A Diphone-Based Maltese Speech Synthesis System

Review Report

Daniel Magro, supervised by Dr Claudia Borg and Dr Andrea De Marco
Department of Artificial Intelligence, Faculty of ICT, University of Malta
Email: { daniel.magro.15 | claudia.borg | andrea.demarco } @ um.edu.mt

ABSTRACT

In Malta, there are 7,100 vision-impaired (1.9% of the Maltese population), and over 24,000 illiterate (6.4% of the Maltese population), Maltese speakers [1]. These people are unable to consume any content written in Maltese, be it a book, a news article, or even a simple Facebook post. This dissertation sets out to solve that problem by developing a Text to Speech (TTS) system for the Maltese language.

At the time of writing, there were no available prior works to build on, nor databases of recorded speech. The type of TTS system that was developed thus had to be chosen with these limitations in mind. The type of TTS system that was best suited for this scenario was a Diphone-Based Concatenative Speech Synthesiser.

A simple Text Normalisation component was developed, along with a Grapheme to Phoneme (G2P) based on Crimsonwing's G2P program [6]. Three separate diphone databases were made available to the system, along with an implementation of Time Domain - Pitch Synchronous OverLap Add (TD-PSOLA) for smoothly concatenating the diphones from the database in the sequence specified by the G2P component.

Keywords

Speech Synthesis, Maltese Language, Language Technology

1. INTRODUCTION

The aim of this thesis is to transform any given string of text in the Maltese Language into a clear and intelligible synthesised voice. Different well-established speech synthesis methods will be explored, and one will be selected based on the resources available (linguistic datasets), computational resources available, amount of training data required and typical output quality. Furthermore, some problems which are common to many speech synthesis systems will be addressed, for example, handling of Non-Standard Words (NSWs).

1.1 Problem Definition

TTS, or Speech Synthesis, is the conversion of a written text in a particular language into speech.

Work on the topic has been carried out in a PhD thesis by P. Micallef [21], however the developed TTS system was not found to be available. The Foundation for Information Technology Accessibility (FITA) also collaborated with Crimsonwing between 2009 and 2012 to develop a Maltese TTS system [19]. This TTS system, however, requires an account to download[1], for which registrations were closed at the time of writing. Furthermore, the solution is 1.1 GB in size, and is only available for Windows, which severely limits its portability to different platforms, especially smartphones. Therefore, there are no available TTS synthesis systems for Maltese.

Maltese is a low-resourced language, meaning very few resources exist for it. There are currently no databases of recorded and labelled Maltese speech available. The Maltese TTS system must be designed with these very minimal resources in mind.

1.2 Motivation

Speech Synthesis Systems, in general, are not just a convenience for the many but also an enabler for certain members of society. For one, it gives the 7,100 vision-impaired (1.9% of Maltese) and over 24,000 illiterate Maltese speakers (6.4% of Maltese) [1], the ability to consume written text from newspaper articles, or any other written media for that matter.

Secondly, speech synthesis is so important nowadays as speech is becoming an increasingly more convenient and effective way for people to interact with their smartphone, smart speaker (such as Google Home), or other smart devices.

1.3 Approach

Since Maltese is a low-resourced language, the majority of components needed to make up a TTS system as well as any datasets will need to be built from scratch. With this low-resource constraint in mind, Concatenative speech synthesisers are an more feasible solution as they require far less data to produce good quality speech. In order to build a Concatenative speech synthesiser, the following components are required:
Text Normalisation - the component which converts any NSWs in the user's text input to a normalised form. G2P - the component which converts the normalised text to sequence of phonemes (a sequence of basic sounds that make up the text when spoken). Speech Unit Database - a database of speech units (very short extracts, usually shorter than a second, from recorded speech) that can be concatenated to synthesise speech. Speech Unit Concatenation Algorithm - the algorithm which smoothly joins together the speech units loaded from the speech unit database as determined

[1] `https://fitamalta.eu/projects/maltese-speech-engine-synthesis-erdf-114/download-maltese-speech-engine/`

by the G2P algorithm.

