learning were also investigated on Korean learners of English in Yoon (2011). A group of students underwent self-

imitation prosody training during a class twice a week (around 20 min.) for five and a half weeks, while another control group did imitation training only. The participants were recorded before and after he experiment and acoustic correlates of prosody were compared with those from native speaker (see Table 2.2). Self-imitation learning lead to improvements in terms of intonation (F0) and duration, while intensity was not affected; in case of control group, no improvements were observed.

The major drawback of all self-imitation studies done so far is that in all studies, but the one conducted by Yoon (2011), only a one-time training session was carried in all cases. It can be argued, that pronunciation, and prosody specifically, is not something that can be improved in a short amount of time. Nevertheless, this area of research seems promising, and the best results might be achieved when applied to low-proficiency students mastering various aspects of foreign language's prosodic system.

2.2 Prosody of Korean-accented English Speech

2.2.1 Prosodic Systems of Korean and English

What are the features of Korean accent of English? When it comes to prosody, one of the main factors might be the differences between the L1 and the L2 prosodic patterns. Abercrombie (1967) and others have proposed that all languages of the world are rhythmically isochronous and can be classified as either 'syllable-time' or 'stress-timed'. According to this, syllable duration should be more varied in stress-timed languages like English, compared to syllable-timed languages (like Korean).

Auer (1991) pointed further differences, such as that in stress-timed languages non-accented syllables are reduced compared to accented ones. The difference can be phonetic or phonological: non-accented syllables can have shorter vowels and/ or vowels can undergo neutralization process. Syllable-

timed languages shouldn't have such a distinction. Another difference is the natures of accent itself. In languages like English, accent has to be realized phonetically very distinctly and be strong. Accent is highly correlated with pitch movement and intensity, often supported by length. In syllable-timed languages, on the other hand, accent is realized weakly, if present, often only by duration, and its placement in the word is usually stable.

When it comes to English, it is generally believed that stress is manifested in terms of three acoustic features: fundamental frequency, duration, and intensity. Native speakers emphasize stressed vowels with greater intensity and pitch, and longer duration. Unstressed vowels are usually reduced to a centralized vowel (schwa) with lower intensity, lower pitch, and shorter duration.

Jun (2009) also compared prosodic systems of Korean and English from ToBI perspective. She noted that both languages have at least two prosodic units above the word level, which are marked by intonation. The intonation phrase (IP) is marked similarly in both languages, and focus is realized by expanded pitch range during the focused word and reduced after focus. Unlike English, in Korean accentual phrase is the lowest level of prosody and that there is no prosodic assignment at the lexical word level. One of the main difference is in how word prominence is realized. In English, prominence of a word is expressed by pitch accent, while in Korean it is achieved by placing the word at the beginning of a phrase.

Overall, most researchers agree, that F0 patterns are the primary means of realization of prosody in Korean, while in English duration and intensity play a much larger role.

2.2.2 Common Prosodic Mistakes

Then what are the typical mistakes, that Korean learners do in prosody

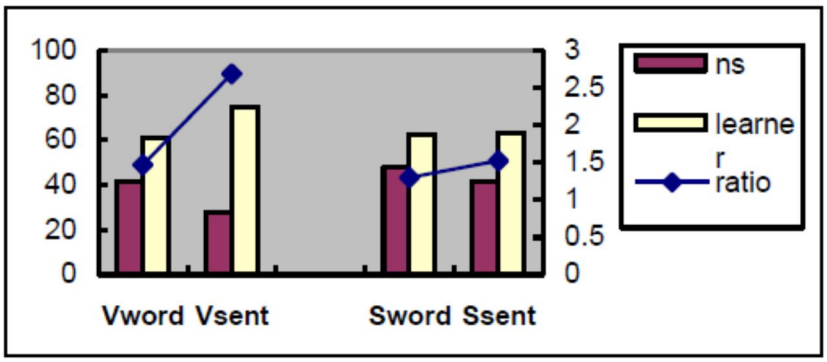


Figure 2.6. Percentage of stress reduction in learner speech as compared to that in native speech. Learner ratio shows greater discrepancy from that of native speech in sentence level. Vword=vowel duration in words, Vsent=vowel duration in sentences Sword=syllable duration in words, Ssent=syllable duration in sentences (Kim and Flynn, 2004).

when speaking English? The most often mentioned ones are related to:

- word stress, reduction of unstressed vowels, compound nouns;
- sentence stress, reduction of unstressed function words;
- focus;
- pitch accents

When it comes to word stress, it has been found that Korean speakers of English learn stress assignment quite well, but have troubled with stress reduction. In English, unstressed vowels are usually perceived as lower in pitch, shorter, and less loud than stressed vowels. The acoustic correlates of these features are lower fundamental frequency (F0), shorter duration, and weaker intensity. Korean learners reduce the duration of unstressed reduced vowels much less, compared to native speakers (Kim and Flynn, 2004; Lee et al., 2006).

Kim (2005) reported that Korean learners show F0 patterns similar to those of native speakers, when it comes to stress assignment. F0 slope (difference between adjacent elements) was measured in compound nouns and noun phrases; both groups had an F0 drop for compound nouns, but an F0 rise for

noun phrases. It is also interesting to note, that the difference was greater for isolate words, compared to words in sentence and paragraph contexts.

Different results were found for stress reduction, however. Kim and Flynn (2004) compared duration of vowels and syllables of target stimulus words pronounced by Korean speakers of English both in isolation and in sentence context (e.g. comparing the duration of vowels in contrasting pair „[a]dd" and „[a]ddition"). The results suggested that learners better execute stress reduction at the word level rather than at the sentence level (see Figure 2.6). They speculated that articulation of a sentence puts additional demands on learners, and makes it hard to attend to all the phonological aspects.

Similar results were found by Lee et al. (2006). They studied the production of unstressed vowels in English by early and late Korean- and Japanese-English bilinguals. Korean groups were nativelike in having a lower F0 for unstressed as opposed to stressed vowels, but made less of an intensity difference between unstressed and stressed vowels, than native speakers, as well as less of a difference in duration. They had longer unstressed vowels and lower intensity of stressed ones.

Why would stress assignment be easier, than stress reduction for Korean learners of English? It may be due to the fact that the Koreans are more used to the realization of prosody through F0 variation. In contrast, stress reduction is difficult, because Korean phonology does not have unstressed vowels.

In English, stress is manifested not only on word level, but on sentence level as well. Yoo (2012) investigated the acquisition of English sentence stress by Korean learners. Acoustic differences (in terms of F0, intensity and duration) between stressed and unstressed vowels were analysed. In general, the patterns of stressed and unstressed vowels produced by the Korean speakers were similar to the patterns of the native speakers, even though ratios differed. The learners depended more on duration and intensity to express the

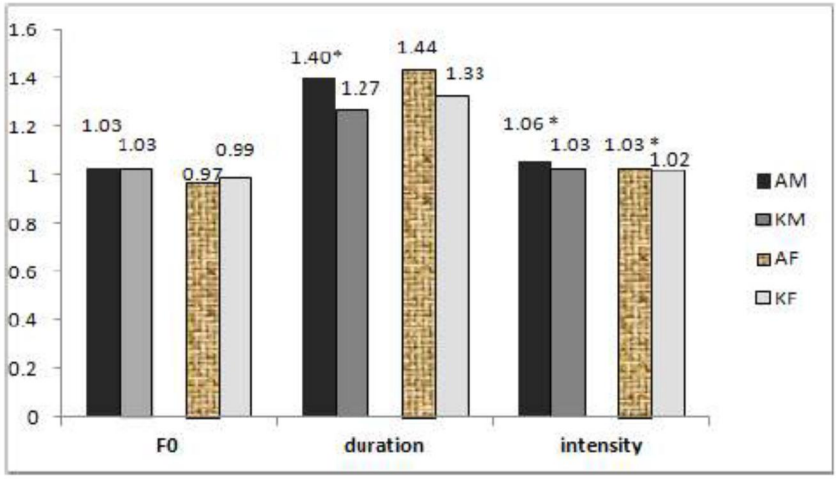


Figure 2.7. Mean ratios of acoustic values between stressed and unstressed vowels. AM = American male speakers, AF = American female speakers, KM = Korean male speakers, KF = Korean female speakers (Yoo, 2012).

difference between the two types of vowels (Figure 2.7). The degree of reduction of function words was also investigated. The most interesting find was that while native speakers reduced all the function words to a relatively similar degree, Korean learners showed great variance between different types of function words. In particular, four categories showed relatively longer duration, compared to native speaker's production: prepositions, relative pronouns, definite articles, and conjunctions.

Other researchers concentrated on studying of the production of intonation patterns by Korean learners of English. Kang et al. (2012) examined the phonetic realization of English focus by Korean learners of English at different levels of L2 immersion experience, but with almost the same L2 proficiency. Features like pitch accent patterns, pitch range, and duration of focused words were measured. Korean learners produced shorter focus words (in relation to the duration of the entire utterance), with a narrower pitch range. It is also interesting that learner's performance differed significantly depending on the semantic type of a sentence. Korean speakers of English

seemed to have the most difficulty with prosodic patterns of unergative sentences. Similar results were found by Um (2004). They noted that Korean speakers have little trouble with the use of phrase accents and boundary tones, but struggle with pitch accents, especially when signalling new or contrastive information. Their pitch values for new information were lower than native speakers', and they also had problem with deaccenting of given information (only 50% did it). This might be partially explained by the fact, that Korean has special morphological markers for focus and topic; and their absence in English might make it harder for Korean learners to apprehend information structure of a sentence.

Thus, we can point two main areas of prosody that Korean speakers seem to have problems with: stress (both at word and sentence level) and focus realization. In both cases there was difference in duration compared to native speakers; when dealing with focus Korean learners had problems with correct pitch range as well. In English these prosodic features are realized through vowels. Hence, for an automatic prosody transplantation system, it will be more sensible to modify vowel segments only, and leave consonants completely unchanged.

Another question we should ask is whether improvement in these acoustic features will result in a more native-like accent? Yoo (2017) examined the effects of acoustic features on comprehensibility ratings of English speech, read aloud by Korean students. Data from two groups of Korean learners (from years 2000 and 2012) were analysed rated by native speakers of English. Results showed that higher comprehensibility ratings correlated stronger with suprasegmentals rather than segmental features. Moreover, acoustic features of stress were also measured and a positive correlation was found between pitch ratio, duration ratio, intensity ratio, pitch range, speech rate and comprehensibility rating. Speech rate and duration ratio seemed to be

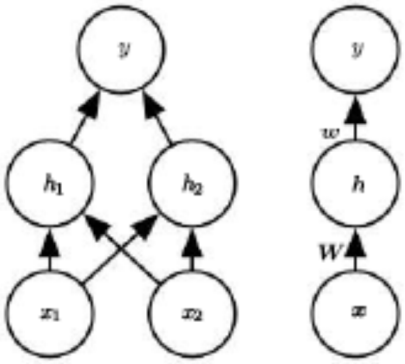


Figure 2.8. Feedforward network with a single hidden layer containing two units (Goodfellow et al, 2016).

associated with comprehensibility to the greatest degree. This supports the idea that teaching of prosodic features can improve learner's overall language proficiency.

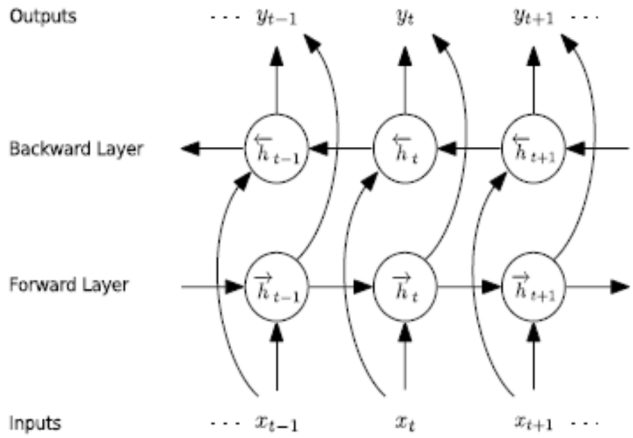
2.3 Deep Learning Based Prosody Prediction

2.3.1 Deep Learning

Deep learning has emerged as a new area of machine learning research since 2006 and techniques developed from it have already been impacting a wide range of signal and information processing work, including natural language and speech processing. Deep learning can be defined as "a class of machine learning techniques that exploit many layers of non-linear information processing extraction and transformation, and used for pattern analysis and classification for supervised or unsupervised feature" (Deng and Yu, 2014, p. 199). There are two essential elements: a model should consist of multiple layers of nonlinear information processing, hence the name "deep"; and feature representation should be done at a higher, more abstract layer.

Historically, the concept of deep learning originated from artificial neural network research. Feed-forward neural networks or multi-layer perceptrons (MLPs) with many hidden layers, which are often referred to as deep neural networks (DNNs), are good examples of the models with a deep architecture (see Figure 2.8). They did not receive wide use though, until the optimization difficulty associated with the deep models was empirically alleviated when a reasonably efficient, unsupervised learning algorithm was introduced in Hinton et al. (2006), and Hinton and Salakhutdinov (2006).

2.3.2 Recurrent Neural Networks



$$\overrightarrow{h}_t = \mathcal{H}\left(W_{x\overrightarrow{h}}x_t + W_{\overrightarrow{h}\overrightarrow{h}}\overrightarrow{h}_{t-1} + b_{\overrightarrow{h}}\right)$$
$$\overleftarrow{h}_t = \mathcal{H}\left(W_{x\overleftarrow{h}}x_t + W_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h}_{t+1} + b_{\overleftarrow{h}}\right)$$
$$y_t = W_{\overrightarrow{h}y}\overrightarrow{h}_t + W_{\overleftarrow{h}y}\overleftarrow{h}_t + b_y$$

Figure 2.9. Bidirectional RNN architecture. $\vec{h}(t)$ stands for the state of the sub-RNN that moves forward through time; $\overleftarrow{h}(t)$ stands for the state of the sub-RNN that moves backward through time (Goodfellow et al, 2016).

