UNIVERSITY OF SOUTHAMPTON

FACULTY OF PHYSICAL AND APPLIED SCIENCES

Electronics and Computer Science

Modern Standard Arabic Speech Corpus

by

Nawar Halabi

Supervisor: **Prof. Mike Wald**
Examiner: **Dr Gary B Wills**

Thesis for the degree of Master of Philosophy

August 2015

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

Corpus design for speech synthesis is a well-researched topic in languages such as English compared to Modern Standard Arabic, and there is a tendency to focus on methods to automatically generate the orthographic transcript to be recorded (usually greedy methods), which was used in this work. In this work, a study of Modern Standard Arabic (MSA) phonetics and phonology is conducted in order to develop criteria for a greedy method to create a MSA speech corpus transcript for recording. The size of the dataset is reduced a number of times using optimisation methods with different parameters to yield a much smaller dataset with the identical phonetic coverage offered before the reduction. The resulting output transcript is then chosen for recording. A phoneme set and a phonotactic rule-set are created for automatically generating a phonetic transcript of normalised MSA text which is used to annotate and segment the speech corpus after recording, achieving 82.5% boundary precision with some manual alignments (~15% of the corpus) to increase the precision of the automatic alignment. This is part of a larger work to create a completely annotated and segmented speech corpus for MSA speech synthesis with an evaluation of the quality of this speech corpus and, where possible, the quality of each stage in the process.

FACULTY OF PHYSICAL AND APPLIED SCIENCES

Computer Science

Thesis for the degree of Master of Philosophy

MODERN STANDARD ARABIC SPEECH CORPUS

Nawar Halabi

Table of Contents

List of Tables

List of Figures

DECLARATION OF AUTHORSHIP

I, Nawar Halabi, declare that this thesis and the work presented in it are my own and has been generated by me as the result of my own original research.

Modern Standard Arabic Speech Corpus

I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;

2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;

3. Where I have consulted the published work of others, this is always clearly attributed;

4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;

5. I have acknowledged all main sources of help;

6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;

7. [Delete as appropriate] None of this work has been published before submission [or] Parts of this work have been published as: [please list references below]:

Signed: Nawar Halabi...

Date: 06-08-2015...

Acknowledgements

I would like to thank Maha and Ouadie for their hard work helping as linguistic experts for correcting, annotating and aligning the corpus.

Definitions and Abbreviations

Back-end: Part of a complete TTS system which converts a sequence of phonemes with linguistic features to a speech signal.

Bootstrapping (HMM models): Training an HMM model/s using manually segmented and aligned speech corpus for potentially using to segment another speech corpus by forced alignment.

Buckwalter Transliteration (Buckwalter 2002): Is a one-to-one mapping between Arabic characters and Latin letters and symbols. Mainly used in this work because HTK cannot handle Arabic script as input.

Diacritics and Diacritisation: Diacritics are symbols added to letters. In Arabic, they correspond to short-vowel phonemes, gemination or absence of short-vowel phonemes (sukoon). Diacritisation is the process of adding those diacritics to Arabic script.

DNN: Deep Neural Network. In simple terms, Neural Networks which have more complicated and layered structure which requires different methods of training.

Emphasis: Here it is the velarisation or pharyngealisation of consonants in Arabic (Laufer & Baer 1988). They are secondary articulations which correspond to changes in the pharynx or epiglottis from the primary articulation. These movements are called 'emphasis' in this work for convenience.

Front-end: Part of a complete TTS system which converts raw text to a phoneme sequence with linguistic features which is used as the input to a speech synthesiser (Back-end).

Gemination: In Arabic, it is usually described as the doubling of a consonant. Usually the effect is dependent on the consonant's articulation category. It is shown in this work that gemination in Arabic is more accurately described as the lengthening of part of the consonant. Linguistically, a geminated consonant is treated as two consecutive consonants when syllabifying a word.

HMM: Hidden Markov Model. A sequential probabilistic model used to model speech for speech recognition and synthesis.

Mel Frequency Cepstral Coefficients (MFCC): A parametric representation of the speech signal's power spectrum in a short interval (Jurafsky & Martin 2008).

MSA: Modern Standard Arabic. Is a standardised variety of Arabic which is used nowadays in official documents, news etc.

Normalisation and Normalised Script: In Speech Synthesis, this refers to the input text after all irregular content in it has been converted into a form that can be phonetised by a machine. For example, abbreviations for example HMM could be converted to “Hidden Markov Models” or “Aitch Em Em” by the normalisation process, making it easier to generate the phoneme sequence to be synthesised. The normalisation process also includes numbers, punctuation (such as brackets) and – in some cases – spell-checking (Taylor 2009).

Phoneme: Not to be confused with phone, is the smallest unit of phonology in a language which – when changed – could change the meaning. Phonemes can be seen as classes of phones meaning that a phone is a realisation of a phoneme in a certain context (Taylor 2009).

Phonotactics: The rules that govern the types of phonemes, syllables, consonant clusters etc. that are allowed to occur in speech (Habash 2010; Biadsy & Hirschberg 2009).

Phonetic Unit: Phone, Diphone, Triphone, Syllable… is a phonetic or phonological segment which in corpus design is used to define the phonemic content required to be covered by the transcript.

Phonetisation: The conversion of normalised script to a phoneme sequence.

Pronunciation Dictionary: A list of pronunciations (phoneme sequences) used mainly in speech recognition and phonetisation. Every entry in a pronunciation dictionary contains an orthographic transcript of a word with the corresponding phoneme sequence describing how the word should be pronounced. Orthographic transcripts of words can repeat in different entries showing different possible pronunciations for the same word.

Speech Corpus Design: The process of gathering prompts for recording by the speech talent. This also involves optimising the phonetic coverage of the speech corpus.

Stress (Syllable Stress): Is the emphasis on a certain syllable in a word for the purpose of emphasising on the word itself to indicate that it has more semantic importance over the rest of the sentence. Emphasis here does not necessarily correspond to a certain articulation process as stress could realise itself in different ways (increased loudness, pitch, vowel length…) (de Jong & Zawaydeh 1999).

Talent or Speech Talent: The person whose voice is recorded for the speech corpus.

TTS: Text To Speech, a complete system for converting raw text to spoken utterances. This involves normalisation, phonetisation and synthesis.

Utterance: A short script containing a small number of sentences (2 to 6) or a short recording corresponding to that script. The latter is sometimes referred to as “recorded utterance”.

Viterbi Algorithm: Is a dynamic programming algorithm for finding the most probable sequence of states of an HMM which generated the observation sequence.

Chapter 1: Introduction

Building a speech corpus – whether it is for speech recognition or speech synthesis – is a laborious and resource consuming task. In speech synthesis, corpus construction could include hiring a team of experts and native speakers to perform syntax error checks on the script before and after recording (recording in itself is resource consuming and should be supervised by experts); and aligning the phonetic transcript with the recorded speech which is the most time consuming (Yuan et al. 2013; Van Bael et al. 2007). Numerous methods to speed up the process have been employed in the past, most of which require previous speech data. This may be data that does not suit the target purpose, but could help in annotating the corpus automatically instead of human annotations which have a high cost and may result in disagreements between experts (Hosom 2009; Zue & Seneff 1996).

One of the reasons why Arabic speech synthesis falls behind the state of the art is the lack of resources. Specifically, the lack of recorded, segmented and annotated material suitable for recently developed speech synthesis engines. This lack of resources makes it more challenging to create the automatic tools required to speed up the corpus construction process.

The only speech corpora found in previous work were created by Almeman et al. 2013 which have been acquired from the author. Their multi-speaker speech corpus was built for speech recognition as it contains transcribed phrases with no granular segmentation at phone level. This corpus could be a valuable resource to this research and the possible uses of it will be investigated. The second is the “KACST Arabic Phonetics Database” (KACST) (Alghmadi 2003). This corpus was made for helping research in speech therapy, speech recognition and synthesis and is useful for this research as the recorded phones can be used for bootstrapping and training initial language models.

These corpora, although possibly useful for bootstrapping models to segment other speech corpora, cannot be used for modern speech synthesis engines (Unit Selection, DNN or HMM). These systems require hours (usually 2 or more) of speech recorded in a controlled environment. The level of control has in previous work varied from studio recordings to audio books to telephone conversations.

In this work, corpus design, recording and annotation are included as a complete process. The term corpus design is used in different ways in the literature. In speech technologies it usually means the selection of prompts to be recorded by the talent. This set of prompts should fit some criteria which the process of corpus design aims to fulfil.

It is intended to build a single speaker Modern Standard Arabic (MSA) speech corpus for speech synthesis, primarily for Unit Selection speech synthesis that could also be used for Statistical Parametric Speech Synthesis. The speech talent recording the corpus is a native Arabic speaker with a Levantine accent which is important to note because – as will be explained later – this affects the phonetics of MSA as some phonetic characteristics of the speaker's dialect could affect their MSA pronunciation as well (Watson 2007) which was observed by experts who supervised the recording sessions.

In section 1.1, the target synthesis method for which the corpus is built is presented and in section 1.2 the research contributions that will result from this effort are highlighted. It is important to note that the word "Arabic" will be used instead of MSA to describe things that apply to both Classical Arabic and MSA.

