A Diphone-Based Maltese Speech Synthesis System

Progress 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

This final year project aims to build a speech synthesis system for the Maltese Language. Various speech synthesis systems will be researched, and a system for Maltese will be built based on what resources are available for the Maltese language, which are somewhat minimal. The outcome should serve as a solid foundation for future refinement of a synthesiser for Maltese.

1. INTRODUCTION

The aim of this thesis is to, for any given string of text in the Maltese Language, output a clear and legible synthesised voice. Different well-established speech synthesis methods will be explored, and one will be selected based on the resources needed (linguistic datasets), computational resources needed, 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 Motivation for the Project

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 7.1 thousand of vision-impaired (1.9% of Maltese), or the more than 24 thousand illiterate Maltese speakers (6.4% of Maltese),[1] the ability to consume written text from news-paper 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, or other smart devices. At the time of writing, there is no available Speech Synthesiser System for Maltese, although work on the topic has been carried out by Dr Paul Micallef and FITA/Crimsonwing[7]. This thesis aims to fill that gap in providing a solid open-source solution which can at least act as a foundation for future work in this field.

1.2 Why the Problem is Non-Trivial

- There are no freely available resources for Maltese TTS systems, thus the system must be designed with these very minimal resources in mind.
- The human speech that will be recorded to create this synthesiser must be carefully chosen so as to obtain as much coverage of the language in as little time as possible.
- The recorded speech will need to be segmented into units, and given that no previous speech database exists, automatic segmentation methods will suffer to produce high quality segments of units.
- The concatenation method needs to be highly efficient so that it can run even on low computational power systems, possibly on a robot with a low-power chip.
- It is quite difficult to obtain high quality speech recordings without specialised recording equipment or studios. It needs to be made sure that any recordings made are of the highest quality possible with the given resources. This may also include some post processing to clean and normalise the recorded audio.
- How to use the outcomes of evaluation to determine if any particular units need re-recording or re-alignment/re-segmentation.

2. AIMS AND OBJECTIVES

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

- Researching different speech synthesis systems, and then choosing which applies best to the Maltese language given the resources available.
- Gather a Dictionary of *word* → *phoneme sequence* pairs, by building a G2P program or otherwise.
- Collecting a database of human voices which can be processed such that they can be used within the chosen speech synthesis system.
- Build 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.
- Build 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 NSWs such as numbers and abbreviations will be implemented. Handling dates and OOV words may be tackled, depending on the time frame.

- Time permitting, the synthesis system may also be ported to a robot.
- A web plugin would be very handy for the vision impaired, however is probably not something that will be accomplished within the time frame of this thesis.
- Experimenting with prosody modification in order to make the synthesised speech sound more natural and human-like.

3. BACKGROUND RESEARCH

Units in a unit-selection speech system or a concatenative speech system are the standard utterances which will be stored in the 'speech database' and be chained together when generating speech. These units are usually phonemes, diphones, half-phones, allophones or syllables.

The centre of a singular phone is the most stable part of that phone, whereas the 'edges', or the areas where a transition from one phone to the next occurs, is the area that machines find hardest to produce. It is because of this that diphones are so often used as a unit, as they start from the most stable part of the first phone, cover the transitionary area, and end on the most stable part of the next phone.[17] This is demonstrated in Figure 1, where the annotations at the bottom are the phones (# being silence) and the annotations on top are the Diphones.

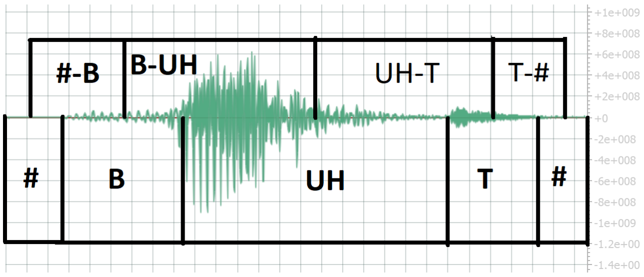

Figure 1: Visual representation of Phones and Diphones

4. LITERATURE REVIEW

4.1 Review of Existing Technologies

Unit Selection based Speech Synthesisers are known to obtain very respectable results, and were the best in their time. Two major types of Unit Selection systems exist, those with 'Large Speech Databases', containing multiple utterances for each unit, and those with a single instance of each unit.