Although DNNs are able to offer improvements over other baseline prosody-prediction models, they still fall short of reproducing the naturalness and range observed in natural speech. One possible shortcoming is that they are typically trained using a "localized" window of input patterns, i.e., the current output is predicted from the current input, plus possibly a few fixed-length adjacent input observations to provide context. Such non-local dependencies are arguably one of the factors contributing to surface prosody. A Recurrent Neural Network (RNN) is a class of models that has been proposed as an alternative to address the challenge posed by time series that have complex contextual dependencies that go beyond a fixed time lag.

A recurrent neural network (RNN) is a class of artificial neural network

where connections between nodes form a directed graph along a sequence. This allows it to exhibit dynamic temporal behaviour for a time sequence. They can use their internal state (memory) to process sequences of inputs. One type of RNNS that is especially suited to work with speech is bidirectional RNN (see Figure 2.9). Bidirectional RNNs combine an RNN that moves forward through time begging from the start of the sequence with another RNN that moves backward through time beginning from the end of the sequence.

Additionally, these bi-directional, deep-in-time structures can be stacked to allow for more complex models that are also deep across layers and can, analogously to simple DNNs, progressively extract structure from the successive layer compositionality.

2.3.2 The Long Short-Term Memory Architecture

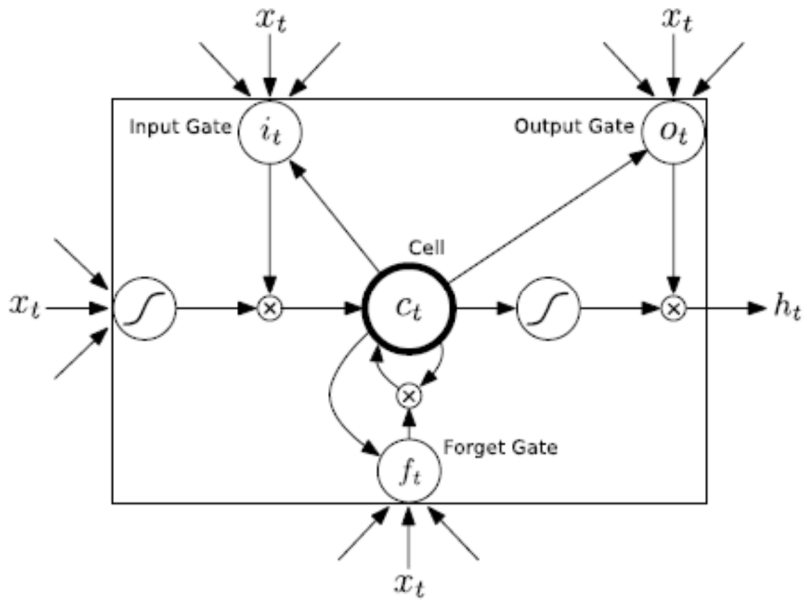


Figure 2.10. LSTM architecture. x_t is input at time t; h_t is hidden state at time t (Deng and Yu, 2014).

Long short-term memory (LSTM) units (or blocks) are a building unit for layers of a recurrent neural network (RNN). An RNN composed of LSTM units is often called an LSTM network (Hochreiter and Schidhuber, 1997). A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate (see Figure 2.10). The cell is responsible for "remembering" values over arbitrary time intervals; hence the word "memory" in LSTM. The LSTM gates compute an activation, often using the logistic function. Intuitively, the input gate controls the extent to which a new value flows into the cell, the forget gate controls the extent to which a value remains in the cell and the output gate controls the extent to which the value in the cell is used to compute the output activation of the LSTM unit. There are connections into and out of these gates. A few connections are recurrent. The weights of these connections, which need to be learned during training, of an LSTM unit are used to direct the operation of the gates. Each of the gates has its own parameters, that is weights and biases, from possibly other units outside the

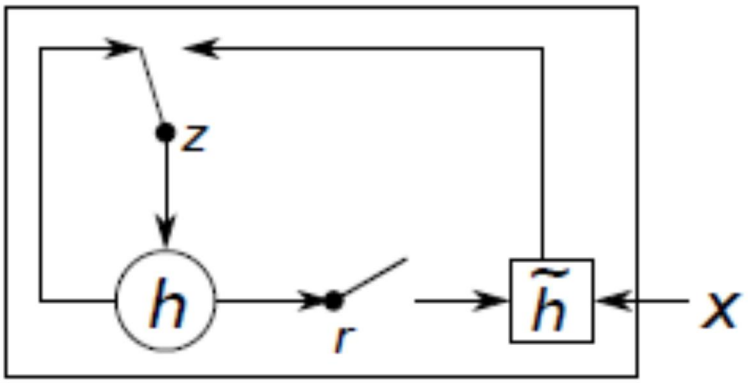


Figure 2.11. An illustration of GRU. The update gate z selects whether the hidden state is to be updated with a new hidden state $\tilde{h}$. The rest gate r decides whether the previous hidden state is ignored (Cho et al, 2014).

LSTM unit.

An LSTM is well-suited to classify, process and predict time series given time lags of unknown size and duration between important events. LSTMs were developed to deal with the exploding and vanishing gradient problem when training traditional RNNs. Relative insensitivity to gap length gives an advantage to LSTM over alternative RNNs, hidden Markov models and other sequence learning methods.

2.3.3 Gated Recurrent Units

Gated recurrent units (GRUs) are a gating mechanism in recurrent neural networks, introduced in Cho et al. (2014). Their performance on polyphonic music modelling and speech signal modelling was found to be similar to that of (LSTM).

GRU architecture was modelled after LSTM, but is simpler to compute and implement (see Figure 2.11). In this model, when the reset gate r is close to 0, the hidden state is forced to ignore the previous hidden state and reset with the current input only. This effectively allows the hidden state to drop any information that is found to be irrelevant later in the future, thus, allowing a

more compact representation.

On the other hand, the update gate z controls how much information from the previous hidden state will carry over to the current hidden state. This acts similarly to the memory cell in the LSTM network and helps the RNN to remember long-term information. As each hidden unit has separate reset and update gates, each hidden unit will learn to capture dependencies over different time scales. Those units that learn to capture short-term dependencies will tend to have reset gates that are frequently active, but those that capture longer-term dependencies will have update gates that are mostly active.

2.3.4 Prosody Prediction Models

Table 2.3. Review of previous research on DNN-based prosody prediction.

Paper	Target Parameter	Prediction Method	Features
Sreenivasa Rao et al. (2007)	Syllable duration	4-layer FFNN	Phonological, Positional, contextual features and gender (25 in total)
Shreekanth et al. (2015)	Syllable duration	FFNN	Syllable identity, position within word
Fernandez et al. (2014)	F0, syllable duration	LSTM-RNN	Categorical labels, counts (number of phones/syllables/words to a phrase/ sentence boundary, etc.), context.
Sheikhan (2017)	Pitch contour, syllable duration, vowel duration	BPSO–PSO–Optimized RNN	POS tags, syllable-level features, position, context
Gu et al. (2010)	Syllable duration, pitch contour	HMM-DNN	Discrete cepstrum computation (DCC) coefficients
Bernardy et al. (2017)	Tonal contour classification	LSTM, CNN	F0 contour
Ding et al. (2015)	Prosodic boundary labels	FFNN, BLSTM-RNN	Embedding feature vectors of characters
Garbe et al (2017)	F0	LSTM	Text-derived linguistics and duration features
Su et al. (2016)	Pitch contour	CNN	spectrogram

Traditionally, majority of speech (and prosody) synthesis systems have been based on Hidden Markov Models Model (HMM). Since the middle of 2000s, however, application of DNN's to this task has seen a considerable increase and is continuing to increase at the moment. As for the prosody, deep neural networks have shown to be successful at predicting various acoustic values (such as duration, pitch, intensity) or prosodic labels.

The first models to be successfully implemented were feed-forward neural networks (Sreenivasa Rao et al., 2007; Shreekanth et al., 2015). RNNs were applied later and began to consistently outperform basic models (Fernandez et

al., 2014; Ding et al., 2015; Su et al., 2016). DNNs can be used not only to predict acoustic features like duration or F0, but also to predict or determine entire pitch contours or numerous prosodic labels. We summarize selected papers on DNN-based prosody prediction in Table 2.3.

Chapter 3. Vowel Duration and Fundamental Frequency Prediction Model

3.1 Data

Data for model training was taken from The Boston University Radio News Corpus, which includes speech from FM radio news announcers (Ostenford et al., 1996). This particular corpus seemed to be the best suited for our task for the following reasons: all the data are from the native speakers of American English – the variant of English that is mostly widely taught in South Korea. As such prosodic patterns from the corpus are well suited to be used as reference in self-imitation learning by Korean students. Second, FM newscaster are generally required to read out text in a pleasant way: that is the prosody should not be too monotonous. That means that this corpus should be especially good for prosodic parameters modelling.

The corpus consists of two parts: main portion contains news stories recorded in the radio studio during broadcast (see Table 3.1). The second, smaller, part consists of news announcers reading short stories in lab environment (see Table 3.2).

For our research we decided to use only the data from female speakers. Female speech typically has much more variability when it comes to prosody, the F0 is higher, and pitch range is broader. As such, it is better suited for prosody modelling. Among the three female speakers, two were selected: F2B and F3A. F3A data has twice more data compared to other speakers; while F2B data is the only one that was fully hand-corrected for phonetic annotations. Data from the two speakers will be used to build two different models: one with a smaller amount of hand-corrected data, and the other with a bigger amount of automatically corrected data. The comparison of the

Table 3.1. Duration in minutes of speech, and other statistics about the radio news stories (Ostendorf et al., 1996).

Speaker	**F1A**	**F2B**	**F3A**	**M1B**	**M2B**	**M3B**	**M4B**
Minutes	52	49	107	48	58	32	91
Stories	43	34	340	36	35	21	62
Clean Paragraphs	276	124	341	161	214	126	236
Noise Paragraphs	1	40	51	108	102	32	41
Words (*times* 1000)	11.9	12.2	28.6	15.7	18.4	10.5	25.6

results from the two models will give us valuable information on optimal data selection for future projects.

The data consists of paragraph sized units, each including several sentences. The annotation includes orthographic transcription, phonetic alignments, part-of-speech (POS) tags, and prosodic labels (ToBI-based). The orthographic transcription and ToBI labels were marked by hand, while the rest were generated automatically. Only the data for F2B speaker received prosodic annotation.

The phonetic alignment is based on the TIMIT phonetic labelling system (Zue and Seneff, 1996). The set consists of 61 labels that can be seen in Appendix 1, among them *ux, ix, ax-h* and *epi* are not used in the BURNC. Additionally, stressed vowels are marked with '+1'; segmentation times are provided in units of 10-msec frames.

Table 3.2. Characteristics of the news stories recorded in the lab for multiple speakers (Ostendorf et al., 1996).

Speaker	CJ	CP	TP	SR
Paragraphs	6	4	7	7
Sentences	23	22	28	36
Words	445	388	577	713

POS tagging was done automatically based on Penn Treebank (Marcus and Santorino, 1993). The set consists of 47 POS tags; the 36 used word categories (excluding punctuation labels) are summarized in Appendix 2. And, finally, prosodic annotation was manually done according to the ToBI system (Silverman et al., 1992).

3.2. Input Feature Selection

The aim of our research is to develop an automatic method of prosodic parameters transplantation. That means, that, ideally, only information that can be obtained automatically, should be used as input features in our model. Yet, parts of the data in corpus received manual prosodic annotation. And seems the aim of this research is to build prosody prediction system, we decided to test two different models using F2B speaker data. The first model will use the following features:

- Phoneme identity
- Phoneme type
- Stress
- Part-of-speech (POS) tag
- Boundary tone tag
- Pitch accent tag

The other model will have the same input features, with the exception of

break indices and pitch accent tags. This way, all the features used in the second model can be automatically obtained, which is crucial for a self-imitation learning system. Additional, third model will use data from the F3A speaker. It contains twice as much speech, but since phone alignment was not hand-corrected, we expect mixed results. When building a prosody prediction system, it is common to use many other features, like number of phones in a syllable, in a word, position of a syllable within the word, identity of neighbouring syllables, words, etc. But the nature of Bi-LSTM model is that it already takes into account contextual information, hence we consider inclusion of such features to be redundant.

Phoneme identity. As the purpose of our model is to predict vowel duration and F0, only phones with the following vowel labels will be used (17 in total): *iy, ih, eh, ey, ae, aa, aw, ay, ah, ao, oy, ow, uh, uw, er, ax, axr*. Besides that, we will also include syllabic consonants into our model: *em, en*. In English, syllabic consonant can take the role of syllable nucleus and play part in the rhythmic organization of speech. Besides, words with syllabic consonants are often transcribed as having a reduced vowel, and are pronounced like that by some speakers, e.g. bottom as /ˈbɒtəm/ or /ˈbɒtm/. In total, phoneme identity feature will consist of 19 labels in total. All of them were one-hot-encoded, that is are represented as a sequence of 'zeroes' with only single position marked by 'one'.

Table 3.3. F2B speaker, duration by *phoneme type*.

Phoneme Type	Mean (m-sec)	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
Monophone	76.4	40.0	75.9	76.9
Diphone	129.1	41.6	127.9	130.4
Syl. consonant	109.2	39.2	107.7	110.7

Table 3.4. F3A speaker, duration by *phoneme type*.