1.1 Target Synthesis Methods

In order to evaluate the corpus after it is built, Unit Selection and Statistical Parametric Speech synthesis were set as a combined target application of the corpus. The Unit Selection method of speech synthesis is one of the types of more general "Concatenative Methods" in speech synthesis. Other methods include diphone concatenation which produces less natural sound but require much less recorded speech and segmentation (Lenzo & Black 2000).

In concatenative speech synthesis a sequence of speech units are chosen from a unit database that is populated from the segmented and aligned speech corpus. The units are chosen by phone identity and other criteria such as prosody, position in phrase, position in word etc. Then, after performing acoustic modifications on the individual segments, they are concatenated to produce the desired utterance (Black 2002).

Along with Concatenative Methods, there are, what are called statistical parametric methods. The naming here is not always consistent. What is usually meant by statistical parametric methods is the assumption that the data is distributed by a probability distribution and the goal is to find the parameters for this probability distribution that optimises some criteria. The data in this case is the recorded speech with the aligned phonetic representation and the features extracted from the phonetic representation. An example of statistical parametric synthesis is HMM-based speech synthesis. In this type of synthesis the input text is converted into a sequence of phones and features representing context are extracted (part of speech, adjacent phones, pitch, prosody …). Based on these phones and features, a sequence of context dependent HMMs are chosen from the trained HMM database and these in turn generate the speech parameters (for example, mel-cepstral coefficients and the excitation). Then, the speech is synthesised from this low dimensional set of parameters using a vocoder such as STRAIGHT (Zen et al. 2007).

There are other types of statistical parametric speech synthesis methods that can be used but are not covered in this work. The literature review shows that the HMM-based speech synthesis is the most popular method used (Zen et al. 2007; Kim et al. 2006; Qian et al. 2008; Lu et al. 2011; Maia et al.

2007). However, recently, Deep Neural Networks (DNNs) have been used to synthesise speech successfully with good results (Zen et al. 2013).

1.2 Research Questions

Having identified the issue of resource scarcity in Arabic speech synthesis, a set of research questions is presented to show how this and other related issues are to be solved:

1- What is the phoneme set for MSA in a Levantine Accent? As shown later, this set could be different between different dialects in Arabic even if the speaker is speaking in MSA. If this is the case, which phonemes are common to all dialects and which are specific to the talents dialect?

2- What are the phonotactic rules that govern MSA phonology in general and how does it change for a Levantine speaker?

3- How accurately does an automatic segmentation system (HMM forced alignment) perform when using the phoneme sequence mentioned in the first research question, and a grapheme to phoneme converter based on the phonotactic rules mentioned in the second research question? This involves making adjustments to the HMM topology, boundary refinement and bootstrapping.

4- As a future work, it is intended to use the corpus to build a Unit Selection synthesiser and perform listening tests to evaluate the quality of the synthesisers based on naturalness, correctness and intelligibility metrics.

A broad overview of this work’s corpus construction process is provided in the following section.

1.3 Creating a Speech Corpus

Generally, the process of creating the speech corpora involves four stages: preparing a script, recording the corpora, generating the phonemic representation and aligning both together as the following sections explain (see Figure 1):

1.3.1 Preparing Transcript

Before recording, the transcript is gathered, corrected and normalised manually as no automatic normalisation system for MSA has been found. The script should originate from a source with relevant content. The content is then reduced (before or after correcting and normalising the transcript) to fit the cost requirements and while keeping phonetic coverage as high as possible within the cost constraints (this is referred to as optimisation). Section 2.1 covers this process in detail.

1.3.2 Recording

Recoding speech is not a trivial effort. Several hours of speech is usually required for Unit Selection and all the parts of the recording should be reasonably uniform in terms of speed (words per minute), loudness (the average amplitude of the speech signal) and mood (happy, sad, angry, singing…). In addition, the recording has to be of a decent quality and preferably recorded in a studio. Black 2002 explains in general the considerations that must be taken when creating a speech corpus.

There have been attempts to create Unit selection voices from recordings that were originally produced for different purposes such as news casts and audio books because of the availability of a transcript (Prahallad 2010; King 2013). This introduces issues in consistency; noise; background music and sounds which are not easy to remove. The transcripts of these recordings do not necessarily correspond to the actual recording, for example, a news anchor might make a mistake and include the word "apologies" with a correction in their speech which might not exist in the transcript. This is less likely to be the case when the transcript has been created prior to the recording and split into short sentences (utterances). This allows the rerecording of utterances where mistakes or mismatches with the script occur.

For unit selection, the whole recording is best done by one voice talent. This puts a great strain on the talent's vocal tract and requires a considerable amount of time and resources as the talent needs to take breaks.

1.3.3 Generating the Phonetic Representation of the Transcript (Phonetisation):

This could be done automatically depending on the language of the recording. In the case of English this requires a dictionary of phonetically transcribed words. In Arabic, this is a different task with less ambiguity as utterances are usually pronounced deterministically based on their written form – given that the transcript has the diacritics – but this is not necessarily always the case. This stage is important for both aligning and annotating the corpus and in speech synthesis front-ends (Malfrère et al. 2003). Section 3.1 covers phonetisation in detail by showing the set of rules used in phonetising MSA script and the irregularities (ambiguities) in these rules.

1.3.4 Aligning the Recording with the Phonetic Transcript:

If done manually, this is the most time and resource consuming out of the three stages. In this stage, each phoneme, syllable or other type of phonetic unit is assigned beginning and end time stamps in the recording. This is done in many ways and heavily covered in the literature (Hosom 2009; Van Bael et al. 2007).

The transcript could either, be done automatically and then optionally revised by a group of human experts, or done by a group of experts in the first place. Even the use of experts is not going to deliver 100% precision because it has been shown that there are always disagreements between experts (Hosom 2009; Van Bael et al. 2007; Zue & Seneff 1996). These disagreements mostly arise on boundaries between consonants and vowels; a consonant and a glide or a glide and a vowel. So the goal of the automatic alignment systems is to achieve as close precision as possible to a human generated alignment.

1.4 Structure of this Work

In this work, the four main stages of producing the speech corpus will be described, highlighting the contributions given. The script extraction, reduction (optimisation) and recording are illustrated in Chapter 2:, and then the phonetisation, annotation and alignment are described in Chapter 3:. In 0, the quality of the alignments and transcript are shown in the generated corpus. As shown in Figure 1, the entire corpus construction process is divided into “Script Generation and Speech Recording” and “Alignment”. The research contributions presented lie in the former. The following is an explanation of each of the activities in the workflow:

1- Scrape Aljazeera Learn Website (Aljazeera 2015): Collect Diacritised MSA script from a language learning website.

2- Transcript Reduction: Reduce size of script to fit the resource limits while maintaining phonetic coverage as high as possible. This balance between cost and phonetic coverage is why this process is called optimisation. It will be referred to as either reduction or optimisation depending on context.

3- First Correction: Orthographic and Syntactic Corrections: Experts perform 3 consecutive revisions of the script to correct orthographic and syntactic errors and normalise transcript.

4- Recording Utterances: Supervised by two experts and a sound engineer in a recording studio.

5- Second Correction: This is to match what was actually recorded to what is in the transcript.

6- Phonetic Transcript Generator (phonetiser): Automatically generates the phoneme sequence for each utterance. It can generate multiple possible pronunciations.

7- Segmentation and Alignment (Forced Alignment): The recordings and the phonetic transcript are aligned together using HMM forced alignment.

8- Manual Alignment Correction: about 15% of the corpus’ alignments are manually correct for bootstrapping and realignment.

9- Boundary Refinement: An optional automatic correction of boundaries generated from forced alignment.

10- Bootstrapping: using manual alignment to increase the precision of forced alignment.

The "Alignment" (bottom) part of the workflow could be repeated after more manual alignments or boundary refinements have been done until an accepted precision is reached.