Unit Selection Concatenative Speech Synthesisers using a Large Speech Database will have a very large database of speech recordings (utterances), which gives the system a plethora of utterances to choose from, rather than just one utterance of a particular unit, say, a phoneme. Since such a large database of speech exists, it is much more often that a desired utterance with the correct and intended prosodic characteristics is found, without the need for waveform modification. Since the units chosen by the Viterbi algorithm will likely be as desired, they will require little to no signal processing, this results in an output with correct intonation and thus a more natural sounding result.[12]
The units are traditionally chosen by the Viterbi algorithm, however improvements of the system exist where the units are selected by a Hidden Markov Model (HMM).[31][24]

It must be noted that no *large speech recording database* for Maltese exists as of yet, which makes this type of system unfeasible for the Maltese language.
There are techniques for building such large speech databases automatically from audio books, however no texts with accompanying audio recording exist in the public domain for Maltese.[19]

The current state-of-the-art lies in the field of Deep Neural Network-based, generative speech synthesis systems. These also are not feasible to use given the hours of data needed for training.[27][30]

Concatenative Speech Synthesisers with single utterances of each unit have also, in their time, more than once been referred to as the state-of-the-art in high quality synthesis.[10] These are much more feasible for Maltese given how little data is needed in comparison to other solutions.

4.2 Choosing a Unit Size

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.
Syllable units have been shown to produce the highest quality results, especially for languages that are 'syllable centred'. Since syllables are longer than diphones or phones, more coarticulation is captured. Furthermore, there will be less points of concatenation for the same text, which will result in higher quality, more natural sounding text.[15][5] This idea can be further extended to polysyllabic units, which display the same advantages over shorter units.[28]
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. Solutions that combat this problem do exist, one of which is Approximate Matching of Units, however such a solution is meant to fill in gaps where a rarely used unit might be missing, and not for the majority.[20] Other solutions involve either finding the closest matching diphone (backing-off), or extending the boundaries of the adjacent two diphones to fill in the missing diphone.[8][14]

While 'Global Speech Databases' do exist, such as GlobalPhone[22], and units from one language can be used for similar sounding and related languages, no language is notably similar to Maltese, and if a units from a language which is not very similar are used, the results will sound off and "silly".[6]

4.3 Generating Free Text and Recording Diphones

With this information, it is reasonable to choose to build a Diphone Based Speech Synthesiser. A syllable is too long of a unit, as too many utterances need to be recorded for decent coverage of a language. On the other hand, there are 41 phones in the Maltese language, meaning no more than 1681 (41^2) diphones need to be recorded for complete coverage. Out of those, only about 1349 (around 80%) diphones were found in a Maltese text corpus.[7]

When choosing a word from which a specific diphone will be extracted, it is important to, whenever possible, have that diphone in the middle of the word, since "it takes time for the human articulation system to start". There seems to be no clear cut answer on on how the text which the speakers will read out should be generated.[17]
One option is to have 'free text', which consists of natural sentences, one advantage of this is that prosody is captured in the extracted diphones, so the result should sound more natural.[7]
Another option is to use 'rainbow text', which is manually prepared by an expert to have at least one instance of each diphone. Sentences are usually non-sensical for monotony and consistency.[7]
Yet another option is to use computer generated nonsense words. This results with having words with the optimal diphone placement for best quality extraction, however this requires expert phoneticians to read, who themselves may find it hard to utter some of the words.[17]

It is being proposed that, as a middle ground between all of these solutions, one word will be chosen for each diphone, ideally with the diphone in the middle of the word. This list of words will then be split randomly into 10-word-long sentences, with a pad word at the beginning and end of each sentence.
With such a procedure, 1,349 words need to be recorded, 1619 counting the 'pad words'. Even at a very slow speaking rate of 100 words per minute, recording these 1,349 diphones would take no longer than 17 minutes.

Since the speaker's voice may vary over one recording session, or if multiple recording sessions take place, it is important to normalise the recordings in some way. The mean RMS power can be calculated over all the words, so that a "modification factor" is produced which can be applied to the words needing normalisation.[17]