1.4 Aims and Objectives

The principal aim of this thesis is to develop a speech synthesis system which given any Maltese text, can produce an intelligible audio output. This aim can be subdivided into smaller objectives as follows:

- Building a front-end for the speech synthesis system which normalises text. It is not within the scope of this thesis to cater for every NSW that can appear in the Maltese language, but rather to build a front-end such that further capabilities can be added at a later stage. To demonstrate the capabilities of the front-end, handling of integers written in their numeral form will be implemented.
- Building a component which can convert any body of Maltese text to a sequence of speech units that will make up the intended speech.
- Collecting a database of human voices which can be processed such that they can be used within the chosen speech synthesis system.
- Building a lightweight, efficient and simple back-end for a speech synthesis system, that apart from being able to synthesise text, can also serve as a solid foundation for any future work in the area. The principal component of the back-end will be the Speech Unit Concatenation Algorithm.

2. LITERATURE REVIEW

2.1 Review of Existing Technologies

The TTS solutions known to produce the best results are Deep Generative Neural-Network based ones, such as Google DeepMind's WaveNet [28, 30]. Other highly regarded solutions are Unit-Selection based systems with 'large speech databases' [13]. This traditionally uses the Viterbi algorithm, however more recent versions use a Hidden Markov Model (HMM) approach to choose the best fitting units [31, 25]. While the aforementioned solutions have proven to produce speech at a very good quality, they all require hours of high-quality, labelled, and sometimes segmented, recorded Maltese speech, of which no dataset is available.

This lack of resources necessitates a solution which will work with minimal data. One such solution is the 'Concatenative Speech Synthesiser' with single instances of each speech unit. By using, for example, a diphone as the base speech unit, such systems require less than 20 minutes of recorded audio for full coverage of a language. Despite their low resource requirement, such synthesisers have also previously been referred to as the state-of-the art [10].

2.2 Speech Unit Sizes

In Concatenative Speech Synthesisers, a decision needs to be made on what unit size to use, be it a phone, diphone, half-phone or a syllable.

Using larger unit sizes, such as syllables or polysyllabic units, will produce higher quality speech as there will be less points of concatenation for the same text, meaning less signal processing is carried out. Furthermore, more coarticulation is captured with each unit, resulting in more natural sounding speech [14]. Using larger units comes at a cost of course, that is, the larger a unit is, the more distinct units are needed for complete coverage of a language.

2.3 Handling NSWs

Text Normalisation, or NSW processing, deals with two major problems, detection of NSWs in text, and transformation of NSWs into a sequence of graphemes which can then be handled by the G2P component [4]. This process of Text Normalisation is regarded as one of the greatest challenges when implementing a speech synthesis system for a new language [25].

One method to build a front-end is to go through the input text, applying G2P rules on standard words. When a NSW is met, it is first classified using a classification tree to determine its type, and then it is normalised/expanded according to the 'procedure' defined for that subclass [4]. These procedures are very often rule based and hand written, however this is often not sufficient. A different, more novel, approach for detection and handling of NSWs exists, that is, through use of language models [24].

2.4 Converting Graphemes to Phonemes - G2P

One of the first steps in the Front-End of Concatenative Speech Synthesisers is the conversion of a body of text, formally a sequence of graphemes, into a "phonemic transcription" [6], or a sequence of phones that make up the sound of the text. This process is known as G2P.

G2P programs usually use a large set of G2P, letter to sound, rules to convert graphemes to phonemes. These rules are very context sensitive, i.e. their use depends not only on the grapheme itself, but also on the surrounding graphemes [21]. G2P implementations in commercial TTS systems usually contain a pronunciation dictionary of the most frequently used words, and then fall back to a rule based implementation for Out of Vocabulary (OOV) words [21].

A G2P program for Maltese was created by Crimsonwing for their implementation of a Maltese TTS system [6]. This program makes use of a 'rule based' approach, using over 100 Maltese grapheme to phoneme rules [21, 6]. This method is reported to achieve 98.5% accuracy when compared to human transcription [21]. Using such a rule based approach allows the G2P program to handle unseen, OOV words, something that a dictionary alone is not capable of.

Once the graphemes have been converted to phonemes, the International Phoentic Alphabet (IPA) phoneme symbols must then be converted to Advanced Research Projects Agency alphaBET (ARPABET) symbols. This step is required as IPA symbols are not "computer friendly" [15], and are thus represented in ARPABET by one or two upper case letters [15].

Unlike the English language, where multiple heteronyms (words that are spelt the same, but pronounced differently) exist, Maltese has a "low degree of heterography" [6], meaning very few words that are spelt the same are pronounced differently.

2.5 Diphone Concatenation

One of the most important features of a good concatenation algorithm is the matching of "pitch, phase, amplitude, and frequency envelope" from one speech unit/diphone to the next for natural sounding speech [27].