Phoneme Type	Mean (m-sec)	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
Monophone	65.5	40.5	65.2	65.8
Diphone	116.2	46.1	115.3	117.2
Syl. consonant	92.2	42.7	91.2	74.6

Phoneme type. We classified vowel phonemes into monophones and diphones, plus syllable consonants. The reason for this is that intuitively, diphones or syllabic consonants should have longer duration, compared to monophones. To test this hypothesis, we analysed the differences between the three groups. We carried out ANOVA tests for data from both speakers: duration and F0 were set as dependent variables, and effects of *phoneme type* labels MONO, DI and R-COL were tested. In this and following cases, we got rid of any outliers before conducting the tests. A value was considered an outlier if it fell below *Q1–1.5*IQR* or above *Q3+1.5*IQR*, where *Q1* – lower quartile, *Q3* – upper quartile, and *IQR* = *Q3–Q1* – interquartile range.

There was a statistically significant difference between groups as determined by one-way ANOVA both for F2B data ($F(2, 35291) = 3835.103$, $p < 0.01$) and for F3A data ($F(2, 79825) = 7121{,}356$, $p < 0.01$). For F2B speaker, differences in duration between all three groups of phonemes were significant ($p < 0.01$). The test revealed that diphones have the longest

duration (129.1 ± 41.6 m-sec) followed by syllabic consonants (109.2 ± 39.2 m-sec) and monophones (76.4 ± 40.0 m-sec). Similar pattern was observed for F3A data as well: diphones (116.2 ± 46.1 m-sec) followed by syllabic consonants (92.2 ± 42.7 m-sec) and monophones (65.5 ± 40.5 m-sec) respectively.

As for the fundamental frequency, the results were significant as well ($F(2, 316455) = 324.361$, $p < 0.01$ for F2B and $F(2, 612247) = 1895.375$, $p < 0.01$ for F3A). All the differences between phoneme types were significant, although F0 value for monophones and diphones did not differ much (see Table 3.5): 165.2 ± 44.1 Hz for monophones and 166.1 ± 43.2 Hz for diphones respectively. While syllabic consonants F0 turned out to be somewhat lower: 158.8 ± 44.3 Hz. followed by monophones () and syllabic consonants (). The same correlation was found for F3A data: higher F0 for monophones and diphones (195.9 ± 44.3 Hz and 197.7 ± 43.7 Hz respectively), followed by syllabic consonants (185.7 ± 44.6 Hz). Overall, fundamental frequency for F3A speaker was significantly higher, compared to F2B data: 195.1 Hz against 164.8 Hz, which is 18.3% higher.

Table 3.5. F2B speaker, F0 by phoneme type.

Phoneme Type	Mean (Hz)	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
Monophone	165.2	44.1	165.7	166.4
Diphone	166.1	43.2	165.0	165.4
Syl. consonant	158.8	44.3	158.4	159.3

Table 3.6. F3A speaker, F0 by *phoneme type*.

Phoneme Type	Mean (Hz)	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
Monophone	195.9	44.3	195.8	196.1
Diphone	197.7	43.7	197.4	197.9
Syl. consonant	185.7	44.6	185.3	186.0

Stress. For stress we used binary encoding: '1' for stressed phones and '0' for unstressed. For both speakers the difference between groups was significant ($F(1, 35292) = 5493.117$ $p < 0.01$ for F2B data and $F(1, 79826) = 3106.277$ $p < 0.01$ for F3A data). Stressed vowels were on average 46.8% longer for F2B speaker (see Table 3.7), and 26.5% longer for F3A speaker (see Table 3.8).

Table 3.7. F2B speaker, duration by *stress*.

Stress	Mean (m-sec)	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
Stressed	101.9	44.5	68.8	69.9
Unstressed	69.4	37.6	.101	.103

Table 3.8. F3A speaker, duration by *stress*.

Stress	Mean (m-sec)	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
Stressed	85.4	46.4	84.9	86.0
Unstressed	67.5	42.6	67.1	67.9

Table 3.9. F2B speaker, F0 by *stress*

Stress	Mean (Hz)	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
Stressed	171.9	43.0	171.7	172.1
Unstressed	153.6	43.1	153.4	153.9

Table 3.10. F3A speaker, F0 by *stress*.

Stress	Mean (Hz)	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
Stressed	208.1	46.4	208.0	208.3
Unstressed	184.9	40.3	184.8	185.0

Not surprisingly, in case of fundamental frequency similar correlation was

Table 3.11. F2B speaker, duration by *POS tag*.

POS tag	Mean (m-sec)	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
CONT	90.7	45.2	90.2	91.2
FUN	67.8	35.4	67.0	68.8

Table 3.12. F3A speaker, duration by *POS tag*.

POS tag	Mean (m-sec)	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
CONT	78.3	45.8	77.9	78.7
FUN	59.6	38.5	59.1	60.2

Table 3.13. F2B speaker, F0 by *POS tag*.

POS tag	Mean (Hz)	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
CONT	165.5	43.9	165.4	165.7
FUN	160.9	44.0	160.6	161.3

Table 3.14. F3A speaker, F0 by *POS tag*.

POS tag	Mean (Hz)	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
CONT	196.4	45.2	196.3	196.5
FUN	188.7	40.6	188.4	188.9

observed. For F2B data stressed vowels have mean F0 of 171.9Hz against 153.6 Hz for unstressed ($F(1, 316456) = 13551.311$ $p < 0.01$). And 208.1Hz F0 of stressed vs. 184.9Hz of unstressed vowels for F3A data respectively ($F(1, 612249) = 43726.256$ $p < 0.01$).

POS tag. The BURNC uses Treebank set for part-of-speech tagging. But for our task, the sheer number of tags it uses, seemed redundant. We tried to find possible way to narrow down the tag set. As was mentioned, in Chapter 2, one of the common area of mistakes for Korean learners of English is sentence stress. Content words receive full stress, while function words are typically reduced. This binary division seemed suitable for our task, and we grouped all tags into two categories:

- Content words – CD, EX, FW, JJ, JJR, JJS, LS, NN, NNS, NNP, NNPS, PRP, PRP%, RB, RBR, RBS, VB, VBD, VBG, VBN, VBP, VBZ (22 in total).
- Function words – CC, DT, IN, MD, PDT, POS, PP, PP$, RP, TO, WDT, WP, WP$, WRB (14 in total).

We analysed differences between the two groups. There was a statistically significant difference between durations for both speakers ($F(1, 35292) = 1690.85$ $p < 0.01$ for F2B and $F(1, 79826) = 2367.671$ $p < 0.01$ for F3A). Similar results were found for both speakers. Content words' duration was 33.7% longer for F2B and 31.3% longer for F3A respectively (see Table 3.11 and Table 3.12). They also had higher fundamental frequency: 165.5 against 160.9 Hz for F2B speaker ($F(1, 316456)= 485.812$ $p < 0.01$) and 196.4 against 188.7 Hz for F3A data ($F(1, 612248) = 2595.859$ $p < 0.01$) (see Table 3.13 and Table 3.14).

Prosodic labels. The data were hand labelled according to the ToBI system. In this study we used boundary indices and pitch accent labels (see Table 3.15 and 3.16). Break indices are typically assigned per word; so each phone (or frame0 was given the label of the word it belongs to. Only accented syllables are labelled for pitch accents. Phones (and frames) were assigned the same label as the syllable they belong to. We introduced an additional label for cases, when a phone belonged to an unlabelled syllable or the annotator was

Table 3.15. ToBI break index values.

Index	Description
0	clear phonetic marks of clitic groups
1	most phrase-medial word boundaries
2	a strong disjuncture marked by a pause or virtual pause, but with no tonal marks
3	intermediate intonation phrase boundary
4	full intonation phrase boundary

Table 3.16. ToBI pitch accents.

Accent	Description
H*	peak accent
L*	low accent
L*+H	scooped accent
L+H*	rising peak accent
H+!H*	a clear step down onto the accented syllable from a high pitch

not sure which accent to mark.

Only F2B data were labelled for ToBI markers. We analysed F0 and duration differences both for break indices and pitch accents. The results for duration differences between break indices were significant ($F(4, 35289) = 378.499$ $p<0.01$) (see Table 3.17). Only the difference between duration in labels 2 and 3 wasn't significant ($p = 0.963$). Break 0 have the shortest duration (69.7 ± 42.3 m-sec), followed by break 1 (76.8 ± 38.4 m-sec), breaks 2 and 3 (86.7 ± 42.3 and 87.3 ± 44.4 m-sec), and break 4 exhibiting the longest duration (97.7 ± 49.7 m-sec). As such break indices are good indicators of vowel duration and can be useful in a prediction system.

Table 3.17. F2B speaker, duration by *break* tag.

Break tag	Mean (m-sec)	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
0	69.7	42.3	65.7	73.7
1	76.8	38.4	76.2	77.4
2	86.7	42.3	85.6	87.9
3	87.3	44.4	86.0	88.7
4	97.7	49.7	96.8	98.7

Table 3.18. F2B speaker, F0 by *break* tag.

Break tag	Mean (Hz)	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
0	168.7	42.9	167.2	170.2
1	168.4	43.6	168.2	168.7
2	170.6	41.6	170.2	171.0
3	178.2	43.5	177.8	178.7
4	154.3	43.2	154.1	154.6

As for the fundamental frequency, results were significant as well, but much less conclusive ($F(4, 316453) = 3005.228$ $p < 0.01$) (see Table 3.18). There was statistical difference in F0 between break indices 3 and 4 and the others, but not between labels 0, 1 and 2 (0 * 1 p=0.995; 0 * 2 p = 0.131). As such, F0 prediction model is unlikely to benefit from the inclusion of break labels.

Results for the duration and pitch accent tags were significant ($F(5, 35288) = 764.523$ $p < 0.01$) (see Table 3.19). But only unlabelled vowels ('_') and **H+!H*** were statistically different from the rest. All other tags did not differ enough, thus we can speculate, that the inclusion of pitch accent labels is

Table 3.19. F2B speaker, duration by *pitch accent tag.*

Break tag	Mean (m-sec)	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
_	76.7	41.2	76.1	77.2
H*	105.4	44.1	104.5	106.4
H + !H*	128.7	52.8	117.4	140.1
L*	111.7	47.1	107.4	116.0
L* + H	97.9	36.0	87.4	108.4
L + H*	110.3	43.0	108.3	112.3

Table 3.20. F2B speaker, F0 by *pitch accent tag.*

Break tag	Mean (m-sec)	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
_	153.1	41.9	152.9	153.3
H*	185.9	40.1	185.7	186.2
H + !H*	170.6	45.9	168.1	173.2
L*	133.0	30.25	132.2	133.7
L* + H	149.8	37.7	146.3	153.2
L + H*	187.7	35.9	187.2	188.2

unlikely to improve performance of duration prediction model.

Opposite results were observed in case of fundamental frequency F(5, 316452) = 10350.711 p< 0.01) (see Table 3.20). Only difference between **'_'** and **L* + H** was statistically insignificant (p = 0.489). **L+H*** showed the highest F0 (187.7 ± 35.9 Hz), followed by H* (185.9 ± 40.1 Hz), H+!H* (170.6 ± 45.9 Hz), and L* (133.0 ± 30.25 Hz). We can conclude, that pitch accent labels can be good predictors of fundamental frequency.

Overall, the input to models F2B-ToBI will have the following dimensions (35 in total):

- 19 for phoneme identity
- 3 for phoneme type
- 1 for stress
- 1 for POS tag
- 5 for break indices
- 6 for pitch accents

Models F2B and models F3A will have 19 input features each respectively. For duration prediction model, the input will be per phoneme. For fundamental frequency prediction model, the input will be per frame instead. In the data F0 is measured by the intervals of 10 milliseconds. We extracted F0 information from each phone and then measured the number of frames in each IP (For F2B data). The average length turned to be 67, the maximum – 228 and the minimal – 11. For F3A data, the average length turned to be 137, the maximum – 500, and the minimal – 9. We decided the use the input length of 40.

3.3 System Architecture and Training

To test the proposed approaches, we built 18 systems overall: 9 for vowel duration prediction and 9 for F0 prediction. The models were trained on 3 different sets of input data: F2B, F2B-ToBI and F3A. And three different DNN architectures were tested: baseline RNN model, LSTM model and GRU model. Otherwise the models have the same amount of layers, nodes per layers and same hyper-parameters. That is done in order to be able to compare different DNN models' performance with each other. The general network outline on the example of LSTM can be seen in Figure 3.1. Feature inputs and target values (duration or F0) are fed into a network that consists of 3 bi-directional layers, followed by an output layer that does the prediction, followed by a loss function. The Python code for the model architecture

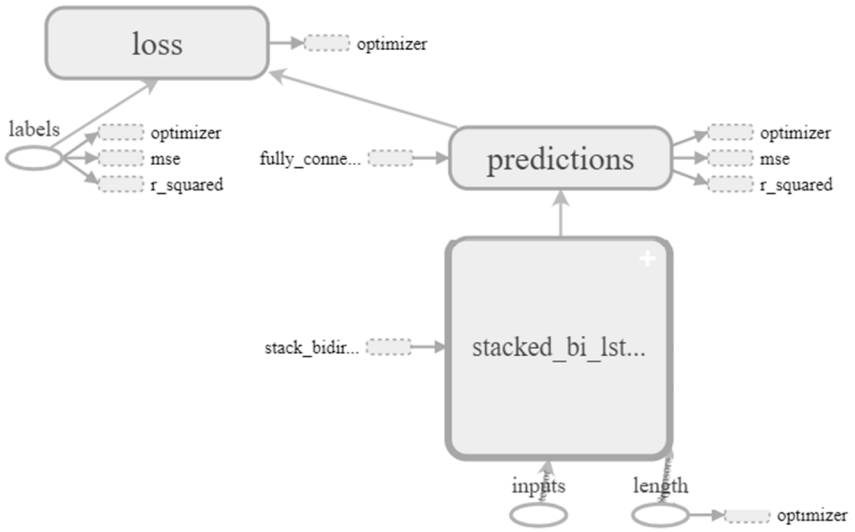


Figure 3.1. LSTM model architecture outline.

(baseline RNN model) can be found in Appendix 3.