4.4 Diphone Extraction and Segmentation

The task of manually selecting the parts of each word which correspond to a particular diphone would take days, which gives rise to the need for an automatic labelling method.
One of the earlier methods of automatic labelling was HMM based, however this was a supervised method, and needed a large speech database of pre-labelled units. A more modern spin on this method pairs HMMs with SVMs, however still needs a manually segmented and hand-labelled speech database for training.[11]
An alternative approach came about shortly after which did not come with the requirement of a pre-labelled database. This approach compares the recorded audio for a sentence in free text, with the audio output of another, already existing concatenative speech synthesiser. This comparison is done using the Dynamic Time Warping (DTW) algorithm. The DTW algorithm works very similarly to the Levenshtein String Distance algorithm. It is used to measure the distance between any two time series, and computes the optimal global alignment of the two series.[18]
Multi-Layer Perceptron (MLP) based labelling methods also exist, however these are usually used as refinement on the borders of the segments, rather than as an initial pass.[16]
Other methods include using a "syllable rate" timing (in the case of syllable based systems) to segment the recording, the output of which is compared to a rule-based or manually segmented signal, and the segmentation is refined using a HMM. This can be further paired with "Vowel Onset Point detection", however this is primarily a syllable specific concept.[13]

4.5 Concatenation of Diphones

One of the most important features of a good concatenation algorithm is the matching of "pitch, phase, amplitude, and frequency envelope" from one diphone to the next for natural sounding speech.[26]

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". Time domain pitch synchronous overlap-add (TD-PSOLA) is a widely used OLA method that produces good results while being notably computationally efficient.[26]

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.[25]

4.6 Prosody Modification

Prosody Modification can be carried out on the output waveform in order to add intonation to synthesised speech, make speech sound more natural and even possibly add hints of emotion. Prosody is made up of microprosody and macroprosody. Microprosody refers specifically to the prosody of a basic "individual speech sound", whereas macroprosody refers to the prosody or intonation present over a larger body of text, as applied by the speaker.[9]
Dynamic Time Warping (DTW) is one technique which can be applied to the output speech waveform to modify the prosody of the synthesised text. In order to make the algorithm faster and require less memory, instead of using DTW on the amplitudes of the speech signals, it can be used on the Mel-frequency cepstral coefficients (MFFCs) of the of the "speech frames". This method has been shown to be very effective with simpler intonation tasks, such as making a sentence declarative or interrogative. It was discovered that intensity, pitch and phoneme duration are the major features which influence prosody, whereas intensity has a relatively minor influence on the macroprosody.[9]

4.7 Front-End and Handling Non-Standard Words

Non-Standard Word (NSW) processing deals with two major problems, detection NSWs in text, and transformation of NSWs into a sequence of phonemes pronounceable by the speech synthesis system. It also deals with other problems such as whether to pronounce the number 3 as an cardinal or an ordinal number, i.e. whether to say '*tlieta*' or '*tielet*'.[4] This problem is an even more present in Maltese where the number 5 can be read as '*ħamsa*' or '*ħamest*' when used as a cardinal number.
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 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.[23]

4.8 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 might make sense if the focus of this paper was to compare how different algorithms affect 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 (CE) evaluates naturalness. The CE test asks evaluators for a score of factors 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.[21]

The method that evaluates best what this thesis aims to achieve is very similar to 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]

5. PROPOSED SOLUTION

Given these possible solutions, the solution that is most feasible given what resources are currently available for the Maltese Language would be a Diphone Based Speech Synthesis system.

The intended speech synthesis system will have two principal components as follows:

The front-end will receive the text to be synthesised and normalise it. The front-end will scan the input for Non-Standard Words (NSWs), i.e. text which the back-end is not natively capable of converting into phonemes, and replace them with text which can then be synthesised into speech by the back-end (e.g. the number "1984" will be converted to "elf disa' mija erbgħa u tmenin"). The main examples of NSWs are numbers, dates, abbreviations and out of vocabulary words (OOVs). The output of this component is still text.

The back-end will receive the normalised text from the front-end and convert it into speech.

The back-end will be built first, followed by the front-end, as the core of the synthesis system is the back-end, and the front-end is simply a mechanism which allows for more inputs to be accepted and synthesised correctly.

5.1 Pronunciation Dictionary

A method to convert any word into a sequence of phonemes is needed. This can be achieved in one of four ways:

1. There exists a CSV file named *Phonems.csv*, which contains 150,000 Maltese words, along with their phonetics, i.e. the sequence of phonemes that are required to pronounce the word. This would be a simple and effective way of building a pronunciation dictionary for the synthesiser. However, The Maltese language makes use of phonemes such as 't', 's', and 'ts', as well as 'd', 'z' and 'dz'. Since the phoneme sequences in the CSV file have no form of delimitation between one symbol and the next, it becomes impossible to distinguish between a 't' phoneme followed by an 's' phoneme, and a singular 'ts' phoneme. Thus, use of this method would mean that some heuristic would have to be used to determine whether, say, a 'ts' is a 't''s' or a 'ts'. For example, if the word contains a 'z', then a 'ts' is likely to be a 'ts' and not a 't''s'.