The most efficient concatenation algorithms are 'overlap-add methods'. Such methods work by selecting certain frames from the two diphones to be concatenated, processing those frames, and "recombining" them "with an OverLap Add (OLA) algorithm", such that the pitches of the two diphones match up and thus, creating continuity. TD-PSOLA is a widely used OLA method that produces good results while being notably computationally efficient [27]. Another concatenation algorithm is the Harmonic Plus Noise Model (HNM), which is a parametric method that produces higher quality recordings than TD-PSOLA, however, TD-PSOLA can be implemented in a much more reasonable time frame [26].

In order for OLA algorithms to work, the sound waves that are to be combined need to be pitchmarked [12]. Pitch marking marks a certain part of a wave, say the crest/peak, at every period. A pitch-period is thus the period of a wave between two consecutive pitchmarks. For speech, the optimal method for pitchmarking is through the use of an Electroglottograph (EGG) [16]. An EGG is a specialised device that measures the contact between the vocal folds (the glottis) [17]. Data from the EGG is synced with the recorded audio so as to extract pitchmarks at the peak of each period [16]. While not ideal, pitchmarks can alternatively be extracted from waves using Pitch Detection Algorithms (PDAs) such as Auto-Correlation [9], 'YIN' [7] and McLeod Pitch Method (MPM) [20].

In TD-PSOLA, after pitch marking, each pitch-period is "windowed" (or 'tapered'), such that the window is "centred on the region of maximum amplitude" and samples close to the centre of the window remain the same, whereas the amplitude of samples towards the edges of the window are scaled down the further they are from the centre [12]. Some popular window functions are Triangle, Hamming, Hanning and Blackman [8]. For speech synthesis, the Hanning Window, is a popular choice [12].

2.6 Evaluation

There doesn't seem to be a 'de facto' way to evaluate TTS systems, and this is understandable as the quality of the result cannot be directly quantified, rather it is usually a qualitative evaluation that can be gathered. Nevertheless there do exist several quantitative evaluations, however most times it simply stems from a score given by the evaluator, which is quite subjective [3].

The evaluation can be focused on one specific feature or aspect of the TTS system. For example one can choose to ask for evaluators' score of the pronunciation of the system, ignoring other aspects such as the naturalness, speed or quality [2]. This method would make sense if the focus of this thesis was to compare how different algorithms affect, say, pronunciation, however this is not the case. The Diagnostic Rhyme Test (DRT) is a test for intelligibility. The DRT plays two pairs of words that are the same, except for a single consonant, to the evaluator, who must choose which word is the correct one. The Categorical Estimation test (CE) evaluates naturalness by asking evaluators for a score for criteria such as the speed, pronunciation, quality and speed. These tests also uncovered that evaluators achieved better scores in DRT in subsequent tries, as they get accustomed to the synthesised voice [22].

The method that evaluates best what this thesis aims to achieve is very similar to the CE. Evaluators are played a number of synthesised texts and they are told to score the audio on criteria such as "intelligibility, naturalness and overall voice quality" on a scale of 1 to 5 [29].

3. CREATING A DIPHONE DATABASE

Since no database of speech units recorded in Maltese could be found, one had to be created from scratch. Alternatively, one recorded in another language, such as FestVox's 'CMU US KAL Diphone'[2] database of US English diphones, could be used. The latter however will sound inferior and "silly" [5] since English and Maltese are distinct, pronunciation-wise. A diphone database for Maltese would need to store at most 1,681 in theory, of which around 1,450 are actually present in Maltese [6].

The first step in creating a speech unit database is to generate 'Carrier Material'. The carrier material was generated in the form of phrases, with a pad word at the beginning and end of each sentence [16]. The phrases were generated such that there would be at least one recorded utterance of a diphone from which it could be extracted. This was achieved by first generating the list of all the diphones. Then, the MLRS corpus[3] was scanned for every unique word, each of which was converted to its phonemic transcription. For each diphone, a word was found from the phonemic transcriptions of the MLRS corpus which had the diphone at its centre. Carrier words were chosen with the diphone at their centre since "it takes time for the human articulation system to start" [16]. Additionally, carrier phrases were made purposefully nonsensical, so that they would be read monotonously and consistently [6].