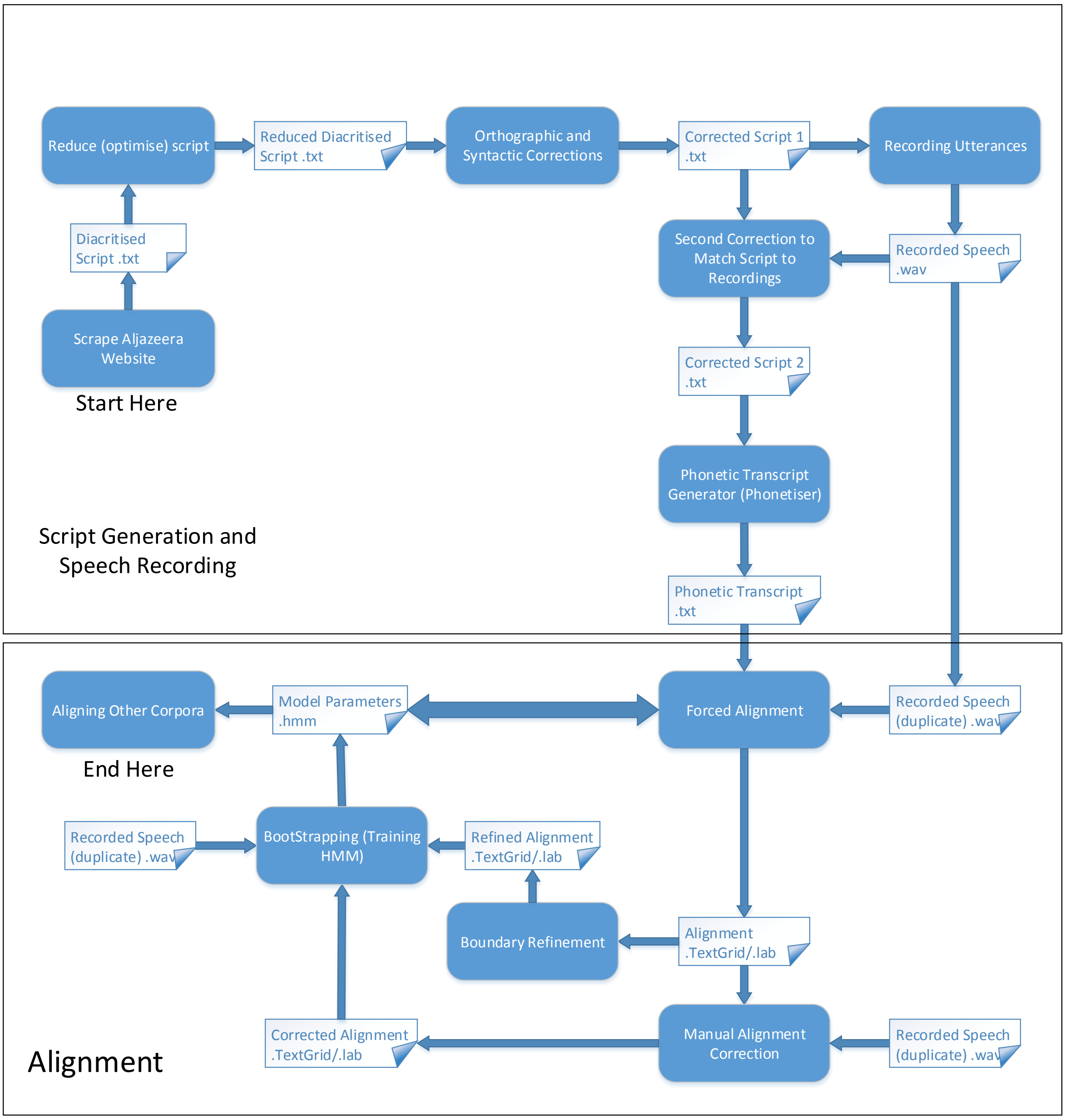

Reduce (optimise) script
Reduced Diacritised Script .txt
Orthographic and Syntactic Corrections
Corrected Script 1 .txt
Recording Utterances
Diacritised Script .txt
Second Correction to Match Script to Recordings
Recorded Speech .wav
Scrape Aljazeera Website
Start Here
Corrected Script 2 .txt
Phonetic Transcript Generator (Phonetiser)
Script Generation and Speech Recording
Phonetic Transcript .txt
Aligning Other Corpora
Model Parameters .hmm
Forced Alignment
Recorded Speech (duplicate) .wav
End Here
Recorded Speech (duplicate) .wav
BootStrapping (Training HMM)
Refined Alignment .TextGrid/.lab
Boundary Refinement
Alignment .TextGrid/.lab
Alignment
Corrected Alignment .TextGrid/.lab
Manual Alignment Correction
Recorded Speech (duplicate) .wav

Figure 1. Speech Corpus Construction Workflow

Chapter 2: Collecting and Reducing Transcript

The transcript was collected from Aljazeera Learn (Aljazeera 2015), a language learning website which was chosen because it contained fully diacritised text which makes it easier to phonetise. The transcript was split into utterances based on punctuation, to make it easier for the talent during the recording sessions.

After splitting the transcript into short utterances, the transcript was reduced (see Section 2.1) while maintaining acceptable phonetic coverage, then inspected to normalise the text and correct errors. This inspection was completed after reducing the transcript in order to decrease the manual labour required to clean the text, but this meant that the numbers and abbreviations were not included in the phonetic optimisation as they were not previously normalised.

After the reduction and before the recording, the text transcripts extracted from Aljazeera Learn were inspected and normalised. Abbreviations and numbers written in digit form were converted to word form. This is because the talent expressed that it was difficult to produce the correct inflection for numbers phrases while reading, if they were not written as words. In this phase, unwanted characters were removed and replaced with their word representation for example '$' which is converted to "دولار" which means "Dollar". After the inspection, only Arabic words were left in the transcript.

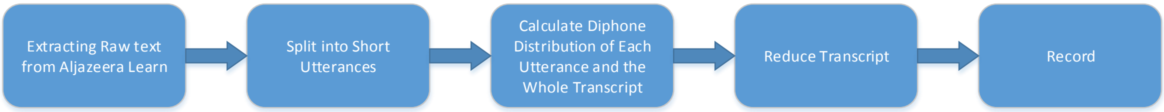

Extracting Raw text from Aljazeera Learn
Split into Short Utterances
Calculate Diphone Distribution of Each Utterance and the Whole Transcript
Reduce Transcript
Record

Figure 2. Collection and Reduction of Transcript.

2.1 Optimisation (Transcript Reduction)

All the works reviewed for corpus optimisation for speech synthesis use greedy methods (François & Boëffard 2002; Bonafonte et al. 2008; Kawai et al. 2000; Kawanami et al. 2002; Tao et al. 2008). Greedy methods as explained in the "National Institute of Standards and Technology" (Black 2005) are methods that apply a heuristic that finds a local optimal solution that is close to an initial solution. The initial solution and the heuristic/s were different between works in the literature. Also the unit of choice for optimisation (triphone, diphone, phone…) varies. It is important to say that greedy methods do not guarantee the production of a globally optimal solution as the corpus selection problem is Non-deterministic Polynomial-time hard (NP-hard) (François & Boëffard 2002) which needs a brute force search to find the optimal solution. This requires astronomical processing power as the number of possible solutions 2^n where n is the number of sentences. In our case the number of solutions is 2^{2092} which is greater than 10^{600}.

François & Boëffard 2002 classified greedy algorithms into three categories:

1- Greedy: The initial solution is the empty set and then utterances that increase coverage the most (relative to solution at iteration) are added to the solution. This is until certain target coverage is achieved or a limit is reached.
2- Spitting: The initial solution is the whole sentence set and then sentences that are least contributing to coverage are removed iteratively until a utterance removal would damage coverage in some way.
3- Exchange: Starting from a specific solution (could be the output of one of the two methods above) exchange on of the solution's utterances with one of the utterances excluded from the solution if this exchange increase coverage. Until no increase in coverage is possible. This maintains a static set size.

François & Boëffard 2002 used diphone as their unit and did not mention prosody or stress in units. The criteria for the three different approaches above are simple. They used unit counts from each sentences to give a score. "Useful units" in a sentence being units that would contribute to the corpus coverage (taking into account the need to have multiple units with the same identity. 3 in their case) and "useless units" being the units that are redundant as the set already has a number of units with the same identity that equals or is higher than the limit (3 is the limit chosen in this work. See *Table 1*). They have used unit counts with the sentence cost (length) in different ways which they compared. They have shown that using "Spitting" after "Greedy" methods improves coverage cost (number of chosen sentences and their average length) but does not necessarily increase phonetic coverage. The way they combined the two methods is by running "Greedy" and then running "Spitting" restricting its choice of sentences to the output of "Greedy".

Table 1. Statistics of this work's transcript before and after reduction. The chosen limit is 3 (blue).

Minimum number of occurrences for each diphone	**1**	**2**	**3**	**4**
Number of utterances	468	700	**884**	1025
Number of Words	5624	8982	**11560**	13479
Recording length	~ 1.1	~ 1.6	**2.1** **(3.7 hours with nonsense sentences (see Section 2.3))**	~ 2.5

Since in this work the primary concern is coverage and not necessarily length of corpus, but the length of the generated speech (2 hours maximum for proper utterances), the "Spitting" method was chosen to reduce the transcript to a size that would potentially generate between 1.5 and 2.5 hours of speech. In future work, a combination of the above methods could be used.

To choose criteria for iteratively choosing utterances, we adopted a simple count where each utterance is scored by the following formula:

$$US(U,C) = \sum_{k=0}^{n} \frac{UUF_k(U)}{CUF_k(C)} \qquad if\ CUF_k(C) > UUF_k(U)\ \ for\ all\ k$$

$$US(U,C) = -1 \qquad\qquad otherwise$$

Where $US(U,C)$ is the “Utterance Score” of the utterance U relative to corpus C, $UUF_k(U)$ is the “Utterance Unit Frequency” which is the number of times a specific unit indexed by k appears in the sentence U. $CUF_k(C)$ is the “Corpus Unit Frequency” which is the number of times a specific unit indexed by k appears in the corpus C at a certain stage of the optimisation.