Hidden layers. The network architecture consists of three stacked bidirectional hidden layers with 50, 40, and 30 units per layer (see Figure 3.2). The diminishing amount of nodes with each layer is supposed to help the net to generalize features better. Each layer consist of forward and backward cells, and after each layer their output is merged together, which allows the network to utilize both previous and following context. For this experiment, we built different models with basic RNN, LSTM and GRU cells respectively.

Activation function. The activation function used for all models is hyperbolic tangent:

$$\tanh(z) == \frac{\sinh(z)}{\cosh(z)} == \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}.$$

This function is well-suited for the use in DNNs due to the fact that its output is bound to the range of (-1, 1), which is ideal (see Figure 3.3).

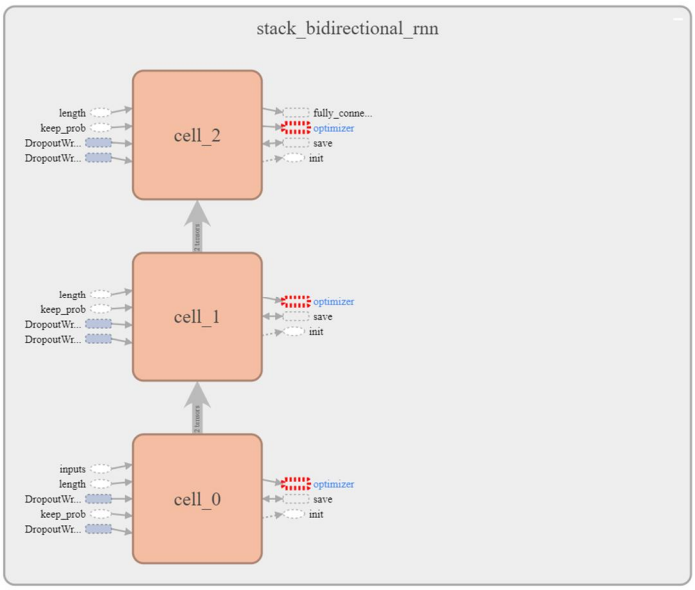


Figure 3.2. Bi-directional LSTM layer.

Dropout. We applied dropout to all hidden cells. Dropout is a regularization technique for reducing overfitting in neural networks by preventing complex co-adaptations on training data. As the name suggests, a specified amount of nodes in a layer are 'dropped', that is they do not fire and do not connect to the same layer. Dropout was also implemented to both forward and backward cells, and set the rate to 0.7 (that means only 70% of nodes will activate). Dropout is only used during training, for testing all nodes are active again. In many ways this method is similar to cross validation.

Output layer. Three stacked hidden layers are then followed by an output layer, which computes the predicted value. We used a simply linear activation function:

$\boldsymbol{y} = \boldsymbol{wx} + \boldsymbol{b}$, where

- $\boldsymbol{y}$ is output,
- $\boldsymbol{x}$ is input from previous layer,
- $\boldsymbol{w}$ are weights,
- $\boldsymbol{b}$ is bias.

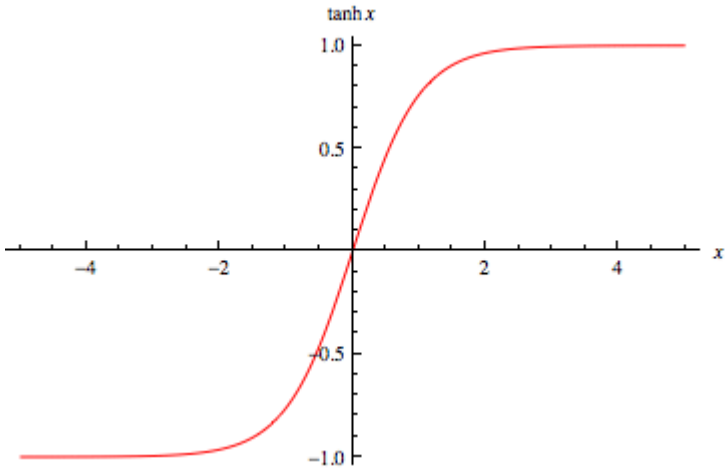


Figure 3.3. Tangent function.

The output is one-dimensional: it predicts duration and F0 values in each type of model respectively.

Loss function. The loss function implemented was based on mean squared error (MSE). MSE measures the average of the squares of the errors—that is, the average squared difference between the predicted values and what is expected. MSE is a risk function, corresponding to the expected value of the squared error loss. For a prediction model such as $h_\theta(x_i) = \theta_0 + \theta_1 x$, where the inputs are a feature vector x_i, the MSE is given by summing across all N training examples, and for each example, calculating the squared difference from the expected value y_i and the prediction $h_\theta(x_i)$:

$$J = \frac{1}{N}\sum_{i=1}^{N}(y_i - h_\theta(x_i)^2$$

Optimization algorithm. The network was trained with the backpropagation algorithm using mini-batch stochastic gradient descent (SGD) as the optimiser. The SGD algorithm we used was Adam (adaptive moment estimation) optimizer (Kingma and Lei Ba, 2015). Unlike other algorithms, instead of adapting the parameter learning rates based on the average first moment (the mean) as in Root Mean Square Propagation (RMSProp), Adam

also makes use of the average of the second moments of the (see Figure). Specifically, the algorithm calculates an exponential moving average of the gradient and the squared gradient, and the parameters beta1 and beta2 control the decay rates of these moving averages. This algorithm is:

- Straightforward to implement
- Computationally efficient
- Little memory requirement.
- Invariant to diagonal rescale of the gradients
- Well suited for problems that are large in terms of data and/or parameters
- Appropriate for non-stationary objectives
- Appropriate for problems with very noisy/or sparse gradients
- Hyper-parameters have intuitive interpretation and typically require little tuning

Input. Since we're using RNN architectures, we must also decide on the length of the input. For F2B data, we used ToBI break indices for the task. Index 4 is used to mark the end of an intonational phrase (IP), so it seemed a natural choice. We divided the data into chunks by break index 4, and then calculated the mean length of the IPs. It happened to be 7,5 phones, with the smallest IP having the length of 2, and the longest having the length of 30. The data for F3A speaker were not labelled for prosodic markers. We had to find another solution, but with a similar approach. Punctuation marks were used to divided the data into chunks. The average length turned to be 17,5, the maximum – 70, and the minimal – 2. We settled on the input length of 15 for the data from both speakers. Chunks, that were smaller than 15, received zero padding to reach the length of 15, while the bigger ones were split into smaller pieces. This way we try reduce the use of zero-padding as much as possible, while preserving input length the same for both models. Then the

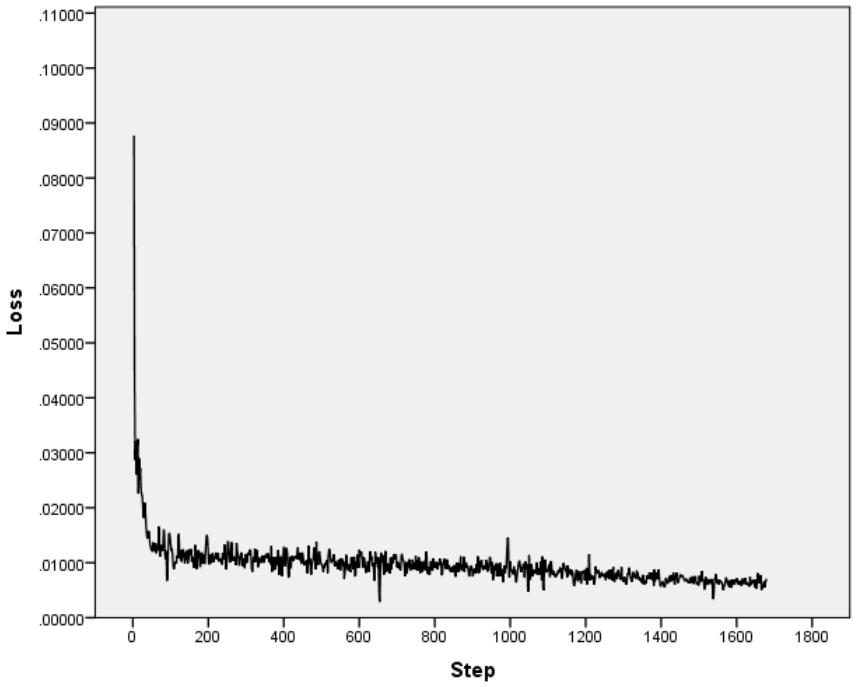


Figure 3.4. Scatter plot of loss value at each step. LSTM-F2B duration prediction model. The loss function is mean squared error (MSE). Each step corresponds to one batch of input data fed into the network.

input was fed into the network by batches of size 64. The input had the following dimensions for each model:

Vowel duration prediction models:

- Models F2B-ToBI – 35 by 15 dimensional vector
- Model F2B and F3A – 24 by 15 dimensional vector

F0prediction models:

- Models F2B-ToBI – 35 by 40 dimensional vector
- Model F2B and F3A – 24 by 40 dimensional vector

. As for the original values, fed into the net, first we checked for any outliers; if found, their values were adjusted to be within those boundaries. We considered such solution to be better, than outright eliminating them, since preserving the context is important. After that the input was z-score normalized to account for speaker differences. F0 values were additionally

changed into log scale, as log-F0 has shown to work better in DNNs. And, finally, the data were regularized into (0,01, 0.99) range, since (-1, 1) range is optimal for the neural network input. Python code for all the data pre-processing done can be found in Appendix 4.

Training. The entire data were divided into training and test sets with the ratio 85:15. The models have the following number of inputs:

- F2B, F2B-ToBI duration prediction – 36k phones
- F3A duration predictoin – 321k 10-msec frames
- F2B, F2B-ToBI F0 prediction – 81k phones
- F3A F0 predictoin – 630k 10-msec frames

We employed cross-validation technique: for each epoch, training set was split into training and validation subsets randomly (with the same ratio of 85:15). All models were trained in batches of size 64, with a learning rate of 0.01. Duration prediction models were trained for 30 epochs, F0 models – for 20 epochs. The aim of the training is to minimize the loss function (MSE). After each step, the optimization algorithm back propagates the error value and adjusts the weights of the network as to minimize MSE value as much as possible. You can see how the loss value changes with each step on the example of LSTM-F2B duration prediction model in Figure 3.4.

3.4 Results and Evaluation

3.4.1 Objective Metrics

To test the accuracy the following metrics were used: root mean squared error (RMSE), coefficient of determination (R-squared), correlation (COR), average deviation, normalized variance (NVAR), and percent of correct predictions within one standard deviation.

Root mean square error. RMSE is a measure of the differences between predicted values and observed values. RMSD is always non-negative, and a

value of 0 (never achieved in practice) would indicate a perfect fit to the data. In general, a lower RMSD is better than a higher one. RMSD is the square root of the average of squared errors:

$$RMSD = \sqrt{MSE(\widehat{\theta})}$$

Coefficient of determination. "R squared" is the proportion of the variance in the dependent variable that is predictable from the independent variable:

$\mathbf{R^2 = 1 - \frac{SS_{res}}{SS_{tot}}}$, where:

$\boldsymbol{SS_{res}}$ is residual sum of squares,

$\boldsymbol{SS_{tot}}$ is total sum of squares.

R-squared is always between 0 and 100%. 0% indicates that the model explains none of the variability of the response data around its mean. 100% indicates that the model explains all the variability of the response data around its mean. R-squared cannot determine whether the coefficient estimates and predictions are biased, which is why it must be used in conjunction with other metrics.

Correlation. The Pearson correlation coefficient (COR) measures the linear relationship between two datasets, and is calculated by the following equation:

$\mathbf{\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y}}$, where:

cov – the covariance,

$\boldsymbol{\sigma_x}$ is the standard deviation of x;

$\boldsymbol{\sigma}_y$ is the standard deviation of y.

It varies between -1 and +1 with 0 implying no correlation. Correlations of -1 or +1 imply an exact linear relationship. Positive correlations imply that as x increases, so does y. Negative correlations imply that as x increases, y decreases.

Normalized variance. Similar to R squared, variance is the expectation of the squared deviation of a predicted value from then mean. It is the square of

the standard deviation. It measures how far predicted values are spread out from their average value. Normalized variance indicates the ratio between the predicted variance and the natural variance of the test set. A value closer to 1 would mean the predicted values are spread in the same manner as the values in the test set. This metric shows how well the model is capable of simulating the variability that occurs in natural speech.

3.4.2 Vowel Duration Prediction Models Results

The results for vowel duration prediction models are summarized in Table 3.21. Four objective metrics are shown: RMSE, R-squared (R^2), Pearson coefficient of correlation (COR), and normalized variance (NVAR).

Overall, LSTM-F2B and LSTM-F2B-ToBI models performed the best. They had the lowest RMSE of 0.0221 and 0.0227, followed by GRU-F2B-ToBI model (0.0272). Lower RMSE value indicates small difference between predicted and original duration values. But it is not indicative of a good model on its own. Sometimes a model predicts values that are clustered together, while in the original data they are widly distributed: although the RMSE would be low in that case, such a model would be bad overall.

That's why we need to look at other values that indicate the correlation between variances in origian and predicted gorups. Again, LSTM-F2B-T0BI and LSTM-F2B models had the highest R-squared, COR, and NVAR values. The closer these values are to 1, the better the model represents natural distribution observed in the original data. If we compare LSMT-F2B-ToBI model with the basic LSTM-F2B model, the difference is not that big: 0.770 against 0.782 (R-squred), 0.889 against 0.885 (COR), and similar 0.783 of

Table 3.21. Objective metrics for the vowel duration model. The best score for each metric is shown in bold.