In order to obtain the best quality when recording the carrier phrases, speech should be recorded in a hemi-anechoic chamber using high quality microphones recording at 96 kHz, which will then be downsampled to 48 kHz "for noise reduction" [25]. Unfortunately, the aforementioned resources were not available, instead, recordings were carried out in a very quiet room, after midnight, to minimise background noise from the street and other people. A smartphone was used as the recording device, as modern smartphones have rather good quality microphones. Using a touchscreen also eliminates any clicking that might be present if a mouse were used instead. The speaker was seated at arms length from the recording device, and was instructed to keep a constant pose, speak loud and clear, enunciate every phoneme in the carrier words, and to give adequate pauses between each word.

Once the carrier phrases have been recorded, the diphones need to be extracted from them. This was accomplished using two distinct methods, each creating a separate diphone database.
The first method was to extract diphones manually, using the 'WebMAUS Basic'[4] web service for audio to phoneme forced alignment, together with EMU-webApp[5] for visualisation of the audio aligned with the phonemes. These two tools together show the start time, duration, and end time of each phoneme in a recorded carrier phrase. While these times are not always perfect, they serve as a decent first indication which can easily be fine-tuned to efficiently and effectively find the diphone boundaries. Using this process, it takes 20 to 25 minutes to extract 10 diphones from a car-

[2] `http://www.festvox.org/dbs/dbs_kal.html`

[3] `http://mlrs.research.um.edu.mt/index.php?page=corpora`

[4] `https://clarin.phonetik.uni-muenchen.de/BASWebServices/interface/WebMAUSBasic`

[5] `https://ips-lmu.github.io/EMU-webApp/`

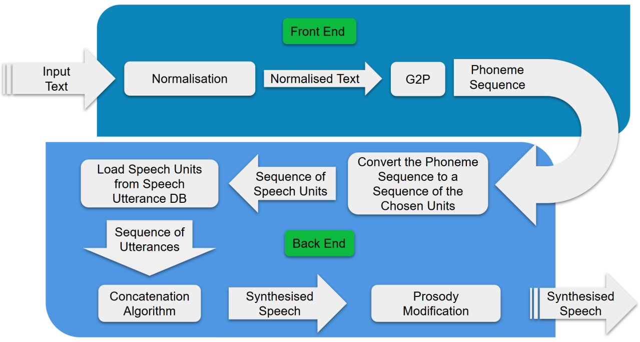

Figure 1: Block Diagram with all the components involved in the entire TTS system

rier phrase. In total, 58 diphones were extracted, enough to synthesise a few phrases to be able to evaluate the TTS system.

The second method extracted diphones automatically, using the Dynamic Time Warping (DTW) algorithm [18]. The DTW algorithm finds the optimal alignment between any two time series, and was used with a diphone from the 'CMU US KAL Diphone' English to extract the same diphone from a Maltese carrier phrase. Since DTW takes O(n^2) time and space, this would take too long to run on 1,450 diphones, so an implementation of FastDTW (a near-optimal approximation) was used, named fastdtw[6], which has O(n) time and space complexity [23].

4. DESIGN AND METHODOLOGY

Given the possible solutions for TTS systems and their requirements, the solution that is most feasible and sensible given what resources are currently available for the Maltese Language would be a Diphone-Based Concatenative Speech Synthesis system.

The block diagram in Figure 1 graphically explains the design of the developed TTS system, the components involved and the flow of data.

4.1 The Front-End

4.1.1 Text Normalisation

The user's text input is first separated by whitespaces, and the individual tokens (words) are sent one by one to the text normalisation component. This component first identifies whether the token is a NSW, and if so, what type. This is currently done by a series of checks, eg if all the characters of a token are digits, then the token is classified as an integer. If a token is not determined to be a NSW, it is sent as is to the next component. If it is determined to be a NSW, the token is passed to that NSW's respective handler, eg if an integer is detected, it is passed to a handler which deals specifically with integers. The normalised text is then sent to the next component.

In the current implementation, the component can only recognise integers, and handle integers between 0 and 9,999. The implemented handler uses a rule-based approach, the rules of which were defined as part of this work. This approach breaks an integer down to its individual digits, and handles each digit one by one, starting from the most significant one (leftmost).

4.1.2 G2P

In order to implement a G2P component, Crimsonwing's G2P program was used. This G2P program, given any text file of Maltese words, will return another text file where each word is replaced by its phoneme sequence, represented in IPA symbols. Crimsonwing's G2P program uses a rule based approach, using letter-to-sound rules [6], and can thus handle OOV words. Once the graphemes have been converted to phonemes, the IPA phoneme symbols are read, and converted to ARPABET through the use of an IPA:ARPABET symbol dictionary. Only the IPA to ARPABET conversions for phonemes present in the Maltese language were implemented, as specified by Farrugia [11].