The optimisation process started from the initial solutions being the whole set of 2092 utterances and iteratively removed utterances which had the lowest $US(U,C)$, excluding utterances which have a score of -1. The processes stopped when removing any utterance would cause at least one phonetic unit to occur in the transcript less than the allowed limit. The allowed limit was a controlled parameter.

In this work, diphones were used as basic phonetic units for optimisation. The reason for using diphones as the unit of choice is the fact it is the most used one in the literature reviewed (Kelly et al. 2006; Kominek & Black 2003; Barros & Möbius 2011; Bonafonte et al. 2008; Matoušek & Romportl 2007a) and the numbers of possible units for each phones, diphones and triphones (see Table 2) favour choosing diphones. Optimising using phones as units is trivial as there are only 82 chosen for the optimisation (see Section 2.3) and phone optimisation is not ideal as it is well known that some co-articulation effects between phones spoken in sequence are not reproducible when using phone segments from different contexts, which in the case when phone optimisation is ignored. Triphone optimisation has been used in the literature (Matousek & Psutka 2001), But no coverage measure was given in this PhD work to compare against diphone optimisation.

In this PhD work, 3 occurrences of each unit as a target is assumed and diphones are chosen as the target unit. Triphone optimisation was excluded as it means that there has to be at least $3 * 551368 = 1654104$ triphone instances occurring in the corpus and this is too good to be true as the unit distribution always follows biased distributions in human generated transcripts. But assuming this prefect scenario, and in our target 12000 word corpus, every word should contain more than 100 unique and novel triphones. This is a very unrealistic constraint which is shown more clearly in Table 2 containing all the possible frequencies of each phone type and the corresponding value in the corpus before optimisation.

Table 2. Theoretical Unit frequencies for different types of units

Phones	Diphones	Triphones
82	$82^2 = 6724$	$82^3 = 551368$

2.2 Optimisation Vocabulary

Please refer to the “Arabic Phonology” appendix for more information about MSA phonemes used in this work.

Not all diphones were included in the optimisation. The optimisation only included “short syllable diphones” and “half syllable diphones” (see *Table 3*). Both of these terms are used in this work for convenience and are not defined elsewhere. In this work, a short syllable is a syllable starting with a consonant (could be geminated) and ending with a vowel (could be long), and a half syllable is the second part of a syllable ending with a consonant (a vowel followed by a strictly non-geminated consonant).

Table 3. Diphones included and excluded from optimisation. V means long vowel and C means geminated consonant.

Short syllable diphones	**Half syllable diphones**	**Excluded Diphones**
cv cV Cv CV	Vc Vc	Cc

2.2.1 Short Syllable Diphones

Some short syllable diphones were excluded for the following reason: Emphatic consonants cannot be followed by a non-emphatic diphthong or a non-emphatic /a/ or /a:/ which are (ا) and (ـَ) in Arabic script correspondingly. This excludes $14 * 2 = 28$ diphones of this form.

The validity of these exclusions was only theoretical and based on rules of Arabic phonology before the recording (Watson 2007), but were found to be true in the talent’s speech, as the experts found during the correction phase, after the recording. The talent never emphasised a diphthong after a non-emphatic letter or vice versa.

According to the above, theoretically, there are $56 * 10 = 560$ possible short syllable diphones. 56 represents the number of consonants doubled to include geminated consonants. 10 represents the number of vowels. This exclusion leaves $560 - 28 = 532$ diphones included in the optimisation.

2.2.2 Half Syllable Diphones

The above short syllable diphone set, explained earlier, covers syllables of the form “cV”, “CV”, “cv” and “Cv”. But in case syllables of the form “cvc”, “Cvc”, “cVc” or “CVc” are to be synthesised by a concatenative speech synthesiser, which have a consonant coda (syllable ending) that is not followed by a vowel (otherwise the coda would have belonged to the following syllable), it would be useful to have segments of the form “vc” or “Vc”, where the consonant (“c” part) is followed by a pause or another consonant rather than a vowel. This is because consonants which are followed by a vowel are highly co-articulated with the following vowel (Yi 2003) making them unfeasible to use for concatenatively creating syllables which end with a consonant as these are not followed by a vowel and hence should not include this co-articulation effect. Half syllable diphones of the form “vc” were added to the phonemic vocabulary. *Table 4* shows how a concatenative speech synthesiser would hypothetically create each of the heavy and super heavy syllables ending with a consonant coda. It is important to note that for the diphones “vc”, the vowel in this diphone could either be a long or short vowel as their identity is merged just for the purpose of optimisation. This is because it is assumed that when concatenating “cv” and “vc” diphones to create a heavy or super heavy syllable, the length of the vowel in the syllable is determined by the vowel in the first syllable.

Ignoring long vowels and geminated consonants in half syllable diphones (as explained above) leaves 6 vowels (one of which is emphatic) and 28 consonants (168 possible half syllable diphones). A further exclusion would be of diphones which are made up of a non-emphatic vowel /a/ followed by an emphatic consonant. This leaves the inclusion of $168 - 1 * 5 = 163$ half syllable diphones.

Table 4. How to generate heavy syllables from short and half syllable diphones.

Short syllable	**Half syllable (the vowel corresponds to the vowel in the short syllable)**	**Heavy and super-heavy syllable**
Cv	vc or Vc	Cvc
Cv	vc or Vc	Cvc
cV	vc or Vc	cVc
CV	vc or Vc	CVc

This PhD work also included consonants at phrase endings (before a pause) as part of the phonemic vocabulary. Silence (represented as “sil” in this work) is considered a phone in its own right. This is to avoid any effect of co-articulation on the consonant being followed by another phone (consonant or vowel) and this consonant can be used at the end of phrases by concatenative speech synthesisers and the concatenation point would be the region of low amplitude before the consonant (Yuan et al. 2013). This adds $56 * 1 = 56$ diphones in the optimisation. 56 is the number of consonant phonemes including geminated consonants. 1 is the pause (“sil”) phoneme.

2.2.3 Consonant Clusters

As for other types of diphones, consonant clusters, only two consecutive consonants allowed in MSA as described in Ali & Ali 2011 – which will be referred to as “cc” – were not included in the optimisation for three reasons:

1- “cc” diphones constitute a big part of Arabic diphones. Theoretically, there are $28 * 28 = 784$ possible “cc” diphones in Arabic out of 6724 total diphones. So being able to exclude them from the optimisation process, makes the possibility of reducing the dataset size higher and simplifies the problem. But the question is: How much would this damage the phonetic and prosodic coverage in the corpus?

   It is important to note that “Cc”, “cC” and “CC” diphones are not possible in MSA (could be in other dialects). This is because consonant cluster of more than 2 are forbidden. This further excludes $3 * 28 * 28 = 2352$ diphones from the total 6724.

2- The 784 theoretically possible “cc” diphones are not all occurring in Arabic (not including foreign imported words. 246 “cc” diphones are either non-occurring or very rare in Arabic (John Alderete 2009). The study that these numbers were taken from does not state specifically which “cc” diphones these are, but states to which consonant classes (articulation type) each of the consonants in the diphone belongs. So it is safe to assume that many of these clusters will not be found in the corpus transcript used for this work before optimisation.

3- Yi, Jon Rong-Wei 2003 show how certain concatenation points between specific types of phones are better than others and would generate natural sounding speech when used in concatenative synthesisers. One of these, is the very brief period of silence and gathering of pressure before the release of a stop letter and other consonants which involve the same phenomena on a different scale (Tench 2015; Yi 2003). This could make it possible to construct those consonant clusters from smaller units by concatenating at the low amplitude region before the consonant. It has been noticed after the recording that the region of low amplitude is clear before stop consonants and less significant before other consonants. To try to alleviate this issue, a consonant from each of the articulation categories was chosen and for each an utterance from the recordings selected. The low amplitude before these consonants was further de-amplified (dimmed) and no effect to naturalness was noticed by the experts. Subjective testing will be conducted later to further justify this finding. The de-amplification of the low amplitude period shows that these points can be used as concatenation even when the consonant is not a stop.

2.3 Results

Table 5 lists results based on all the $163 + 532 + 56 = 751$ diphones that were included in the optimisation. For more detailed results please refer to Halabi (2015). After running the optimisation script, 884 utterances were left in the data set out of the complete 2092. The optimisation process was run through several times with the threshold for the allowed minimum number of diphone occurrences changed. The threshold 3 was chosen because of resource limitations (15 hours recording studio time and talent time) and more utterances were planned for recording in case extra studio time was left (see *Table 1*).

It is important to note here that even with the threshold chosen at 3, this does not guarantee that all diphones have occurred at least 3 times in the optimised corpus. Diphones that occur less than the chosen threshold – before the optimisation started – were not included in the optimisation process and any utterance that includes them is never excluded.