Model	Data	RMSE	R^2	COR	NVAR
RNN	F2B	0.0316	0.559	0.757	0.574
	F2B-ToBI	0.0326	0.540	0.773	0.597
	F3A	0.0382	0.392	0.650	0.417
LSTM	F2B	**0.0221**	**0.782**	0.885	**0.783**
	F2B-ToBI	0.0227	0.770	**0.889**	**0.783**
	F3A	0.0264	0.711	0.843	0.711
GRU	F2B	0.0328	0.518	0.761	0.577
	F2B-ToBI	0.0272	0.681	0.831	0.690
	F3A	0.0312	0.594	0.772	0.596

NVAR. Although inclusion of prosodic labels into the input did lead to slight increase in some of the metrics, in the others basic LSTM-F2B model outperformed it. If we consider, that prosodic labels have to be annotated manually, and as such can not be used in a fully automatic system, F2B model without ToBI labels seems a better choice for the automatic prosody transplantation system. Only in case of GRU models, the use of F2B-ToBI data lead to considerable gains in perfomance: 0.0272 against 0.0328 (RMSE), 0.681 against 0.518 (R squared), 0.831 against 0.761 (COR), 0.690 against 0.577 (NVAR).

If we compare F2B and F3A model, F2B model showed better results in

Table 3.22. Objective metrics for the fundamental frequency model. The best score for each metric is shown in bold.

Model	Data	RMSE	R^2	COR	NVAR
RNN	F2B	52.49	0.036	0.378	0.141
	F2B-ToBI	47.263	0.193	0.509	0.258
	F3A	46.77	0..057	0.390	0.150
LSTM	F2B	46.89	0.245	0.511	0.258
	F2B-ToBI	38.60	0.482	0.707	0.499
	F3A	40.357	0.305	0.584	0.339
GRU	F2B	47.285	0.223	0.503	0.253
	F2B-ToBI	**36.371**	**0.536**	**0.736**	**0.538**
	F3A	42.163	0.243	0.606	0.358

case of baseline and LSTM models. For GRU models, on the other hand, F3A model outperformed F2B model. It has lower RMSE (0.0312 againt 0.0328), and higher R^2 (0.594 against 0.518), COR (0.772 against 0.761), and NVAR (0.596 against 0.577). The differences in performance for RNN and LSTM models can be explained by the difference in traning data. Although F3A had twice as much training data, which usually leads to better results, its annotations were not hand-corrected. For GRU model, however, hand-corrected annotation do not seem to be that inportant, and instead, the amount of training data might be a more important factor.

3.4.2 Fundamental Frequency Prediction Models Results

Different results for F0 prediction models (see Table 3.22). GRU-F2B-ToBI and LSTM-F2B-ToBI models considerably outperformed all the other models. It showed better performance in all parameters, RMSE of 36.371 and 38.600, r squared of 0.536 and 0.482, COR of 0.736 and 0.707, and NVAR of 0.538 and 0.499. The obvious conclusion is that prosodic ToBI labels are much better predictors of F0, compared to duration. If the aim of self-imitation prosody traning system is to learn how to place correct pitch accents or boundary tones, the inclusion of prosodic labels into the prediction system might be obligatory. If the aim of the CAPT system is to learn sentence stress or similar rythm-related phenomena, the inclusion of prosodic labels will be redundant as we saw in the previous sub-chapter. One might even omit F0 transplantation all together, and perform duration transplantation only. These might help learners to pay more attention to problematic areas, and not be distracted by pitch differences.

If we compare F2B and F3A models, F3A did better in all cases: lower RMSE, higher COR and NVAR. One possibility is that bigger amount of traning data leads to better resutls in F0 prediction, than it did for duration prediciton. It does not still explain F2B model's very low R-squared score, and a further investigation of this is required.

3.4.3 Comparison with other models

While in the previous section we compared our models with each other, it did not really tell us whether their perfomance was satisfactory or not. To find that out, we additionally compared our best duration prediction LSTM-F2B

and F0 prediction GRU-F2B-ToBI models with duration and F0 prediciton models from other studies. You can see the results in Table 3.23.

We compared our models with Bi-LSTM models from Fernandez et al. (2014). Their models have a similar archittectur: 3 stacks of bidirectional LSTMs with layr sizes of 67, 57 and 46. Their input consisted of text-based features, such as phonetic identity, POS tags, as well as different counts and context-related features, that were not used in our research. On the other hand, no prosodic labels were used in those models. The biggest differnce, though, comes in the amount of traning data: 7 and 10 hours compared to around 1 and 2 hours for our models. Speakers for both traning sets were female speakers of American English, just like in our study.

Table 3.23. Comparison with other models. RNN1 and RNN2 models and their performance data is taken from Fernandez et al. (2014). The best score for each metric is shown in bold.

Model	Parameter	RMSE	NVAR	Architecture	Training Data
LSTM-F2B	DUR	**0.0221**	0.786	Bi-LSTM	1 hour
GRU-F2B-ToBI	log F0	0.207	0.518	Bi-GRU	
RNN1	DUR	0.0622	**0.872**	Bi-LSTM	7 hours
	log F0	0.037	0.437		
RNN2	DUR	0.1040	0.668	Bi-LSTM	10 hours
	log F0	**0.010**	**0.673**		

When it comes to F0 prediction, our model was significantly outperformed: RMSE of 0.207 against 0.010 (RNN2) and NVAR of 0.518 against 0.673 (RNN2). These huge differences ca be easiliy acounted for by the difference in the amount of training data (1 hour against 10 hour). But RNN2 model significantly outperfmed their other model as well, although in that case the difference in the amount of data is not that big (10 hours agaisnt 7 hours). One possible explanation might be the influence of the style of training data. RNN1 model was trained on news-style data, just like our model. RNN2, on the other hand, was trained on data from variuous genres and domains. Contrary to what was believed, news speech data might not be as good for prosody prediction. An additional investigation comparing prosody prediction based on data from different domains is required.

As for the duration prediction, our LSTM-F2B model showed the best RMSE score of 0.0224, while RNN1 had better NVAR (0.872). RNN2 model, on the other hand, showed the worst score in both cases, atlhough it

had more training data. One possible explanation is that, unlike in case of F0 prediction, news speech data might be more suitable for duration prediction. Or it might also be that after some point, the increase in the amount of training data leads to diminishing returns in performance.

Chapter 4. Automatic Prosody Transplantation

4.1 Data

Trial transplantation experiment was performed on Korean-accented English data. The utterances were taken from the ETRI corpus. They are labelled with TIMIT set, which is the same one that is used in BURNC. Overall, 20 utterances were selected: 5 utterances per speaker (all female, 4 speakers in total). The data did not contain any POS tags, so first, it had to be labelled. We used NLTK library in Python, which contains pre-trained part-of-speech taggers, that are also based on the Penn Treebank tag set. Labels were assigned for each phone (see Figure 4.1). Stress and phoneme type tags were assigned as well, and a sequence of vowels and syllabic consonants was extracted from each utterances.

The data was fed into the duration prediction model first. LSTM-F2B model was used for the trial experiment. After that, the results were used to calculate the number of 10 m-sec frames (per each phone), to be used in the F0 prediction model. All the F0 values were de-normalized to match the mean and deviation of the target speaker. To do that, we calculated the mean and

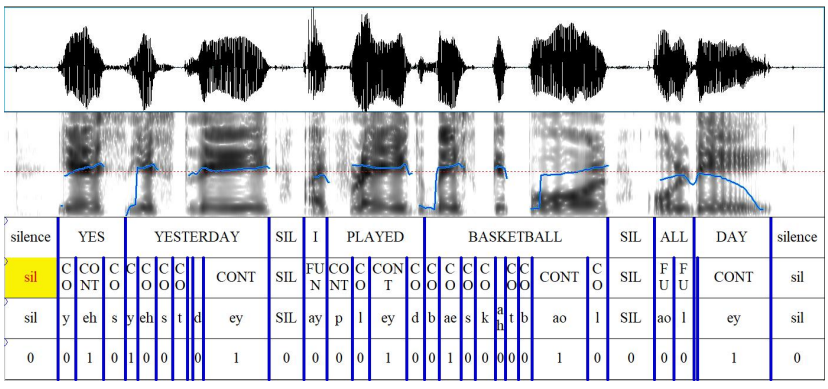


Figure 4.1. Labelled non-native utterance.

Table 4.1. Comparison of original non-native and predicted vowel duration values. The original utterance is: *Yes, yesterday I played basketball all day*. YEHS YEHSTERDEY AY PLEYD BAESKTAHTBAOL AOL DEY.

Phone	Original (m-sec)	Predicted (m-sec)
EH	155	95
EH	101	52
ER	30	106
EY	365	135
AY	129	132
EY	208	107
AE	121	43
AH	54	60
AO	310	44
AO	106	59
EY	404	163

standard deviation of F0 in the utterance. For the F0 prediction LSTM-F3A model was used instead. Although GRU-F2B-ToBI and LSTM-F2B-ToBI showed better results, we do not have a way to predict prosodic labels for the target utterance, and as such this models cannot be used.

You can see the comparison of original non-native and predicted duration and F0 values in Table 4.1 and Table 4.2. In case of duration, some predicted values are bigger, than original ones, while the others are shorter, although majority of the predicted values are significantly shorter, which is expected, as low proficiency foreign speaker's speech is typically slower compared to native speakers. In case of fundamental frequency, the predicted F0 values turned out to be higher than original ones. We can also observe a falling F0 by the end of the utterance, which is characteristic of natural speech.

Table 4.2. Comparison of original non-native and predicted F0 values. Values are averaged for each phone.

Phone	Frame	Original (Hz)	Predicted (Hz)
EH	1	230	253
EH	2	231	253
ER	3	207	254
EY	4	223	250
AY	5	201	248
EY	6	229	247
AE	7	233	236
AH	8	228	235
AO	9	194	219
AO	10	165	208
EY	11	175	221

4.2 Transplantation Method

After that, the predicted values were used for the prosody transplantation. The transplantation was performed using Python's ProMo library. The library allows to perform F0 and duration manipulation using Python code, but the resynthesis itself is performed in Praat. The transplantation was done in 2 stages: first duration transplantation was performed, followed by F0 transplantation.

The duration transplantation is done by comparing two TextGrid files segment by segment. To do that, we first had to create a 'fake' TextGrid' file: it contained all the data from the original utterance's TextGrid, but vowel and syllabic consonant values were switched for the predicted values (see Figure 4.2). Durations within each pair are compared and the difference ratio is

```
File type = "ooTextFile short"
Object class = "TextGrid"

0.0
3.5736305713431444
<exists>
1
"IntervalTier"
"seg"
0.0
3.5736305713431444
32
0.0
0.30109404001578416
"sil"
0.30109404001578416
0.4
"y"
0.4
0.4954436293893194
"eh"
0.4954436293893194
0.6149264787236076
"s"
0.6149264787236076
0.6835996635716458
"y"
0.6835996635716458
0.7365927311416653
```

Figure 4.2. Example of a TextGrid file with predicted duration values.

stored alone with the start and end time points of each segment. Additionally, we introduced a modification limit factor: minimum limit is 0.5 (can only be twice shorter), maximum limit is 2 (can only be twice longer). In case the ratio exceeds the limits, the limit is used instead of it (see Figure 4.3). It is done to minimize the introduction of artefacts, which severely worsens the quality. After that, TD-PSOLA algorithm either adds extra ST signals or removes redundant ones to match the specified ratio (if ratio is 1, no modification is performed). The output of the procedure is a new modified wav file and a new TextGrid file to match it. Python code for the duration transplantation can be find in Appendix 5.

```
def getMorphParameters_ph(fromTGFN, toTGFN, tierName, mod_limit_min,
mod_limit_max):

    fromEntryList = utils.getIntervals(fromTGFN, tierName,
includeUnlabeledRegions=False)
    toEntryList = utils.getIntervals(toTGFN, tierName, includeUnlabeledRegions=
False)

    assert (len(fromEntryList) == len(toEntryList))

    durationParameters = []
    for fromEntry, toEntry in zip(fromEntryList, toEntryList):
        fromStart, fromEnd = fromEntry[:2]
        toStart, toEnd = toEntry[:2]
        toStart += PRAAT_TIME_DIFF
        fromStart += PRAAT_TIME_DIFF
        ratio = (toEnd - toStart) / float((fromEnd - fromStart))
        if ratio < mod_limit_min:
            ratio = mod_limit_min
        elif ratio > mod_limit_max:
            ratio = mod_limit_max
        durationParameters.append((fromStart, fromEnd, ratio))

    return durationParameters
```

Figure 4.3. Python code for comparison of duration between segments.

The next step was F0 transplantation. First, we had to specify which segments from the TextGrid file should receive transplantation, and which should not. This way, we leave segments of the original pitch contour, that we don't want to change, untouched by the morph process. After that we created a list of 'fake' target pitch regions and fill them with predicted F0 values. Then, the two lists are compared with each other and fundamental frequency of the specified pitch regions in the original wav file is modified to match the target values. TD-PSOLA algorithm changes the pitch by moving neighbouring ST signals closer or further apart from each other. It is also possible to limit the scale of modification, in this experiment the limit of 0.5 was implemented. The code for F0 transplantation can be found in Appendix 6.

The original and modified (limit of 0.5) F0 contours can be seen in Figure 4.4 and Figure 4.5. The modified contour differs significantly from the

original one. The original contour contains many drops and rises, while the modified one looks more smooth. That overall makes the utterance to sound more smooth.

	A	B	C	D	E
1	Sample		Accentedness	Comprehensability	
2	1		1	3	
3	2		2	3	
4	3		4	5	
5	4		2	4	
6	5		1	5	
7	6		2	5	
8	7		3	5	
9	8		3	4	
10	9		3	5	
11	10		1	5	
12	11		2	5	
13	12		2	5	
14	13		1	5	
15	14		1	5	
16	15		4	5	
17	16		5	5	
18	17		3	4	
19	18		1	5	
20	19		5	5	
21	20		4	5	
22	21		1	2	
23	22		2	5	
24	23		2	5	
25	24		3	5	
26	25		1	2	
27	26		2	3	
28	27		1	5	
29	28		2	5	
30	29		1	5	
31	30		1	2	

Figure 4.6. Evaluation spreadsheet.