4.2 The Back-End

4.2.1 Diphone Concatenation

Once the phonemic transcription has been generated by the G2P component, this is converted into a sequence of diphones. The specified diphones are then loaded from the diphone database into memory. Diphone Concatenation, as the name suggests, deals with concatenating the individual diphones to form the target speech, as a single audio wave. The concatenation algorithm of course does not simply stick one diphone after the other, but carrier out some Digital Signal Processing (DSP) at the transitions so that the flow of the audio is smooth, and there are no noticeable jumps in pitch, intonation or volume in the speech.

The first step in the TD-PSOLA algorithm is pitchmarking all the waves that will be concatenated. Two PDAs were implemented that accomplish this task. The first method uses the 'YIN' algorithm implemented in the open source program 'aubio'[7]. The second approach uses the 'Auto Correlation' algorithm implemented in Praat[8], another open source program made specifically for working with speech. Once each wave has been pitch-marked, a 'two period window' is extracted from the wave, which will be used after the OLA part of the algorithm. The wave is then split into its pitch periods ('sub-waves'), i.e. the periods between one pitchmark and the next.

Then, each pitch period is tapered using a 'window function'. The Triangle, Hamming and Hanning windows were all implemented, with the Hanning window set as the default since it is the most popular for TTS applications [12].

Once all the pitch periods have been extracted from a wave and tapered, they can be reconstructed into a single wave by moving each pitch-period making up the wave closer together creating an overlap (to increase the wave's pitch), or further apart from each other by adding a short pause between the pitch periods (to decrease the wave's pitch). By how much the pitch-periods are moved depends on the target pitch. The target pitch is usually the pitch of the preceding wave, so that there are no jumps in pitch [12]. The aforementioned two period windows are used when OLA is carried out, to compensate for any lost samples from overlapping, so as to keep the original duration of the diphone.

[6] https://pypi.org/project/fastdtw/

[7] https://aubio.org/

[8] http://www.fon.hum.uva.nl/praat/

Table 1: The results of the evaluation survey

	DB1	DB2	DB3
Naturalness	1.81	1.13	2.57
Clarity	2.69	1.26	2.72
Intelligibility	2.44	1.04	3.06
Overall Score	13.18	6.12	15.82

5. RESULTS AND DISCUSSION

Since the developed speech synthesiser is made up of many components, these were evaluated individually, then also as part of a complete system.

5.1 Text Normalisation and G2P

A simple Text Normalisation algorithm was created for Maltese, which was tested and evaluated. The Text Normalisation component of the Front-End was first tested by inputting a number of NSWs which it should be able to handle, and others which it shouldn't, to test whether it functions as expected. The G2P component was tested by trying out some Maltese words and phrases and verifying that the expected output is produced.

The Text Normalisation component was tested with multiple inputs, especially those which might break the system. The Text Normalisation component was successfully able to deal with all integers between 0 and 9,999, as intended. It was not, however, able to handle other NSWs such as decimals, negative numbers, measurements, dates and abbreviations, since these were not implemented. NSWs which are not handled do not break the TTS system, and are simply skipped.

The G2P component based on the program by Crimsonwing was also tested and evaluated. Since the program uses a rule-based approach, no input or OOV word can break the program. One notable bug is that inputting any word starting with the letter 'gh̄', such as '*gh̄andek*', alone will incorrectly add a 'K HH' sound to the beginning of the word. This error does not occur when the word is used in a sentence.

5.2 Diphone Databases and Overall Synthesis

Since 3 separate Diphone Databases were built and made available to the speech synthesiser, the performance of each individual database was evaluated. Through this process, the overall performance of the TTS system was also evaluated. This was done by distributing a survey based on the CE as described by Rashad et al. in [22]. The survey demos 3 phrases, each synthesised using each of the three diphone databases. These are then scored for Intelligibility, Clarity and Naturalness on a scale of 1 to 5. Following that, respondents are asked to choose which diphone database they feel produced the best quality speech overall, as this allow respondents to give an overall evaluation without focusing on specific criteria.