To cover the gap of these underrepresented diphones, 896 nonsense utterances were recorded. Nonsense utterances have been used before in the literature to study language phonetics (including Arabic) (John Alderete 2009; Kain et al. 2007; Laufer & Baer 1988). The benefit of using them is being able to cover many units with less material but a talent may find them more difficult to pronounce and this could potentially slow the recording time and cause more errors in the final recording output. This is also because of the absence of syntax which makes the prosody of the generated utterances potentially random. The nonsense utterances used here are experimental and after recording them, the talent did express that they were more difficult than news transcripts, but the fact that they were generated by a template made the effort easier as the talent recorded more of them as they were similar in length and orthographic structure and utterances from the same template were grouped together on the prompt shown to the talent. The nonsense utterances were automatically generated using 4 templates (The sections between brackets are replaced by a short syllable diphone to generate a nonsense utterance and underlines represent stress. Some stress depends on the diphone which is not shown):

1- /(cv)Sbara wata(cv)S~ara watu(cv)SA(c)un taSar~u(cv)/
2- /(cv)sbara wata(cv)s~ara wati(cv)sU(c)in tasar~u(cv)/
3- /ta(Cv)Saw~ara wata(Cv)Sara watu(Cv)Sa taSa(Cv)/
4- /ta(Cv)saw~ara wata(Cv)sara watu(Cv)sa tasi(Cv)/

Templates 1 and 3 guarantee that all short syllable diphones with emphatic vowels are included, and templates 3 and 4 guarantee that all short syllable diphones with geminated consonants "C" are included, and sentences 1 and 2 mild /u1/ and /i1/ short syllable diphone are included. All the templates repeat the same diphone in different locations in the word to include stressed and non-stressed diphones. Note here that the replacement is only done orthographically. The eight vowels in Arabic, the 28 consonants and the 28 geminated consonants were used to replace "v", "c" and "C" respectively. But those vowels (including in diphthongs) are uttered emphatically or non-emphatically depending on the context in the template. This generated a total of $28 * 8 * 4 = 896$ nonsense utterances that cover all the short syllable diphones (four times each at least) with different stress. Half syllable diphones were not included as this would have doubled the amount of recoding required.

It is suggested that future work could add emphatic, non-emphatic, stressed and non-stressed vowels (a stressed vowel being a vowel in a stressed syllable) as separate phonemes in the optimisation process. This would require much more data as shown in the results.

Table 5. Coverage statistics for different parts of the transcript.

Part	**Aljazeera before optimisation**	**Aljazeera after optimisation and normalisation**	**Nonsense utterances**	**Aljazeera after optimisation with nonsense utterances**
Number of diphones covered at least once	561	544	547	**669**
Percentage of diphones covered at least once	74.70	72.44	72.84	**89.08**
Number of diphones covered at least three times	492	476	545	**646**
Percentage of diphones covered at least three times	65.51	63.38	72.57	**86.02**

Finally, the short and half syllable diphones left a total of 896 + 884 = 1780 utterances for the recording (see Section 2.4 for more information on the recording and error correction procedures). The coverage of these utterances is shown in *Table 5* for each of the nonsense utterances and the news transcript and both combined. *Table 6* shows the complete set of phonemes used in this work excluding geminated consonants which are represented by doubling the consonant phoneme's symbol. The symbols on the right of the columns will be used to refer to phonemes henceforth.

Table 6. Final Phoneme set (82 in total). Note that geminated consonants are not included in the table for simplicity purposes. The left hand column in each section represents the phoneme in Arabic script and the right hand column *is the Buckwalter representation.*

أ	<	ر	r	غ	g	ي	y	ـُ	u0	[ـِ]	i1
ب	B	ز	z	ف	f	ڤ	v	ـِ	i0	[و]	uu1
ت	T	س	s	ق	q	پ	p	(ا)	AA	[ي]	ii1
ث	^	ش	$	ك	k	چ	G	(و)	UU0	([ـُ])	U1
ج (ʒ)	j	ص	S	ل	l	ج (d͡ʒ)	J	(ي)	II0	([ـِ])	I1
ح	H	ض	D	م	m	ا	aa	(ـَ)	A	([و])	UU1
خ	X	ط	T	ن	n	و	uu0	(ـُ)	U0	([ي])	II1
د	D	ظ	Z	ه	h	ي	ii0	(ـِ)	I0	pause	sil
ذ	*	ع	E	و	w	ـَ	a	[ـُ]	u1	distortion	dist

Green means: Only in foreign words used in Arabic like فيديو

Blue means: Vowels

Black means: Consonants

2.4 Recording Utterances

The recording of the corpus was spread over 5 days. Each day involved a 3 to 4 hour session including one or two breaks to avoid straining the talent's voice. This is the same time as reported by Matoušek & Romportl 2007b and two hours more than Oliveira et al. 2008. The fifth recording day was used to go through the recordings and rerecord unreliable utterances. A sound engineer, the voice talent and at least one expert were always present during the recording. The experts provided feedback to the talent about speed, emotion, loudness and pitch consistency and errors in pronunciation. The sound engineer started each session with a sound check to test if the talent was within an acceptable distance from the microphone for human voice recording and to produce recordings with consistent loudness. Loudness and speed were less of an issue as long as the talent spoke within a comfortable range set by the sound engineer. The sound engineer was able to change the speed and intensity (loudness) of recordings based on the experts' opinion and the readings from the software used (Pro Tools 11) without affecting the naturalness of the recordings.

The sound engineer also played recordings from previous sessions to the talent at the start of each session and when the experts felt that the talent was deviating from the acceptable ranges described above. The talent was a native Arabic speaker and recordings were repeated on request if he felt it wasn't suitable for our purpose.

The recording was done in a studio. The microphone used was "Neumann TLM 103 Studio Microphone" known to be used for high quality human speech recordings. It had a pop shield to reduce the sound impact of exhaled air on the microphone. The talent sat in a soundproof anechoic recording booth. The booth only contained a prompt screen and the microphone. After the recording was finished, the sound engineer went through the whole recording in order to perform the following edits:

1- Adding short silences at the beginnings and ends of utterances. This is needed to give each recorded phone a context (a preceding and trailing phone) as pauses are modelled as phones in HMM forced alignment, which is used later.

2- Preforming "Dynamic Range compression" for the intensity (loudness) of all the utterances. This is used to make intensity as uniform as possible with a dynamic gain that is multiplied by the signal to keep the signal within a set limit. -12 db (Decibels) was chosen by the sound engineer but it is possible to re-export the output with different limits.

3- Reduce the length of speech pauses that are too long. No specific length was agreed but the sound engineer was given feedback about long pauses to reduce them which keeps acceptable variability in pause length without jeopardising the automatic alignment whose precision might be affected by long pauses.

4- Normalise speed (change utterance's' speed each separately to a predefined speed).

The sound engineer was also given feedback after the second error correction phase about the errors still in position, in order to fix them and redeliver the recordings. The errors included only clipped phones next to pauses, unreliable edits (recording radically different from transcript) and speed inconsistencies.

The recordings were delivered in separate files for each utterance (1780 files in total with 33 extra utterance because of residual studio time) which correspond to 17040 words overall after transcript corrections. 896 of the utterances correspond to the sentences that were automatically generated. The rest correspond to the optimised automatically chosen utterances from Aljazeera Learn (Aljazeera 2015) (see Section 1.3.1). Each utterance starts and ends with a short pause (about 100ms). The speech was not delivered in one large file as it is known that sequence models align shorter utterances more accurately than longer ones (Moreno et al. 1998). But still, having utterances with different length is usually considered a goal as it may enrich the prosodic coverage of the corpus (Umbert et al. 2006; Vetulani 2011). This corpus's utterance statistics are shown in *Table 7*. It is not claimed here that these statistics are optimal. The lack of diacritised Arabic text constrained the choice of utterances and the optimising utterance length distribution is not covered in this work.

Table 7. Recording Statistics.

	Total Utterances	**Nonsense Utterances**	**Proper Utterances**
Count	1780	896	884
Average duration (sec)	7.481522	5.915011	9.046307
Mode duration (sec)	5	5	5
Max duration (sec)	36	8	36
Min duration (sec)	1	3	1
Total Duration (hours)	3.7	1.5	2.1

After the recording sessions were over, two experts went through the corpus in sequence (for more scrutiny) to correct orthographic errors in the transcript and to change the transcript so that it reflected what was actually pronounced by the talent. All punctuation was removed and a special symbol was used to represent a pause in this phase. Most pauses were easy to detect as they were long enough (over 0.3 seconds). Due to some pauses or errors being hard to detect in normal speed, the speed of the recordings was slowed down in this correction phase. Even with the speed reduced, it was hard to detect some hesitations in word boundaries and to decide whether to classify them as a pause or not. The decision was made to classify as a pause, any word boundary that could be pronounced more naturally if the two phones surrounding the boundary were uttered closer to each other. This is justified by the fact that if these two phones are not naturally following each other, the pause mark would tell the synthesis system that these two phones do not naturally follow and their concatenation (in case of concatenative speech synthesis) would be give a high cost (less likely to be chosen) (Yi 2003).

Later, in the phonetic “manual corrections” phase (see Section 3.4), experts were allowed to remove or add pauses that were incorrectly added or missed in the transcript. In this phase, it is easier to classify a segment as a pause or not, because the signal’s spectrum and amplitude is visible to the expert.

Chapter 3: Corpus Segmentation and alignment

The terms segmentation and alignment are used interchangeably in the literature to describe the general processes of annotating a speech corpus with phone labels and finding the timestamps of the boundaries that delimit those phones. This could involve annotating pauses and stress (Braunschweiler 2006).