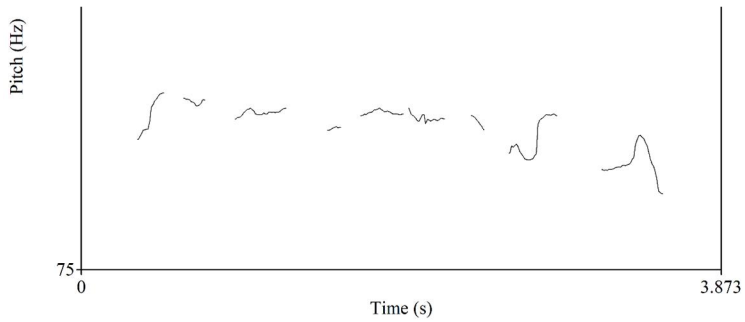


Figure 4.5. Modified F0 contour.

The length of the utterance also changed. It shortened from 4.56 sec (original) to 3.87 sec (modified) due to the reduction in the duration of vowels. That is not surprising, as many of the speakers from ETRI corpus have relatively low level of English proficiency, so their speech is generally slower, compared to native speakers.

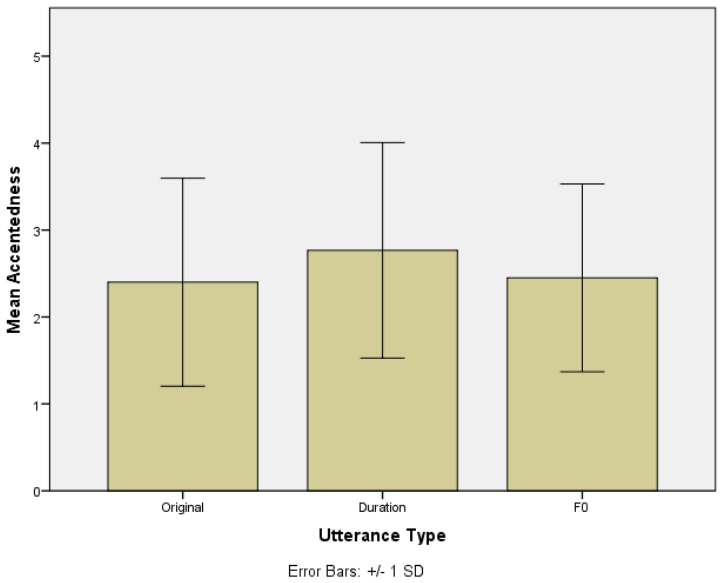


Figure 4.7. Mean accentedness ratings for original and manipulated utterances. Duration = utterances that received duration transplantation only. F0 = utterances that received both duration and F0 transplantation.

4.3 Perceptual Evaluation

To investigate the effectiveness of the proposed transplantation method, a perceptual experiment was carried out.

Listeners. Three native speakers of American English were recruited (two females and one male). All of them have been residing in Seoul, South Korea for over a year.

Materials. 60 utterances were presented to the listeners: 20 original utterances, 20 utterances that only received duration modification, and 20, that received both duration and F0 modification. Each listener had to listen to all of the utterances.

Procedure. The utterances were presented using Microsoft Excel spreadsheet. Each participant sat in front of a monitor and had to click on the audi0 file link to listen to it. Participant could listen to each utterances as

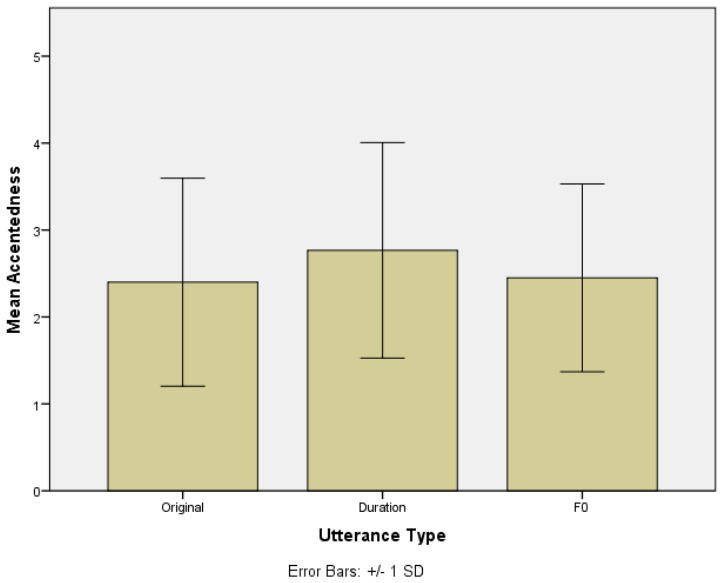


Figure 4.8. Mean comprehensibility ratings for original and manipulated utterances. Duration = utterances that received duration transplantation only. F0 = utterances that received both duration and F0 transplantation.

many times as they wanted, but no transcripts were provided (see Figure 4.6).

The listeners were asked to rate each utterance's accentedness and comprehensibility on a Likert scale from 1 to 5. In case of accentedness, 5 signified "the most native-like", and 1 – "the least native-like". In case of comprehensibility, the scale as well ranged from 1 ("difficult to understand") to 5 ("easy to understand").

Before starting the experiment, the listeners were informed that some utterances were manipulated and might sound artificial. They were asked to ignore artificiality. Furthermore, they were instructed to ignore possible mistakes in consonant production and to pay more attention to vowels and sentence stress, instead.

4.4 Results

A one-way repeated measures analysis of variance (ANOVA) showed no

significant effects of prosody transplantation on accentedness ratings ($F(2, 177) = 1.721$, $p = 0.182$) (see Figure 4.7). Although on average, utterances with duration transplantation only received a better rating of 2.77 compared to the original non-native utterances (2.40) or utterances with both duration and F0 transplantation (2.45).

One possible explanation for the lack of any significant difference might the influence of consonants. Although the listeners were instructed not to pay attention to consonant production, it might not be possible to completely avoid it. One possible solution might be to substitute all consonantal segments with silence or noise, or employ another similar technique.

For comprehensibility, on the other hand, the results were significant ($F(2, 177) = 3.986$, $p < 0.05$) (see Figure 4.8). Utterances with duration transplantation showed the best results (rating of 4.37); slightly better than the original ones (4.33). Bad performance of utterances with both duration and F0 transplantation (rating of 3.87) can be explained by the presence of more severe artefacts, which makes the entire utterance harder to understand.

The results of the transplantation experiment are inconclusive. In case of accentedness, no significant difference was observed; as for the comprehensibility, utterances with modified duration received a slightly better score. The bad rating of utterances with both duration and F0 modification can be attributed to the influence of introduced artefacts. One of the ways to improve the quality is to adjust limit factor. The higher the factor, the closer to the target the modified utterance will sound; but at the same time the quality will degrade. Which means we need to find the optimal trade-off between accentedness and overall quality.

Another possibility is to apply a smoothing technique, that is typically used in concatenative speech synthesis. This will ‘smooth’ the transitions between modified and unmodified segments, which should improve the overall quality

and potentially lead to the increase in comprehensibility rating.

Additionally, a new approach to perceptual evaluation is required, that will allow to filter out the influence of unmodified segments on the overall ratings. First, a study that examines the influence of consonants and vowels on the accentedness and similar ratings.

Chapter 5. Conclusion

5.1 Summary

In this study, we investigated the possibilities of automatic prosody transplantation procedure for self-imitation learning and suggested a new model, that predicts target parameters (vowel duration and F0) instead of extracting them from pre-recorded utterances. The results of this study answered research question proposed in Chapter 1.

In answering the first question, we surveyed prosodic transplantation research done so far, and pointed out the areas, that were still lacking. We also analysed the literature on the characteristics of Korean English, which showed that, when it comes to prosody, Korean learners struggle the most with sentence stress, especially with the reduction of unstressed function words. As such we suggested a selective transplantation strategy, when only the duration

and F0 of vowels would be transplanted onto non-native speech, to emphasize the possible areas of mistake.

The next research question was what kind of input features should be used in a prosody prediction model. We analysed the data from Boston University Radio News Corpus. The results showed that basic features that can be obtained automatically, like phoneme type, stress, POS tags, can be good predictors of vowel duration and F0.

Then in Chapter 3 we described our proposed prosody prediction models. The models are based on RNN, LSTM and GRU architecture, which utilize the context around the target. This allows to get rid of many contextual input features, traditionally used, as they become redundant. All proposed models were trained on small amount of data and then compared to state-of-the-art prosody prediction models from speech synthesis, trained on larger amount of data. Although our models could not compare with them in terms of performance, nevertheless, considering the huge difference in the amount of training data and input features, the results of this study are still promising.

We also compared models with prosodic labels (F2B-ToBI model) with those without, and the results showed, that omission of prosodic labels does not lead to any significant drop of performance in case of duration prediction. In case of F0 prediction, however, models without ToBI labels did significantly worse and the results are not satisfactory. That means that automatically derived features can be sufficient for duration transplantation system. If the aim is to transplant F0, a reliable way to automatically assign prosodic labels is first required. For the duration transplantation, the proposed models seem satisfactory (LSTM-F2B model).

Finally, in Chapter 4 we showed how our model can be applied in practice. We performed selective prosody transplantation using predicted vowel duration and F0 values (LSTM-F2B and LSTM-F3A models). The results

lead to significant change in the overall length of utterance as well as individual phonemes. The shape of pitch contour changed as well. The results were evaluated by native speakers of English. No significant changes in accentedness ratings were observed; duration transplantation only lead to a slight improvement in comprehensibility rating. Both duration and F0 transplantation lead to decrease in comprehensibility, and as such F0 transplantation requires significant additional refinements, before it can be implemented in any CAPT environment.

5.2 Contribution

The results of this study can contribute significantly to the prosody transplantation research. We introduced a new automatic method of transplantation. The proposed prediction model can be used instead of pre-recording native utterances. Duration prediction model's performance was comparable to other existing models, used in speech synthesis, while our model used a significantly less amount of input features, that can be automatically obtained. Implementation of the model to the duration transplantation will remove the need to record each sentence with a native speaker. That can significantly cut down the costs of a self-imitation training system, that employs prosody transplantation.

Additionally, when it comes to prosody transplantation procedure, we proposed a new selective transplantation method. We consider selective transplantation to be more effective for self-imitation prosody training, as it allows the learner to concentrate of potentially problematic areas, when listening to feedback. The proposed duration prediction model can be used in an online CAPT system designed for Korean learners of English, especially

when learning sentence stress or other rhythm-related phenomena.

5.3 Limitations

This study had a number of limitations. The research was limited to the Korean-English language pair, although we believe, that the results can be extrapolated to other languages. The proposed models were trained on small-scale corpus of English. As such, the results were not as good, as they could have been when using more data.

We used data from two female speakers, based on the amount of speech available. As such, there was no criteria for speaker selection, although that might be one of the most crucial stages in prosody transplantation. Introduction of acoustic or similar criteria for choosing a perfect native voice, might be necessary. Additionally, only female voices were used for this experiment. Male voices are known to show narrower pitch range and less prosodic variability in general. A study, that compares the use of female and male voice for automatic prosody transplantation is required.

Finally, only a small-scale perceptual evaluation was carried out. An additional evaluation with more participants and more material can give a better result. Additionally, we were not able to limit the influence of unmodified consonantal segments on the accentedness and comprehensibility ratings.

5.4 Recommendations for Future Study

The proposed model needs additional testing and refinement. More specifically, the model should be tested in a real learning environment. The results of the automatic transplantation experiment can be used as feedback in prosody training session. The speech of Korean learners of English then will

be recorded and evaluated before and the after the training and analysed for any possible improvement. This will give a more objective evaluation of the model's performance, compared to a perceptual listening experiment.

Also, as was pointed out in the previous section, a number of models can be trained on a larger dataset of both male and female speech, and then compared with each other. Additionally, some kind of objective criteria for native speaker selection should be introduced.

For the F0 prediction model, a way to improve the performance without the inclusion of prosodic labels must be investigated. Alternatively, an automatic assignment of prosodic labels can be investigated and implemented, which can benefit both F0 and duration prediction models.

The optimal limit factor for transplantation must be found as well. Transplantation can be applied to the same utterance, but with a different limit factor, and the utterances then can be compared with each other in a perceptual evaluation or similar experiment.

References

[1] Abadi, M. et al. (2016). TensorFlow: A System for Large-Scale Machine Learning. *OSDI'16 Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation*, pp. 265-283.

[2] Abercrombie, D. (1967). Elements of General Phonetics. Edinburgh University Press, Edinburgh.

[3] Aryal, S., Felps, D., and Gutierrez-Osuna, R. (2013). Foreign accent conversion through voice morphing. *In INTERSPEECH-2013*, pp. 3077-3081.

[4] Aryal, S., and Gutierrez-Osuna, R. (2014). Accent Conversion through Cross-Speaker Articulatory Synthesis. *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7694-7698.

[5] Aryal, S., and Gutierrez-Osuna, R. (2015). Reduction of non-native accents through statistical parametric articulatory synthesis. *Acoustical Society of America*, 137(1), pp. 4330446.

[6] Auer, P. (1991). 'Stress-timing' vs. 'syllable-timing' from a typological point of view. In B. Palek & P. Janota (Eds.). *Proceedings of the conference linguistics and phonetics: prospects and applications*, pp. 292–305.

[7] Bernardy, J.P., Themistocleous, C. (2017). *Modelling prosodic structure using Artificial Neural Networks. Proceedings of 8th Tutorial and Research Workshop on Experimental Linguistics*.

[8] Boersma, P. (2001). Praat, a system for doing phonetics by computer. *Glot International,* 5 (9/10), pp. 341-345.

[9] Bonneau, A., Colotte. V. (2011). Automatic Feedback for L2 Prosody

Learning. Ivo Ipsic. *Speech and Language Technologies*, Intech, pp.55-70.

[10] Boula de Mareuil, P., Marotta, G., and Adda-Decker, M. (2004). Contribution of prosody to the perception of Spanish/Italian accents. *In SP-2004,* pp. 681-684.