The responses from the survey were averaged and tabulated as shown in Table 1. The **FestVox English 'CMU US KAL' Diphone database (DB1)** obtained a rather poor score for naturalness, however a decent score for clarity and intelligibility. The score for clarity can be explained due to this database being professionally recorded in a studio. The score for Intelligibility is also quite respectable, given that this is an English diphone database and not a Maltese one.
The **Diphone database built by using DTW on Maltese carrier phrases (DB2)** performed terribly, as not one respondent was able to understand what was being said. This shows that the implemented DTW algorithm was not effective.
The **Diphone database built by manually extracting diphones from the Maltese carrier phrases (DB3)** scored the highest for each criteria. It was scored the most natural, significantly beating DB1. Surprisingly, it also beat DB1 in clarity, albeit barely, despite not being recorded in a studio with professional equipment.

The fact that diphone database 3 outperformed the English diphone database (DB1) in, not only intelligibility, but every other criteria, alone proves that the defined procedure for creating a diphone database for a new language is, in fact, effective. Furthermore, when asked which diphone database respondents thought sounds the best overall, more than 70% of respondents voted for diphone DB3.

5.3 Diphone Concatenation Algorithm

A diphone concatenation algorithm (TD-PSOLA) was also implemented, and evaluated. This algorithm was evaluated by playing two versions of a couple of phrases, one of which had been synthesised using simple concatenation, and the other using the TD-PSOLA algorithm, and asking respondents which version they preferred.

53.7% of respondents said both audio files sounded exactly the same, 35.2% said they preferred the speech generated by simple concatenation, while the remaining 11.1% said they preferred the speech generated when TD-PSOLA was used. Given that more than half the respondents did not manage to find any differences between the methods, it is rather hard to draw conclusions about the effectiveness of the implemented diphone concatenation algorithm, however, more respondents seem to agree that simple concatenation outperforms the implementation of TD-PSOLA.

6. CONCLUSION AND FUTURE WORK

A Diphone-Based Concatenative Speech Synthesiser for Maltese was successfully developed in this dissertation, receiving an average rating of 3.06 (out of 5) for intelligibility.

The developed implementation can receive Maltese text from the user and deal with some NSWs, such as integers between 0 and 9,999. The G2P component, using Crimsonwing's G2P program, converts any body of text into a phonemic transcription, without needing a pronunciation dictionary.

Three diphone databases were made available to the speech synthesiser, with the one made completely as part of this work (DB3) performing the best in all criteria. To facilitate the creation of new Diphone Databases for Maltese, or the completion of DB3, a program was written which can automatically generate Carrier Phrases for any given list of diphones. A process was then defined for recording these carrier phrases at the best quality possible, with no specialised equipment. A process was also defined which mentions tools that greatly optimise the process and increase the precision of manually extracting the diphones.

The TD-PSOLA algorithm was implemented as the Diphone Concatenation algorithm. This was however scored by evaluators as adding more noise to the synthesised speech than a simple 'appending' concatenation algorithm. Due to the modular design of the speech synthesiser, the simple concatenation algorithm can be set as the default until a more refined algorithm is implemented.

Not only is this work a significant step forward in building a state of the art speech synthesiser for Maltese, but it can also be used by other low resourced languages. The 'Text Normalisation' and 'G2P' components would of course need to be adapted, however the rest of the developed programs and processes can be applied to any other language with minimal additional effort.

6.1 Future Work

While the work done satisfies many of the initial aims and objectives of this thesis, there still remains a considerable amount of work to be done for the speech synthesiser to be considered state of the art, or even usable in extent.

In future work, the functionality of the Text Normalisation component can be expanded upon to handle more types of NSWs. More diphones can be recorded and extracted so that a greater coverage of the language is achieved. The Diphone Concatenation algorithm can also be revisited since it wasn't found to perform very well. Finally, a prosody modification component can be added which modifies the intonation and expression of speech based on the what is being said and the punctuation used.

Apart from the suggested improvements to the core functionality of the synthesiser, there are some accessibility improvements which can be implemented that would, by far, improve the usability of this synthesiser. Keeping in mind that a large portion of the users of speech synthesisers are vision impaired individuals, making the synthesiser highly accessible and effortless to use is paramount. Building the synthesiser as a web browser plug-in or a smartphone application would improve its usability. For example, a user reading an article written in Maltese on a website can simply press a button within their browser, or perhaps issue a voice command, for the text to be read to them.