In this work, the term segmentation will be used to refer to annotation of the speech corpus with a sequence of phone labels taken from the phonetic transcript of this corpus which is in turn automatically generated from the textual transcript as described (see Section 3.1). Segmentation also involves finding boundaries that surround these phone labels. The timestamps of these boundaries do not have to be 100% accurate (or anywhere close to that) in segmentation, just the sequence of phone labels should match what is in the audio. Both the creation of the phonetic transcript from the textual transcript and the segmentation of the corpus are done automatically in this work. The former has been completed by an algorithm developed in this work and the latter using HMMs built using the Hidden Markov Model Toolkit (HTK) framework (Young et al. 1997).

Alignment here is the determination of the exact timestamps of the phone boundaries. This could be done either automatically (using boundary refinement techniques for example or HMM models as described above) or manually by a group of experts whose job it is to correct the boundaries generated from the segmentation (or they could perform segmentation and then alignment manually which is known to be very time consuming). Note that the segmentation process could produce high precision alignments as shown in previous works (Hosom 2009). This depends on the quality of the recording, speech, text transcript, phonetic transcript and the algorithm used for segmentation (and alignment in this case).

To assess the necessity of experts aligning the corpus manually, the experts manually corrected a portion of the corpus. This correction was used to assess the quality of the automatic segmentation, the experts' agreement and also the quality of any further alignments carried out using the same algorithm with different parameters or using the manually aligned data to bootstrap the automatic segmentation process.

In this work, the size of the corpus created exceeds 3 hours of speech. To avoid manual segmentation of the corpus, forced alignment (Murphy 2012) was used in different modes to create an initial segmentation of the corpus (see Section 3.3). This was carried out after the corpus transcript was revised twice by the experts, so at this stage, the corpus transcript had to be converted into a phonetic transcript to be used for segmentation and alignment. The following is a summary of the steps of the segmentation and alignment process:

1- Generating the phonetic transcript: Text transcript is automatically converted to a phonetic transcript which includes phonemes from *Table 6*. In this PhD work, the phonetic transcript was in the form of a pronunciation dictionary because the software used for alignment requires a pronunciation dictionary as input with the textual transcript.

   The dictionary contained several possible pronunciations of each word in the textual transcript.

2- Automatic Segmentation: The phonetic transcript and the speech corpus audio are used as input to forced alignment which produces the segmentation with initial boundaries.

3- Manual corrections: Three experts go through a portion of the segmented corpus to correctly align the boundaries with the speech. This could be repeated iteratively where in each iteration the corpus is automatically realigned after the system is trained on the manually aligned data (some left for evaluation) and then the precision of this alignment is calculated to determine if an acceptable precision has been reached or if precision is not increasing anymore with more iterations.

3.1 Generating the phonetic transcript

This was done automatically using a set of rules taken from classical Arabic orthography rules (Elshafei 1991; Thelwall & Sa'Adeddin 2009; Watson 2007; Ali & Ali 2011; Gadoua 2000; de Jong & Zawaydeh 1999; Halpern 2009); the nature of the text transcript harvested from the web and the dialect of the speech talent (Levantine from Damascus). The experts have noticed that different segments of the text taken from different articles applied different rules for orthography. This was dealt with by iteratively creating a list of all these rules. During the text transcript's error correction stage (see Section 1.3.1), the experts discussed and assembled what they found and added a list of rules as they corrected the script. The complete list of rules for generating the phonetic transcript is as follows:

1- All characters that are not Modern Standard Arabic letters or diacritics are omitted. Even Arabic letters in classical Arabic that are no longer used need to be omitted. Letters to be excluded are shown in the table below (see *Table 8*).

Table 8. Classical Arabic characters excluded from the transcript.

Description	Unicode	Arabic Script
Arabic Tatweel	U+0640	ـ
Subscript Alif	U+0656	◌ٖ
Superscript Alif	U+0670	◌ٰ
Alif Wasla	U+0671	ٱ

All punctuation characters are also omitted. This is because the experts located the pause locations during the manual correction of the textual transcript (see Section 1.3.1). This renders the punctuation characters useless as the locations of pauses are already known. But it is important to note that the punctuation could be used later for prosodic feature extraction as the prosodic features of utterances correspond strongly with punctuation (Taylor 2009).

2- Arabic orthography is described as a phonemic orthography (sometimes Arabic script and alphabet are called “phonetic”, having the same meaning) and the correspondence between letters and phones has been studied in the literature (Watson 2007; Elshafei 1991; Newman 1986) this allows one to think of Arabic letters as phonemes. However, as will be shown, this is not always the case. Arabic symbols (letters and diacritics in this case) usually correspond to phonemes in a regular manner. Unlike English spelling where – for example – the word “enough” has combinations of letters that offer different possible pronunciations from “enouw” where the “ough” combination is said as in the “bough” of a tree to “enoch” with the “ou” as in “though” and the “gh” as in a Scottish loch and yet the word is pronounced as “enuf”. Similar but rarer instances of this phenomenon exist in Arabic. Arabic (including MSA) includes a set of words (nouns and function words) which have an implicit “Alif” vowel (or “ا” in Arabic orthography) which is not written and corresponds to the /aa/ vowel phone in table (see *Table 9*). This set of words is small and unchanging. The system uses a table lookup method to resolve those words when they are encountered where the phonemic transcriptions of each of these words is predetermined by the experts. Note that these words could be affixed or suffixed but their pronunciation stays the same.

Table 9. Irregularly pronounced words in Arabic.

Arabic word	Pronunciation	Arabic word	Pronunciation
هَذا	/h aa TH aa/	ذَلِكُمْ	/TH aa l i0 k u1 m/
هَذِهِ	/h aa TH i0 h i0/	أولَئِكَ	/AH u l aa AH i0 k a/
هَذانِ	/h aa TH aa n i0/	طَهَ	/T aa h a/
هَؤُلاءِ	/h aa AH u0 l aa AH i0/	لَكِنْ	/l aa k i1 n/
ذَلِكَ	/TH aa l i0 k a/	رَحْمَنْ	/r a H m aa n/
كَذلِكَ	/k a TH aa l i0 k a/	لله	/l AA h/

3- Manually annotated silences were represented by the phone /sil/ in the phonetic transcript.

4- All consonant letters except Waw and Ya’ (“و” and “ي” correspondingly) are simply converted to their phonetic representation without ambiguity (see table). An exception is when the consonant is followed by a Shadda or (ّ), then it is represented by a doubling of the consonant’s phonetic representation. For example, /b/ for “ب” becomes /bb/ for “بّ”.

5- Ta’ marboota or “ة” is converted to “t” if followed by a diacritic, otherwise it is ignored.

6- Madda or “آ” is converted to a glottal stop /</ followed by /aa/ or /AA/ long vowels based on the amount of emphasis.

7- Vowels are emphasised if they follow or precede an emphatic consonant with exception of /x/ and /g/ (“خ” and “غ”) which only affect following vowels and not preceding ones. Emphasis is represented by capitalising the vowel’s phonetic transcription’s representation (see table).

8- Short vowels /i/ and /u/ corresponding to diacritics (ـِ) and (ـُ) have – in addition to the possibility of being emphasised – the possibility of being leaned towards /a/ or (ـَ). This means that the pronunciation of the /i/ or /u/ will be closer to a Schwa. The phenomena is not documented anywhere and was noticed by the experts after recording the corpus. The talent leaned towards /i/ and /u/ when they preceded a word-ending consonant which is not followed by a short vowel. In the phonetic transcription this is represented by the numbers 0 and 1. 0 meaning “not leaned” and 1 meaning “leaned”. For example, the /i/ in the word “مَغْرِبْ” which means “west” or “morocco” (which pronounced as /m a g r i1 b/) is phonetically represented as /i1/. (See *Table 6*).

9- Waw and Ya’ (“و” and “ي”) are transcribed phonetically as either vowels or consonants. This is determined by their context. If followed by a vowel, they are identified as consonants. If followed by a consonant then the preceding phone determines their identity, if preceded by a vowel, they are consonants, otherwise vowels.

10- Alif (“ا”) is transcribed as a vowel /aa/ or /AA/ depending on emphasis. An exception is a type of Alif called Hamzat Alwasel which is not pronounced in Arabic (including MSA). Also, Hamzat Alwasel becomes a glottal stop /</ at the beginning of sentences or phrases (after silences). Alif is realised as a Hamzat Alwasel when it is the first letter in the word or the second (after an affix).

The phonetic transcription produced was in the form of a pronunciation dictionary similar to the ones used in speech recognition systems for example HTK and Sphinx (Young et al. 1997; Lamere et al. 2003). The dictionary is a long list of orthographic representations of words each followed by their corresponding phonetic transcript. Note that multiple repetitions of the same orthographic representation can occur showing different possible pronunciations. “Hamazt Alwasel” in rule 10 when not in the beginning of the word is ambiguous and could be pronounced or not. Both pronunciations were added to the dictionary to be resolved in the forced alignment stage as HTK will choose the most probable sequence of phonemes that generated the speech signal. Other instances of ambiguity are Alif “ا” after Waw “و” at the end of a word. Here the Alif is not pronounced if the Waw is a plural Waw which is difficult to automatically determine with high precision (as in foreign words transliterated into Arabic). For example, the word “Nicaragua” is written “نيكاراغوا” in Arabic and the final Alif represents a long vowel phoneme /aa/. Both possible pronunciations for each word ending with a Waw followed by an Alif were included. Word-ending long vowels were also optionally shortened in the pronunciation dictionary due to the phenomena of vowel reduction (de Jong & Zawaydeh 1999; Biadsy & Hirschberg 2009) which was noticed in this corpus as well.