[11] Charpentier, F., Stella, M. (1986). Diphone synthesis using an overlap-add technique for speech waveforms concatenation. *ICASSP '86. IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 2015-2018.

[12] Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *Proceedings of the EMNLP 2014*, pp. 1724-1734.

[13] De Meo, A., Vitale, M., Pettorino, M., Cutugno, F., and Origlia, A. (2013). Imitation/self-imitation in computer-assisted prosody training for Chinese learners of L2 Italian. In. J. Levis & K. LeVelle (Eds.). *Proceedings of the 4th Pronunciation in Second Language Learning and Teaching Conference*, pp. 90-100.

[14] Deng, L., and Yu, D. (2014). Deep Learning: Methods and Applications. Now Publishers Inc., Hanover, MA, USA.

[15] Ding, C., Xie, Lei, Yan, J., Zhang, W., & Liu, Y. (2015). Automatic Prosody Prediction for Chinese Speech Synthesis using BLSTM-RNN and Embedding Features. *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 98-102.

[16] Eskenazi, M. (2009). An overview of spoken language technology for education. *Speech Communication*, 51, pp. 832-844.

[17] Felps, D., Bortfeld, H., & Gutierrez-Osuna, R. (2009). Foreign accent conversion in computer assisted pronunciation training. *Speech*

Commun. 51, 10, pp. 920-932.

[18] Felps, D., Geng, C., Gutierrez-Osuna, R. (2012). Foreign Accent Conversion through Concatenative Synthesis in the Articulatory Domain. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(8), pp. 2301-2312.

[19] Fernandez, R., Rendel, A., Ramabhadran B., & Hoory, R. (2013). F0 contour prediction with a deep belief network-Gaussian process hybrid model. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6885-6889.

[20] Fernandez, R., Rendel, A., Ramabhadran B., & Hoory, R. (2014). Prosody Contour Prediction with Long Short-Term Memory, Bi-Directional, Deep Recurrent Neural Networks. In *INTERSPEECH-2014*, pp. 2268-2272.

[21] Garbe, K., Glowka, A. (2017). Modeling intonation using a bidirectional Long Short-term Memory Recurrent Neural Network. Ms., Stanford University.

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

[23] Gu, H.Y., Lai, M.Y., Tsai, S.F. (2010). Combining HMM spectrum models and ANN prosody models for speech synthesis of syllable prominent languages. *7th International Symposium on Chinese Spoken Language Processing,* pp. 451-454.

[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] Hochreiter, S., and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Comput.*, 9(8), pp. 1735-1780.

[27] Jilka, M., Mohler, G. (1998). Intonational foreign accent: speech technology and foreign language teaching. *In: Proc. ESCA Workshop on Speech Technology in Language Learning*, pp. 115–118.

[28] Jugler, J., Zimmerer, F., Trouvain, J., Möbius, B. (2016). The Perceptual Effect of L1 Prosody Transplantation on L2 Speech: The Case of French Accented German. Proc. Interspeech 2016, pp.67-71.

[29] Jun, S. (2009). Prosody in sentence processing. In P. Li (Author) & C. Lee, G. Simpson, & Y. Kim (Eds.). *The Handbook of East Asian Psycholinguistics*, pp. 423-432.

[30] Kaglik, A., Boula de Mareuil, P. (2010). Polish-accented French prosody in perception and production: transfer or universal acquisition process? 5th International Conference on Speech Prosody, 2010, pp. 1-4.

[31] Kang, S., Ahn, H., Hong, M. (2012). The Acquisition of L2 English Focus by Korean Learners. *Korean Journal of Linguistics*, 37(1), pp. 1-23.

[32] Kim, J. (2005). Stress assignment rule in Korean English. 음성음운형태론연구, 11 (2), pp. 71-82.

[33] Kim, J., Flynn, S. (2004). What makes a non-native accent? a study of Korean English. *Proceedings of the 8th International Conference on Spoken Language Processing*, ICSLP/INTERSPEECH-2004, pp. 1845-1848.

[34] Kingma, D.P., Lei Ba, J. (2015). Adam: A Method for Stochastic Optimization. *3rd International Conference for Learning Representations ICLR-2015*.

[35] Lee, B., Guion, S. G., and Harada, T. (2006). Acoustic analysis of

the production of unstressed English vowels by early and late Korean and Japanese bilinguals. *Studies in Second Language Acquisition*, 28(3), pp. 487-513.

[36] Marcus, M.P., and Santorino, B. (1993). Building a Very Large Natural Language Corpora: The Penn Treebank. *Computational Linguistics*, 19(2), pp. 313-330.

[37] Moulines, E., Charpentier, F. (1990). Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones. *Speech Communication*, 9 (5–6), pp. 453-467.

[38] Murray, G.L. (1999). Autonomy and language learning in a simulated environment. *System*, 27(3), pp. 295-308.

[39] Ostendorf, M., Price, P., and Shattuck-Hufnagel, S. (1996). Boston University Radio Speech Corpus LDC96S36. DVD. Philadelphia: Linguistic Data Consortium.

[40] Park, H. (2013). A study of an Independent Evaluation of Prosody and Segmentals: With Reference to the Difference in the Evaluation of English Pronunciation across Subject Groups. 말소리와 음성과학 제 5 권 제 4 호, pp. 091-098.

[41] Pellegrino, E. (2012). The perception of foreign accented speech. Segmental and suprasegmental features affecting the degree of foreign accent in L2 Italian. *Proceeding of the VIIth GSCP International Conference: Speech and Corpora*, pp 261-267.

[42] Pellegrino, E., Vigliano, D. (2015). Self-imitation in prosody training: a study on Japanese learners of Italian. *In SLaTE-2015*, pp. 53-57.

[43] Pettorino, M., and Vitale, M. (2012). Transplanting native prosody into second language speech. In M. G. Busà and A. Stella (eds.), *Methodological Perspectives on Second Language Prosody: Papers*

from ML2P 2012, pp. 11-16.

[44] Probst, K., Ke, Y., & Eskenazi, M. (2002). Enhancing foreign language tutors - in search of the golden speaker. *Speech Communication*, 37(3-4), pp. 161-173.

[45] Rognoni, L., Grazia Busa, M. (2013). Testing the Effects of Segmental and Suprasegmental Phonetic Cues in Foreign Accent Rating: An Experiment Using Prosody Transplantation. Proc. International Symposium on the Acquisition of Second Language Speech, pp. 547-560.

[46] Sereno, J., Lammers, L., and Jongman, A. (2016). The relative contribution of segments and intonation to the perception of foreign-accented speech. *Applied Psycholinguistics*, 37(2), pp. 303-322.

[47] Silverman, K., Beckman, M., Pitrelli, J., Ostendorf, M., Wightman, C., Price, P., Pierrehumbert, J., Hirschberg, J. (1992). TOBI: a standard for labeling English prosody. *In Proceedings of the International Conference on Spoken Language Processing (ICSLP-1992)*, pp. 867-870.

[48] Sheikhan, M. (2017). Improvement of Prosody Modeling Using Semantic Role labeling, Hybrid Feature Selection and BPSO-PSO-Optimized RNN. *Natl. Acd. Sci. Lett.*, 40(3), pp. 171-175.

[49] Shreekanth, T., Udayashankara, V., Chandrika M. (2015). Duration Modelling Using Neural Networks for Hindi TTS System Considering Position of Syllable in a Word. *Procedia Computer Science*, 46, pp. 60-67.

[50] Sreenivasa Rao, K., Yegnanarayana, B. (2007). Modeling durations of syllables using neural networks. *Computer Speech and Language*, 21, pp. 282–295.

[51] Su, H., Zhang, H., Zhang, X., Gao, G. (2016). Convolutional neural

network for robust pitch determination. *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 579-583.

[52] Sundstrom, A. (1998). Automatic prosody modification as a means for foreign language pronunciation training. *In: Proc. ISCA Workshop on Speech Technology in Language Learning (STILL 98)*, pp. 49–52.

[53] Um, H. (2004). The English Intonation of Native Speakers and Korean Learners: A Comparative Study. *Speech Sciences*, 11(1), pp. 117-130.

[54] Valbret, H., Moulines, E., Tubach, J.P. (1992). Voice transformation using PSOLA technique. *Speech Communication*, 11, pp. 175-187.

[55] Yoo, H. (2012). Acquisition of English Sentence Stress by Korean EFL Learners. 영어학연구, 18(1), pp. 75-101.

[56] Watson, C., Kewley-Port, D. (1989). Advances in computer-based speech training: Aids for the profoundly hearing impaired. *Volta-Review*, 91, pp. 29–45.

[57] Yoo, H. (2017). Comprehensibility of Korean EFL speakers' English pronunciation. *Studies in Phonetics, Phonology and Morphology*, 23(1), pp. 95-115.

[58] Yoon, K. (2007). Imposing native speakers' prosody on non-native speakers' utterances: The technique of cloning prosody. *Journal of the Modern British and American Language & Literature*, 25 (4), pp. 197-215.

[59] Zhao, S., Koh, S.N., and Luke, K.K. (2012). Accent reduction for computer-aided language learning. *In Proceedings of the 20th European Signal Processing Conference (EUSIPCO),* pp. 335-339.

[60] Zue, V., Seneff, S. (1996). Transcription and alignment of the TIMIT database. In Hiroya Fujisaki (Ed.), *Recent research toward advanced*

man-machine interface through spoken language, pp. 464-447.

[61] 규철, 윤. (2011). 운율 복제 기법을 이용한 영어 운율 교육. 영어영문학연구, 37(1), 245-272.

Appendix

Appendix 1. TIMIT phoneme set.

	Phone	Example		Phone	Example		Phone	Example
1	iy	b*ee*t	22	r	*r*ay	43	jh	*j*oke
2	ih	b*i*t	23	w	*w*ay	44	ch	*ch*oke
3	eh	b*e*t	24	y	*y*acht	45	b	*b*ee
4	ey	b*ai*t	25	hh	*h*ay	46	d	*d*ay
5	ae	b*a*t	26	hv	a*h*ead	47	g	*g*ay
6	aa	bo*b*	27	el	bott*le*	48	p	*p*ea
7	aw	b*ou*t	28	m	mo*m*	49	t	*t*ea
8	ay	b*i*te	29	n	moo*n*	50	k	*k*ey
9	ah	b*u*t	30	ng	si*ng*	51	dx	mu*dd*y
10	ao	b*ough*t	31	em	bott*om*	52	q	glottal stop
11	oy	b*oy*	32	en	butt*on*	53	bcl	b closure
12	ow	b*oa*t	33	eng	Wash*ing*ton	54	dcl	d closure
13	uh	b*oo*k	34	nx	wi*nn*er	55	gcl	g closure
14	uw	b*oo*t	35	s	*s*ea	56	pcl	p closure
15	ux	t*oo*t	36	sh	*sh*e	57	tcl	t closure
16	er	b*ir*d	37	z	*z*one	58	kcl	k closure
17	ax	*a*bout	38	zh	a*z*ure	59	epi	epenthetic silence
18	ix	deb*i*t	39	f	*f*in	60	pau	pause
19	axr	butt*er*	40	th	*th*in	61	h#	begin/end marker
20	ax-h	s*u*spect	41	v	*v*an			
21	l	*l*ay	42	dh	*th*en			

Appendix 2. Part-of-speech tags (excluding 11 punctuation labels)

used in labelling the corpus, from the Penn Treebank set.

Tag	**Part of speech**	**Tag**	**Part of speech**
CC	coordinating conjunction	PP$	possessive pronoun
CD	cardinal number	RB	adverb
DT	determiner	RBR	adverb, comparative
EX	existential *there*	RBS	adverb, superlative
FW	foreign word	RP	particle
IN	preposition/ subordinating conjunction	SYM	mathematical symbol
JJ	adjective	TO	*to*
JJR	adjective, comparative	UH	interjection
JJS	adjective, superlative	VB	verb, base form
LS	list item marker	VBD	verb, past tense
MD	model	VBG	verb, gerund or present participle
NN	noun, singular or mass	VBN	verb, past participle
NNS	noun, plural	VBP	verb, non-3rd person singular present
NP	proper noun, singular	VBZ	verb, 3rd person singular present
NPS	proper noun, plural	WDT	wh-determiner
PDT	pre-determiner	WP	wh-pronoun
POS	possessive ending	WP$	possessive wh-pronoun
PP	personal pronoun	WRB	wh-adverb

Appendix 3. Python code for model architecture (RNN).

```
def build_rnn(batch_size, bi_lstm_size, learning_rate):
    tf.reset_default_graph()
    # Declare placeholders we'll feed into the graph
    inputs = tf.placeholder(tf.float32, [batch_size, max_time_step, embedding_size],
name='inputs')
    length = tf.placeholder(tf.int32, [batch_size], name='length')
    durations = tf.placeholder(tf.float32, [batch_size, max_time_step, 1],
name='durations')
    keep_prob = tf.placeholder(tf.float32, name='keep_prob')
    # Build uni-LSTM cell
    def lstm_cell(lstm_size):
        lstm = tf.nn.rnn_cell.LSTMCell(lstm_size, use_peepholes=True,
initializer=tf.contrib.layers.xavier_initializer(), activation=tf.tanh,
reuse=tf.get_variable_scope().reuse)
        return tf.contrib.rnn.DropoutWrapper(lstm, output_keep_prob=keep_prob)

    cell_fw = [lstm_cell(ls) for ls in bi_lstm_size]
    cell_bw = [lstm_cell(ls) for ls in bi_lstm_size]
    initial_state_fw = [cfw.zero_state(batch_size, dtype=tf.float32) for cfw in
cell_fw]
    initial_state_bw = [cbw.zero_state(batch_size, dtype=tf.float32) for cbw in
cell_bw]
    outputs_bi, final_state_fw, final_state_bw =
tf.contrib.rnn.stack_bidirectional_dynamic_rnn (cells_fw=cell_fw, cells_bw=cell_bw,
inputs=inputs, initial_states_fw=initial_state_fw, states_bw=initial_state_bw,
sequence_length=length, scope = 'bi_lstm')


    # Make the predictions
    outputs = tf.contrib.layers.fully_connected(outputs_bi, num_outputs=1,
activation_fn= None, weights_initializer=tf.contrib.layers.xavier_initializer(),
biases_initializer=  tf.zeros_initializer())


    # Calculate the cost
    with tf.name_scope('loss'):
        loss = tf.reduce_mean(tf.losses.mean_squared_error(labels = durations,
predictions = outputs))
        tf.summary.scalar('loss', loss)


    # Train the model
    with tf.name_scope('optimizer'):
        optimizer = tf.train.AdamOptimizer(learning_rate).minimize(loss)
    # Merge all of the summaries
    merged = tf.summary.merge_all()

    # Export the nodes
    export_nodes = ['inputs', 'length', 'durations', 'keep_prob', 'outputs', 'loss',
'optimizer', 'merged']
    Graph = collections.namedtuple('Graph', export_nodes)
    local_dict = locals()
    graph = Graph(*[local_dict[each] for each in export_nodes])

    return graph
```