2. The second option would be to use the MalteseG2P program, developed by Crimsonwing. The Maltese Grapheme-to-Phoneme program, given any text file of Maltese words, will return another text file where each word is replaced by its phoneme sequence by applying the 107 Maltese grapheme to phoneme rules. This method would also be very effective, however faces the same problem as the previous method, i.e. there is no delimiter between phonemes.

3. The third method would be to extract *word - phoneme sequence* pairs from the Maltese 'T-corpus from Speech Synthesis Project'. This corpus contains hundreds of text files from 5 different domains/areas. For each text file there are three versions, the first being the corpus itself, the second being a normalised text (where numbers and abbreviations are converted into a representative symbol) and the last being the phoneme sequences for each word. What is different in this corpus from the previous, is that words are separated by symbols, and phonemes within that word separated by spaces. Having this space between the phoneme symbols solves the aforementioned problems of having ambiguous groupings of phoneme symbols. The downside with using this method is that it will take a considerable amount of time to correctly pair the words from one version of a file with their phonemes from another file.

4. The last solution is to implement the G2P algorithm, using the 107 rules, essentially recreating the Crimsonwing G2P program. The problem with this method is that it will take rather long to encode 107 linguistic rules into a program. Despite the time cost, this might be a worthwhile method as the output can be generated in whatever format best suits the needs of the system. Furthermore, this method allows for handling of words that might not be in the pronunciation dictionary.

Method one will be used for this scenario, as it will provide very reasonable outputs, and will not take too much time to implement. This will be implemented with modularity in mind, such that it can be easily replaced with a possibly better method in the future.

Once a dictionary of *word → phoneme sequence* pairs has been acquired, the phonemes will be converted from IPA symbols to ARPABET symbols. This process is standard practice in speech synthesis and is necessary to convert the IPA symbols to ASCII characters. This step has already been done, only that some IPA symbols, such as 'ɐ', do not have an ARPABET symbol, and thus one needs to be defined.

5.2 Diphone Database

Next, a Diphone Database needs to be created. This will store a recording of every possible Diphone, and will consist of at most 1,681 diphones, of which around 1,450 are likely to be generated and stored. To capture the required diphones, a word which contains the required diphone in the middle will be chosen, assuming one exists. If not, any word which contains the required diphone will be chosen. Once the words have been chosen, they will be randomised and split into sentences of random length (7-10 words). Another random word will be added to the begging and end of each sentence, as a 'padding' to the start and end of speech, from which no diphones will be extracted. This method ensures that there will always be at least one recording of each diphone, and the randomness ensures that speech will be 'monotonous' and consistent and thus eliminate unwanted intonation and expression in speech.
A more refined method would be to have multiple recordings of each diphone, according to its statistical usage in the corpus, such that more frequently used diphones will have more recordings and thus there will be more 'selection' for an optimal recording.[7]

5.3 Diphone Extraction

Once this is done, the recordings of words need to be segmented to isolate the diphone they were meant capture. To accomplish this task, the recordings might either be segmented manually, by hand, or automatically with the use of DTW. DTW will require a speech synthesis database for another language, since one for Maltese is not available, however this has shown to still be very effective for a first-pass, however manual finetuning will likely be required.

5.4 Concatenation of Diphones

The final step for this component is to develop a program which will take the input text, convert it to its sequence of phonemes, select the diphones which fit every pair of phonemes and finally concatenate the diphones.
Several methods exist for concatenation, however TD-PSOLA will be implemented as it is known to reliably produce satisfactory results while being relatively simpler to implement than other parametric concatenation algorithms.

5.5 The Front-End

At this point the back-end will be virtually completed, and thus some work will be put into the front-end. Normalisation of abbreviations can be done by building a dictionary of abbreviations for the Maltese language. This will probably have to be done manually, unless an already existing dictionary is found. An abbreviation dictionary is not a perfect solution, as some abbreviations have multiple meanings, solving this is not within the scope of this thesis, however, could be done by detecting the context and applying the abbreviation which is used most frequently within that context.
For normalisation of numbers, an algorithm which can convert a number written in Arabic numerals such as '1984' to a number as it would be read in Maltese "elf disa' mija erbgħa u tmenin" will have to be developed. The back-end should then be able to synthesise this normalised number.