3.2 Automatic Segmentation

The automatic segmentation was done using flat start forced alignment in a similar way to the method described in the “HTKBook” (Young et al. 1997). HTK version 3.4.1 was used which was the most recent version at the time the segmentation was conducted. HTK contains several tools to perform tasks such as extracting acoustic features like the Mel Frequency Cepstral Coefficents (MFCCs) (see the “Acoustic Features” appendix for more information about MFCC) from the raw speech signal; constructing (training) HMM models (HCompV, HRest and HERest) from aligned and non-aligned data (the former being flat start training); using previously trained HMM models to align new data with the transcript using Viterbi decoding (Murphy 2012) (HVite); and performing other text processing tools (HHEd, HCopy…). These tools were built mainly for speech signals but HTK has been used for other purposes. For more depth on what exactly each of these tools do, please refer to the HTK book (Young et al. 1997).

Because of the complexity of the HTK training scheme, which is due to the fact that it requires manual manipulation of the text files between stages of training and alignment; and the complexity of HTK’s syntax used to write the HMM topology, a python wrapper was used to script the different stages and tasks. Another motivation for using and enhancing this wrapper was the fact that the training and alignment were conducted several times due to changes in parameter values and errors found in the results that required some alteration to the data. The wrapper used is called Prosodylab-Aligner (Gorman et al. 2011) developed by the Department of Linguistics, McGill University. The aligner contained two main features before modification in this work, HTK’s flat start training scheme (that generates and HMM model and also aligns the training data) and alignment using previously trained models. Flat start training is a term commonly used in the literature when the initial training stage is not done with manually labelled data and the input utterances are uniformly segmented. For example, an utterance that is 10 seconds long with 100 labels would be split into 100 segments each being 100 milliseconds long. Flat start trained HMMs usually produces less accurate alignment than aligning using HMMs trained with manually aligned data (D. R. Van Niekerk 2009; Brognaux et al. 2012) but this was used in this case as an initial alignment for the experts to start from when creating the manual alignments.

In this work, a third feature was added to the python wrapper which is to bootstrap (train) the HMM models using previously aligned data (data with timestamps of phone boundaries). All three features in the wrapper were modified to optionally allow different HMM topologies for different phones (it is possible to use a default topology for all phones). The three features were used in the following general stages:

1- HTK alignment: The output phonetic transcription system as described in Section 3.1 along with the raw audio is input to the python wrapper which in turn uses the HTK flat-start training scheme to generate the automatic alignments of the corpus.

2- Manual corrections: The output alignments are given to the linguistic experts for manual inspection and correction. The correction involves adjusting the boundaries of phones and correcting false phone labels, deleting labels for phones that did not exist in speech or adding labels for phones that were missed by the phonetic transcriber. The corrected alignments are used to calculate the precision of the automatic alignment of the different runs of stage 1 (with different parameters).

3- HTK bootstrapping: The output of stage 2 is used for further refining the automatic alignments by bootstrapping the training with manually corrected boundaries. This could be done iteratively until an acceptable precision is reached.

4- Boundary Refinement: Optionally, before or after stage 3 (or both) a novel approach to boundary refinement was performed. It was inspired by the results of the evolution of the first stage (see table). The results show strong tendencies of certain predicted boundaries of predominant boundary types to deviate from the correct boundary location by a regular interval (delta) both by magnitude and direction. For example, boundaries between vowels and consonants (v/c boundaries) have shown to have, over 80% of the time (only counting boundaries that were moved by the experts), a negative delta with an average delta of -0.01615 seconds.

The following is a more detailed description of the two first stages. Note that text processing done between those stages is not described here because it is redundant. Please see code for more details (Halabi 2015).

3.3 HTK Alignment

After calculating the MFCC acoustic features of the raw audio signal using HCopy, HCompV is used to calculate the initial means and variances of the Gaussians, whose mix makes up the observation probability distributions (Ghahramani 2001). These means and variances are the same for all phone models initially and is the global mean and covariance for all the data points (all frames for the audio signals). HCompV also generates Variance Floors (VFloors) which are lower bounds for the variances that can be used later to prevent over fitting by prohibiting the variance from going below those values at each training iteration. By default, the variance floors are just taken to be 0.01 times the global variance which was used in this work. Note that there are other ways of calculating variance floors that are not included in HTK (Young et al. 1997) and are not covered in this work.

The global means and variances generated by HCompV, alongside a default initial transition matrix are used to create the initial HMM definition. This is similar to HTK's initial model state and can be seen in Young et al. 1997.

The generated initial HMM models are then used iteratively as input to HRest. HRest updates the means and variances discussed above and also the transition matrixes of HMM states. HRest uses the Baum-Welch algorithm (Jurafsky & Martin 2008) which is based on the more general Expectation Maximisation (EM) method for probabilistic model parameter estimation when the probabilistic model contains hidden variables like in HMM. In EM, the goal is to maximise the data log-likelihood function which is given by:

$$L(\theta) = \sum_{i=0}^{N} P(X_i, Z_i|\theta)$$

where X_i are the observed variables, Z_i are the hidden variables, θ generally represents the model parameters and N is the number of data entries in the dataset.

Since no values for the hidden variables are known during training, the current estimates of the HMM parameters are used to find the expectation of the log-likelihood function which is given by:

$$Q(\theta, \theta^{t-1}) = \mathrm{E}[\mathrm{L}(\theta)|\mathcal{D}, \theta^{t-1}]$$

where $\mathcal{D}$ is the dataset and θ^{t-1} is the current estimate of the model parameters.

Then the function $Q(\theta, \theta^{t-1})$ is itself maximised with respect to θ which yields the new estimate of the model parameters θ^t as shown here:

$$\theta^t = argmax_\theta Q(\theta, \theta^{t-1})$$

The same is repeated for either a predefined number of times or if it reaches convergence. It can be shown that each iteration of the EM algorithm will either increase the value of the log-likelihood function or keep it the same for new estimates of the parameters θ. The monotonic increase of the above log-likelihood function at each iteration in the EM algorithm has been proven (Murphy 2012).

In the HTK implementation of Baum-Welch (EM algorithm) the number of epochs (iterations) is adjustable and different numbers were used to test precision increase.

After HRest has finished optimising the parameters of the phone models, HVite is used to generate the final alignment using the Viterbi algorithm (Murphy 2012) which is a dynamic programming method that finds the most probable sequence of states (hidden variables) that generate the observed variables in an HMM.

After one third of the iterations have elapsed, the aligner adds optional “pause” models between words in the script. HTK chooses the most probable pronunciation – given the HMM, the pronunciation dictionary and the silence (pause) models – which finally includes the boundaries and the detected pauses. This is done using finite state word networks that model all the possible pronunciations of the utterances (Young et al. 1997). These networks are built from the pronunciation dictionary which contains possible multiple pronunciations of the same word. This work’s phonetic transcription system (see Section 3.1) deals with ambiguous pronunciations of words by adding multiple entries in the output pronunciation dictionary. For more on using finite state networks for multiple pronunciations in speech recognition refer to Pereira & Riley (1996) and Young et al. (1997).

3.4 Manual corrections

According to Yuan et al. (2013), segmenting and aligning speech given only the phonetic transcript and raw audio with no initial segmentation could require as much as 400 times the audio time to finish with acceptable precision. This means that every minute of speech would take over 6.5 hours to segment. Segmenting and aligning the whole corpus produced (which is 3.5 hours long) would require around 1400 hours of work. As has already been noted there is a considerable amount of emphasis on the difficulty of segmentation and alignment in the literature (Van Bael et al. 2007; Malfrère et al. 2003; Mporas et al. 2009).

The alignments produced at stage 1 were used to decrease this time required. It is assumed here that correcting automatic segmentation and alignments is quicker than creating them from scratch. This manual correction stage has been done in previous work and is usually recommended for speech synthesis corpora (Peddinti & Prahallad 2011; Jakovljević et al. 2012; Black 2002) but it suffers from:

1- The huge efforts required to segment and align a medium sized corpus, of for instance 3-4 hours, for speech synthesis. The suggested solution in this work is an iterative method where experts correct small parts of the corpus, which are then used to realign the corpus automatically after bootstrapping with manually corrected data. This is done iteratively with accuracy calculated at every step to check if improved precision can be obtained without manually correcting the entire corpus.

2- Requiring a team of qualified linguists with good knowledge of the target language’s phonetics and training them on the conventions, phone sets, boundary types, potential errors and the software used for correction. Three 3-hour sessions were conducted before the segmentation and alignment tasks were distributed to train the experts. The experts kept in contact throughout the alignment to report common errors and to send enquiries.