6.2 Discussion

All in all, a basic, yet modular and expandable, Text Normalisation component as well as a very robust and versatile G2P component were built. Three separate Diphone Databases were made available to the speech synthesiser, and the TD-PSOLA concatenation algorithm was implemented. The diphone database of manually extracted diphones from Maltese carrier phrases concatenated by the TD-PSOLA algorithm, both created as part of this dissertation, outperformed the speech synthesised when using the professionally recorded English diphone set.

7. REFERENCES

[1] Census of population and housing. Technical report, National Statistics Office, 2011.

[2] A. Al-Wabil, H. Al-Khalifa, and W. Al-Saleh. Arabic text-to-speech synthesis: A preliminary evaluation. pages 4423–4430, 2007.

[3] N. K. Bakhsh, S. Alshomrani, and I. Khan. A comparative study of arabic text-to-speech synthesis systems. 6(4):27–31, 08 2014.

[4] S. Beliga and S. Martincic-Ipsic. Text normalization for croatian speech synthesis. pages 1664–1669. IEEE Publishing, 2011.

[5] A. W. Black and K. A. Lenzo. Multilingual text-to-speech synthesis. volume 3, pages iii–761, May 2004.

[6] M. Borg, K. Bugeja, G. Mangion, and C. Gafa. Preparation of a free-running text corpus for maltese concatenative speech synthesis. Apr 2011.

[7] A. d. Cheveigne and H. Kawahara. Yin, a fundamental frequency estimator for speech and music. 111(4):1917–30, 2002.

[8] A. W. Doerry. Catalog of window taper functions for sidelobe control. 2017.

[9] J. Dubnowski, R. Schafer, and L. Rabiner. Real-time digital hardware pitch detector. 24(1):2–8, February 1976.

[10] T. Dutoit. *An Introduction to Text-to-speech Synthesis*. Kluwer Academic Publishers, Norwell, MA, USA, 1997.

[11] P.-J. Farrugia. Text to speech technologies for mobile telephony services. 2005.

[12] W. Holmes. *Speech synthesis and recognition*. CRC press, 2001.

[13] A. J. Hunt and A. W. Black. Unit selection in a concatenative speech synthesis system using a large speech database. volume 1, pages 373–376, May 1996.

[14] S. P. Kishore and A. W. Black. Unit size in unit selection speech synthesis. pages 1317–1320, 2003.

[15] A. Klautau. Arpabet and the timit alphabet, 2001.

[16] K. A. Lenzo and A. W. Black. Diphone collection and synthesis. 2000.

[17] E. P.-M. Ma and A. L. Love. Electroglottographic evaluation of age and gender effects during sustained phonation and connected speech. 24(2):146 – 152, 2010.

[18] F. Malfrere and T. Dutoit. High-quality speech synthesis for phonetic speech segmentation. 1997.

[19] F. Malta. Erdf 114 maltese text to speech synthesis.

[20] P. McLeod and G. Wyvill. A smarter way to find pitch. 2005.

[21] P. Micallef. *A text to speech synthesis system for Maltese*. PhD thesis, University of Surrey (United Kingdom), 1997.

[22] M. Rashad, H. M. El-Bakry, and I. R. Isma'il. Diphone speech synthesis system for arabic using mary tts.

[23] S. Salvador and P. Chan. Fastdtw: Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580, 2007.

[24] R. Sproat, A. W. Black, S. Chen, S. Kumar, M. Ostendorf, and C. Richards. Normalization of non-standard words. *Comput. Speech Lang.*, 15(3):287–333, July 2001.

[25] A. Stan, J. Yamagishi, S. King, and M. Aylett. The romanian speech synthesis (rss) corpus: Building a high quality hmm-based speech synthesis system using a high sampling rate. 53(3):442 – 450, 2011.

[26] Y. Stylianou. Applying the harmonic plus noise model in concatenative speech synthesis. *IEEE Transactions on Speech and Audio Processing*, 9(1):21–29, Jan 2001.

[27] S. Toma, G. Tarsa, E. Oancea, D. Munteanu, F. Totir, and L. Anton. A td-psola based method for speech synthesis and compression. pages 123–126, June 2010.

[28] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. In *SSW*, page 125, 2016.

[29] M. Zeki, O. O. Khalifa, and A. W. Naji. Development of an arabic text-to-speech system. pages 1–5, May 2010.

[30] H. Zen. Generative model-based text-to-speech synthesis. pages 327–328. IEEE, 2018.

[31] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. W Black, and K. Tokuda. The hmm-based speech synthesis system (hts) version 2.0. 09 2007.