Other NSWs exist, out of vocabulary words (OOVs)/proper nouns and dates for example, however they will not be tackled in this thesis, as the focus is on the back-end of the synthesiser, and the front-end is mainly being implemented to set the foundation for future work, and have a front-end which is built in a modular fashion, such that support for more NSWs can be easily added.

6. EVALUATION PLAN

Since the front-end and back-end are somewhat separate components, they will be evaluated separately.

The front-end can be evaluated by trying out 5-10 examples of NSWs that the system should be able to handle (and those which it isn't able to) and determining whether the text output is as expected.

The back-end needs to be evaluated by native Maltese speakers. 3 or 4 words, phrases, sentences, and paragraphs will be chosen, inputted to the system, and their outputs stored.
The outputs will then be compiled into a form, with the sentence they are trying to read. (May also not include the sentence to see whether the user can tell what is being spoken, especially for ambiguous words). For each recording, the user will be asked to rate the generated speech on various criteria. This may be a 1 to 5 score for the intelligibility, how natural the voice sounds, and whether it is too slow or too fast.

An interesting idea would be to allow for accommodation to a specific user's comprehension ability. For example, a child or an elderly might prefer slower reading speeds whereas an adult may prefer faster speeds. An outcome of this evaluation should be whether this would be in fact a needed feature.

7. CONCLUSIONS, EXPECTED OUTCOMES AND DIFFICULTIES/CHALLENGES

Since Diphone Based Speech Synthesis is being applied, the outcome cannot be expected to produce a human-like voice, with intonation or emotion, but is expected to produce an intelligible audio output. The front-end is expected to normalise just two special cases, numbers and abbreviations, which means there will still be inputs which the system will not be able to synthesise correctly.

8. 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. In

EdMedia: World Conference on Educational Media and Technology, pages 4423–4430. Association for the Advancement of Computing in Education (AACE), 2007.

[3] N. K. Bakhsh, S. Alshomrani, and I. Khan. A comparative study of arabic text-to-speech synthesis systems. *International Journal of Information Engineering and Electronic Business*, 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. Bellur, K. B. Narayan, K. R. Krishnan, and H. A. Murthy. Prosody modeling for syllable-based concatenative speech synthesis of hindi and tamil. In *2011 National Conference on Communications (NCC)*, pages 1–5, Jan 2011.

[6] A. W. Black and K. A. Lenzo. Multilingual text-to-speech synthesis. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages iii–761, May 2004.

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

[8] R. A. Clark, K. Richmond, and S. King. Multisyn: Open-domain unit selection for the festival speech synthesis system. *Speech Communication*, 49(4):317 – 330, 2007.

[9] M. O. Co and R. C. L. Guevara. Prosody modification in filipino speech synthesis using dynamic time warping. In *TENCON 2003. Conference on Convergent Technologies for Asia-Pacific Region*, volume 1, pages 397–401 Vol.1, Oct 2003.

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

[11] H. Frihia and H. Bahi. Hmm/svm segmentation and labelling of arabic speech for speech recognition applications. *International Journal of Speech Technology*, 20(3):563–573, Sep 2017.

[12] A. J. Hunt and A. W. Black. Unit selection in a concatenative speech synthesis system using a large speech database. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, volume 1, pages 373–376 vol. 1, May 1996.

[13] R. Janakiraman, J. C. Kumar, and H. A. Murthy. Robust syllable segmentation and its application to syllable-centric continuous speech recognition. In *2010 National Conference On Communications (NCC)*, pages 1–5, Jan 2010.

[14] P. Kasparaitis and K. KanÄ■ys. Phoneme vs. diphone in unit selection tts of lithuanian. *Baltic Journal of Modern Computing*, 6(2):162–172, 2018.

[15] S. P. Kishore and A. W. Black. Unit size in unit selection speech synthesis. In *IN PROC. EUROSPEECH 2003*, pages 1317–1320, 2003.

[16] K.-S. Lee. Mlp-based phone boundary refining for a tts database. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(3):981–989, May 2006.

[17] K. A. Lenzo and A. W. Black. Diphone collection and synthesis. In *Sixth International Conference on Spoken Language Processing*, 2000.

[18] F. Malfrere and T. Dutoit. High-quality speech synthesis for phonetic speech segmentation. In *Fifth European Conference on Speech Communication and Technology*, 1997.

[19] K. Prahallad and A. W. Black. Segmentation of monologues in audio books for building synthetic voices. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(5):1444–1449, July 2011.