Appendix 4. Python code for data pre-processing.

```
data = np.loadtxt(filename_1, delimiter = ' ', dtype = 'str', unpack = True)
phone_dic = np.loadtxt(filename_2, delimiter = ' ', dtype = 'str', unpack = True)
v_type_dic = np.loadtxt(filename_3, delimiter = ' ', dtype = 'str', unpack = True)
#Reading data
phone_dic = dict(zip(phone_dic[0], np.transpose(phone_dic[1:])))
v_type_dic = dict(zip(v_type_dic[0], np.transpose(v_type_dic[1:])))
#Getting rid of outliers and normalizing lable data to (0, 1)
durs_4 = [float(x) for x in data[8]]
m = np.mean(durs_4) # Used  to restore original data by x * m
std = np.std(durs_4)
q3, q1 = np.percentile(durs_4, [75, 25])        #Checking for outliers using
interquartile percentile;
iqr = q3 - q1
durs_3 = []
i,j = 0, 0
for x in durs_4:
    if x > q3 + 1.5 * iqr:
        x = q3 + 1.5 * iqr
    elif x < q1 - 1.5 * iqr:
        x = q1 - 1.5 * iqr
    durs_3.append(x)
durs_3 = stats.zscore(durs_3)   #Doing z-score normalization
max_dur = max(durs_3)
min_dur = min(durs_3)
#Norlmalizing duration for [0.01, 0.99] range - optimal for neural networks
durs_3 = [ ((0.99 - 0.01) * (x - min_dur) / (max_dur - min_dur) + 0.01) for x in
durs_3]
```

Appendix 5. Duration transplantation code

```
PRAAT_TIME_DIFF = 0.000001


# Some convenience functions -- we'll be using these a lot
def pitchForPlots(pitchFN):
    pitchTier = dataio.open2DPointObject(pitchFN)
    x, y = zip(*pitchTier.pointList)
    return x, y


def doPlot(axis, title, pitchFN):
    axis.plot(*pitchForPlots(pitchFN))
    axis.set_title(title)
    axis.set_xlabel("time(s)")
    axis.set_ylabel("F0(hz)")

vowels = ['EN', 'EM', 'ENG' 'EL', 'ER', 'AXR', 'AH', 'AX', 'AE', 'AO', 'AA', 'IH', 'IY',
'EH', 'UW', 'UH', 'AY', 'EY', 'OW', 'AW', 'OY']

#No modifiation of of stops and affricates
def getMorphParameters_ph(fromTGFN, toTGFN, tierName, mod_limit):

    fromEntryList = utils.getIntervals(fromTGFN, tierName,
includeUnlabeledRegions=False)
    toEntryList = utils.getIntervals(toTGFN, tierName, includeUnlabeledRegions= False)

    #fromEntryList = [entry for entry in fromEntryList]
    #toEntryList = [entry for entry in toEntryList]
    assert (len(fromEntryList) == len(toEntryList))

    durationParameters = []
    for fromEntry, toEntry in zip(fromEntryList, toEntryList):
        fromStart, fromEnd = fromEntry[:2]
        toStart, toEnd = toEntry[:2]

        toStart += PRAAT_TIME_DIFF
        fromStart += PRAAT_TIME_DIFF

        #Introduce a new function that will look at phoneme type
        #fromEntry[2], toEntry[2] is phoneme type
        #Make a dictionary of phones and phoneme types
        if fromEntry[2].upper() in vowels:
            ratio = float((toEnd - toStart)) / float((fromEnd - fromStart))
            if ratio > 1 + mod_limit:
                ratio = 1 + mod_limit
            elif ratio < 1 - mod_limit:
                ratio = 1 - mod_limit
        else:
            ratio = 1.0

        durationParameters.append((fromStart, fromEnd, ratio))

    return durationParameters
```

```
def changeDuration_ph(fromWavFN, durationParameters, outputName, outputMinPitch,
outputMaxPitch, praatEXE, outputPath, toTGFN):

    rootPath = os.path.split(fromWavFN)[0]

    # Prep output directories
    outputPath = outputPath
    durationTierPath = outputPath

    fromWavDuration = audio_scripts.getSoundFileDuration(fromWavFN)

    durationParameters = copy.deepcopy(durationParameters)
    # Pad any gaps with values of 1 (no change in duration)

    durationPointList = []
    for start, end, ratio in durationParameters:
        durationPointList.append((start, ratio))
        durationPointList.append((end, ratio))

    outputPrefix = "%s" % (outputName)
    durationTierFN = join(durationTierPath, "%s.DurationTier" % outputPrefix)
    outputWavFN = join(outputPath, "%s.wav" % outputPrefix)
    durationTier = dataio.PointObject2D(durationPointList, dataio.DURATION, 0,
fromWavDuration)
    durationTier.save(durationTierFN)

    #Saving a TextGrid file
    tg = tgio.openTextgrid(toTGFN)
    ph_tier = tg.tierDict["seg"]
    phones = [ph for _, _, ph in ph_tier.entryList]
    st = [s for s, _, _ in ph_tier.entryList]
    en = [e for _, e, _ in ph_tier.entryList]
    x = np.loadtxt(durationTierFN, dtype='float', skiprows=6)
    st_m = []
    en_m = []
    j = 0
    l = 0
    i = 0
    while i < len(x) - 1:
        if i == 0:
            j = 0
        else:
            j = x[i] * x[i + 1]
        i += 2
        k = x[i] * x[i + 1]
        i += 2
        st_m.append(l)
        l = l + (k - j)
        en_m.append(l)
    dr = []
    for i, x in enumerate(phones, 0):
        dr.append((st_m[i], en_m[i], x))
    new_ph_tier = ph_tier.new(entryList=dr)
    new_tg = tgio.Textgrid()
    new_tg.addTier(new_ph_tier)
    new_tg.save(join(outputPath, "%s.TextGrid" % outputPrefix))
```

```
    praat_scripts.resynthesizeDuration(praatEXE, fromWavFN, durationTierFN,
outputWavFN, outputMinPitch, outputMaxPitch)



praatEXE = r"C:/Users/Matvei/Desktop/praat6023_win64/Praat.exe"  # Windows

minPitch = 50
maxPitch = 600

# Define the arguments for the code
root_1 = "C:/Users/Matvei/Desktop/evaluation/originals/"
root_2 = "C:/Users/Matvei/Desktop/evaluation/pred_values/"
outputPath = "C:/Users/Matvei/Desktop/evaluation/results/"

fld = r'C:WUsersWMatveiWDesktopWevaluationWoriginalsW*'
for file in glob.glob(fld):
    if 'data' in file:
        print(file)
        fromName = file[-23:-9]
        fromWavFN = root_1 + fromName + ".wav"

        tierName = "seg"

        fromTGFN = root_1 + fromName + ".TextGrid"
        toTGFN = root_2 + fromName + "_dur_mod.TextGrid"
        mod_limit = 0.5
        utils.makeDir(outputPath)
        outputName = "%s_dur" % (fromName)
        outputTG = join(outputPath, "%s_dur.TextGrid" % outputName)
        durationParams = getMorphParameters_ph(fromTGFN, toTGFN, tierName, mod_limit)


        changeDuration_ph(fromWavFN,
                          durationParams,
                          outputName,
                          outputMinPitch=minPitch,
                          outputMaxPitch=maxPitch,
                          praatEXE=praatEXE,
                          outputPath = outputPath,
                          toTGFN = toTGFN)
```

Appendix 6. Fundamental frequency transplantation code

```
praatEXE = r"C:₩Users₩Matvei₩Desktop₩praat6023_win64₩Praat.exe"  # Windows paths

minPitch = 75
maxPitch = 600
vowels = ['EN', 'EM', 'ENG' 'EL', 'ER', 'AXR', 'AH', 'AX', 'AE', 'AO', 'AA', 'IH', 'IY',
'EH', 'UW', 'UH', 'AY', 'EY', 'OW', 'AW', 'OY']


fld = r'C:₩Users₩Matvei₩Desktop₩evaluation₩originals₩*'
for file in glob.glob(fld):
    if 'data' in file:
        print(file)
        name = file[-23:-9]
        inputWavFN = root_1 + name + "_dur.wav"
        pitchFN = root_1 + name + '_dur.PitchTier'

        filename_1 = root_2 + name + "_pred_f0.txt"
        filename_2 = root_1 + name + '_dur.TextGrid'


        fromPitchTier = pitch_and_intensity.extractPitchTier(inputWavFN, pitchFN,
praatEXE, minPitch, maxPitch, forceRegenerate=False)

        tg = tgio.openTextgrid(filename_2)
        ph_tier = tg.tierDict["seg"]

        cv = f0_morph.getPitchForIntervals(fromPitchTier.pointList, filename_2, 'seg')
        fromPitchRegions = []
        toPitchRegions = []
        for i, (x, y, z) in enumerate(ph_tier.entryList):
            if z.upper() in vowels:
                fromPitchRegions.append(cv[i])


        f0 = np.loadtxt(filename_1, delimiter = ' ', dtype = 'float', usecols= (0, 2),
unpack = True)
        j = 0
        ch = []
        y = [fromPitchRegions[i][0][0] for i in range(len(fromPitchRegions))]
        k = y[j]
        for i, x in enumerate(f0[0]):
            if int(x) == j:
                ch.append((k, f0[1][i]))
            else:
                toPitchRegions.append(ch)
                j += 1
                k = y[j]
                ch = []
                ch.append((k, f0[1][i]))
            if i == len(f0[0])-1:
                toPitchRegions.append(ch)
            k += 0.01

        stepList = [0.5]
```

```
f0_morph.f0Morph(fromWavFN= inputWavFN,
                 pitchPath = output,
                 stepList=stepList,
                 outputName= name,
                 doPlotPitchSteps=False,
                 fromPitchData=fromPitchRegions,
                 toPitchData=toPitchRegions,
                 outputMinPitch=minPitch,
                 outputMaxPitch=maxPitch,
                 praatEXE=praatEXE,
                 keepPitchRange=True,
                 keepAveragePitch=True,
                 sourcePitchDataList=fromPitchTier.pointList)
```

요약(국문초록)

지난 수십년동안 컴퓨터를 사용하여 외국어의 발음을 가르치는 것은 급속히 증가하고 있었다. 이러한 컴퓨터 보조 발음 훈련 (CAPT - computer-assisted pronunciation training) 시스템들이 주로 정확한 음소 발음을 가르침에만 줍중했는데 운율은 별로 관심을받지 못했다. 운율 훈련에 대한 새로운 접근법 중 하나는 자기 모방 (self-imitation) 학습이다. 원어민의 발화에서 운율적 요소를 학습자의 발화로 복제하고 수정적 피드백으로 학습자에게 다시 제공해 주는 것이다. 이 기술의 제일 큰 단점은 원어민과 학습자의 똑 갘은 발화 두 개 꼭 필요한 것이다.

운율 복제의 새로운 방법을 개발하기위한 예비 연구인 제 1 장은 선행 연구를 조사하고 장점과 단점을 지적한다. 또한 한국어와 영어의 운율 체계를 비교하고, 한국인 학습자가 주로 하는 실수을 지적한 다음에 그런 실수와 관련있는 음향 특성을 분석한다. 모음 길이와 기본 주파수의 복제는 한국인 영어 학습자에게 제일 효과적이라고 제안한다.

본 연구의 두 번째 부분은 새로운 운율 복제의 기법을 소개한다. 미리 녹음된 원어민 발화에서 음향 값을 복제하는 대신에 심층신경망(DNN)을 사용하는 예측 모데를 제안한다. RNN (recurrent neural network), LSTM (long short-term memory) 및 GRU (gated recurrent unit) 세 가지 예측 모델을 기술하며 설명한다. 모델들은 Boston University Radio Speech Corpus로 훈련시켰는데 성능은 음성합성 연구의 최첨단 운율 예측 시스템 또한 서로와도 비교했다.

제안한 운율 예측 모델을 이용하는 자동 운율 복제의 실행 방법을 설명하며 그의 결과를 분석한다. 영어 원어민에 의한 지각 평가 실험을 수행했다. 원래 한국인 학습자의 발화와 운율 복제를 받은 발화의 말투 및 이해도를 서로와 비교했다. 그의 결과는 모음 길이 복제가 이해드를 향상시킬 수 있음을 보여주었다. 본 연구는 완전 자동화된 자기 모방 운율 훈련 시스템의 초석을 마련한다. 본 연구의 결과는 한국인 영어 학습자가 문장 스트레스와 같은 영어 운율의 문제 영역을 마스터하는 데에 도움이 될 수 있다.

주요어: 컴퓨터 보조 발음 훈련, 한국인의 영어 운율, 운율 복제, 운율 예측, 심층 신경망.

학 번 : 2015-23283