[20] E. V. Raghavendra, B. Yegnanarayana, and K. Prahallad. Speech synthesis using approximate matching of syllables. In *2008 IEEE Spoken Language Technology Workshop*, pages 37–40, Dec 2008.

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

[22] T. Schultz. Globalphone: a multilingual speech and text database developed at karlsruhe university. In *Seventh International Conference on Spoken Language Processing*, 2002.

[23] 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.

[24] 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. *Speech Communication*, 53(3):442 – 450, 2011.

[25] 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.

[26] . Toma, G. TÃćrÅ§a, E. Oancea, D. Munteanu, F. Totir, and L. Anton. A td-psola based method for speech synthesis and compression. In *2010 8th International Conference on Communications*, pages 123–126, June 2010.

[27] 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.

[28] M. V. Vinodh, A. Bellur, K. B. Narayan, D. M. Thakare, A. Susan, N. M. Suthakar, and H. A. Murthy. Using polysyllabic units for text to speech synthesis in indian languages. In *2010 National Conference On Communications (NCC)*, pages 1–5, Jan 2010.

[29] M. Zeki, O. O. Khalifa, and A. W. Naji. Development of an arabic text-to-speech system. In *International Conference on Computer and Communication Engineering (ICCCE'10)*, pages 1–5, May 2010.

[30] H. Zen. Generative model-based text-to-speech synthesis. In *2018 IEEE 7th Global Conference on Consumer Electronics (GCCE)*, 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. *Proc. of ISCA SSW6*, 09 2007.

9. WORK PLAN

Figure 2 is a Gantt chart representing the work plan:

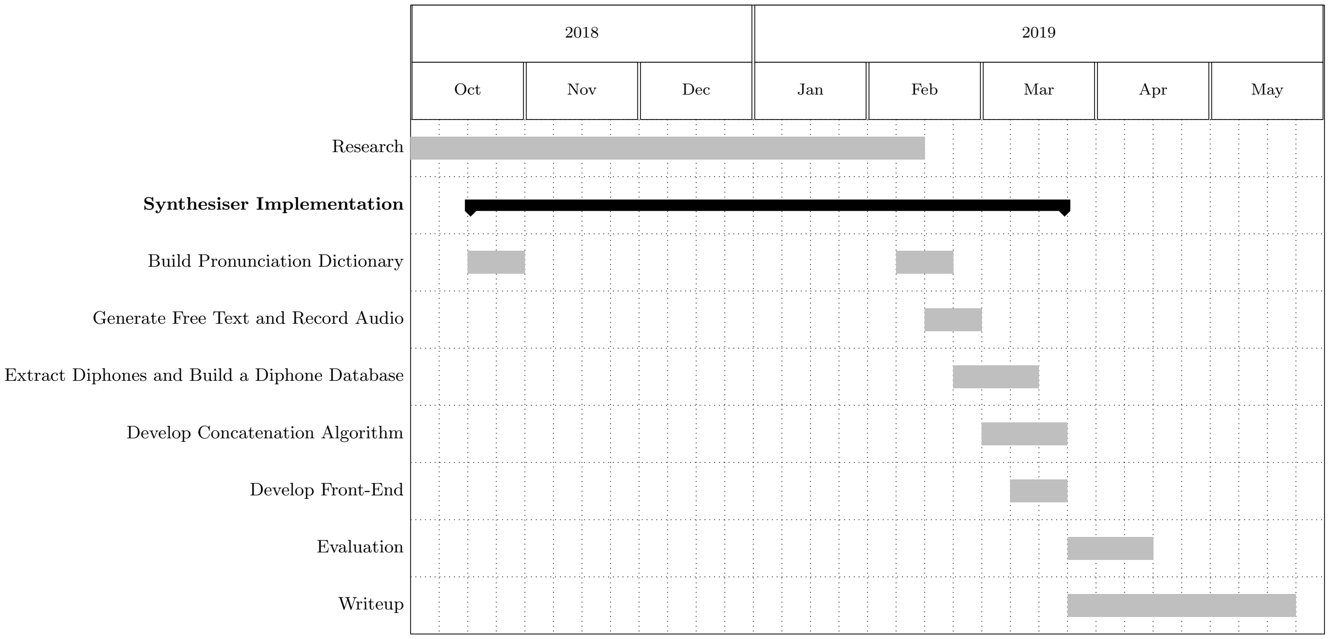

Figure 2: Gantt Chart