3- Inter-expert agreement. This is due to the subjective nature of some phone boundaries or phone boundary types (Yi 2003). To solve this issue, each utterance was corrected at least twice by a minimum of two different experts. This helps in two ways. One is calculating inter-expert agreement which a measure of how close the expert's corrections were to each other (see Section 4.4.2). The other is to increase the precision of the alignment with more experts analysing the same set of utterances.

Each expert was given batches of 50 utterances per iterationand then the experts exchanged the utterances for the second correction. The software used for the correction of the boundaries was Praat (Boersma & Weenink 2015) which accepts a file format that Prosodylab-Aligner is able to generate. Praat was chosen as it was the only freely available tool for this purpose at the time of the experiment. *Figure 3* shows the interface of Praat used by the experts. Tier 1 contains the phone labels and boundaries to be corrected by simple keyboard and mouse actions. Tier 2 contains the Buckwalter representation of words in the original transcripts. These were not to be changed by the experts.

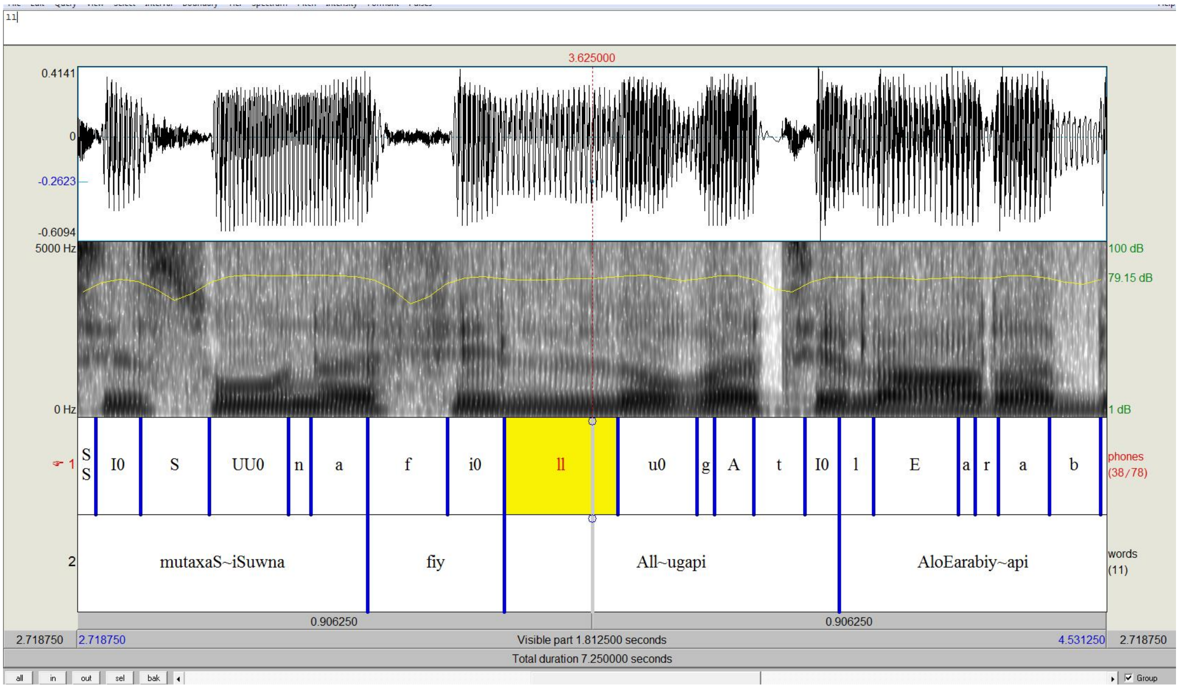


Figure 3. Praat interface.

Chapter 4: Segmentation and Alignment Evaluation

4.1 Evaluation metrics

To measure precision, the percentages of boundaries found by the system within a controlled distance from the correct boundary were calculated. This is the agreed on metric in the literature (Hosom 2009; Yuan et al. 2013; Mporas et al. 2009; Jakovljević et al. 2012). In most attempts, this percentage is used on all boundaries combined (Jakovljević et al. 2012; Yuan et al. 2013; Hoffmann & Pfister 2010) (with some boundary types excluded sometimes (Jarifi et al. 2008; Stolcke et al. 2014)). Some attempts calculated the precision for different boundary types separately (Hosom 2009). This inspired the inclusion of boundary types in this work, as this had not previously been done for MSA. The distances are all a multiple of 5 milliseconds. These distances are referred to sometimes as the tolerance T and the percentage of boundaries within a tolerance will be referred to as the precision for that tolerance $P_{T,B}$ where B is the boundary type. In addition, the average absolute value of delta D (absolute value of boundary shift caused by the experts' corrections), the number of positive and negative deltas and the standard deviation of the deltas are calculated for each boundary type. The following are the calculated values and metrics in more detail:

Table 10. Metrics used in evaluating segmentation (in red).

Value or Metric	Symbol	Formula
Tolerance	T	-
Number of boundaries of type B	N_B	-
Number of boundaries of type B Within Tolerance T	$N_{T,B}$	-
Precision	$P_{T,B}$	$\frac{N_{T,B}}{N_B} * 100$
Predicted time stamp of boundary b	$tp(b)$	-
Expert corrected time stamp of boundary b	$tc(b)$	-
Delta	$D(b)$	$tp(d) - tc(d)$
Number of positive deltas for boundaries of type B	D_B^+	-
Number of negative Deltas for boundaries of type B	D_B^-	-
Average Delta for boundaries of type B	D_B^*	$\sum_{b \in B} \frac{\mathrm{D(b)}}{\mathrm{N}_B}$
Standard deviation of Delta for boundaries of type B	D_B^σ	$\sqrt{\frac{\sum_{b \in B}(\mathrm{D(b)} - D_B^*)^2}{D_B^\#}}$

Metrics in red (see Table 10) are the metrics used to assess segmentations. They were used to find which types of boundaries were most incorrectly predicted by the system or identify misunderstandings between the experts. Symbols in column 2 are sometimes used in the rest of this work for convenience. As stated above, the $P_{T,B}$ metric is not novel and is the metric used in the literature to evaluate segmentation precision. The five other metrics ($\mathrm{D}_B^\#$, D_B^+, D_B^-, D_B^*, D_B^σ) are novel, and as future work, could also be used for boundary refinement (they are called shift metrics in this work).

Table 11. Insertion, deletion and update metrics.

Value or Metric	**Symbol**
Number of boundaries added	B^{+}
Number of boundaries deleted	B^{-}
Number of phone labels changed	L^{c}

Table 11 shows three other metrics for assessing expert performance in alignment. Even though the textual transcript was corrected before generating the phonetic transcript and aligning, the experts were not only required to correct boundary locations but also add missing boundaries, remove unnecessary ones and correct phone labels that do not match the speech.

There are several reasons why there could still be errors in the phonetic transcript after manual revision:

1- Experts did not detect an error in the first stage of text correction or in second stage when matching text with recorded speech.

2- The phonetic transcript generated automatically could contain errors as some parts of the algorithm are non-deterministic and generate multiple possible pronunciations of the same word. HTK's forced alignment would need then to choose the best possible pronunciation for each word and this did not result in matching phonetic sequences in all cases. This was sometimes due to the talent pronouncing letters with some imperfections which makes the identity of the phone disputed. An example of these imperfections would be emphasising a vowel in a non-emphatic context or visa-versa.

3- Some words were not pronounced according to the rules found in this work for automatic phonetic transcription (see Section 3.1). This includes foreign words that are written in Arabic but pronounced using phones that may not be part of the Arabic's (or MSA's) phonetic vocabulary (see Table 6. *Final Phoneme set (82 in total). Note that geminated consonants are not included in the table for simplicity purposes*.). The pronunciations of these words had to be entered manually by the experts.

4- The experts were given the option to add a distortion label (see Table 6) to segments where pronunciations were not clear.

It is important to note that all the causes listed above were found after correcting the first three batches of utterances (150 utterances with 50 each). Any systematic errors in the transcript were attributed to either a flaw in the algorithm generating the phonetic transcript (see Section 3.1), or to irregular pronunciation caused by a mistake by the talent or the nature of the word (foreign nouns). The former type of errors was dealt with by modifying the algorithm and rerunning the alignment. Two issues were found in the phonetic transcript after the first corrections phase:

1- Geminated consonant letter “ي” is pronounced inconsistently depending on context. If preceded by the diacritic ”◌ِ”, the generated phonetic transcript is /ii0 y/. Otherwise - if preceded by a “◌ُ” or a “◌ِ”- it is transcribed as /u0 yy/ and /i0 yy/ accordingly. The reason for this issue is that no previous formalisation of the phonetic transcription of geminated “ي” was found and it was assumed that the effect of geminating a consonant “ي” would always result in a separate consonant phone /yy/ but this was seen not to be the case in practice and in the context explained above, geminated “ي” is pronounced as a combination of a vowel followed by a consonant.

2- Similar to the first issue, geminated consonant letter “و” is pronounced inconsistently based on the preceding short vowel. The only difference is that a preceding “◌ُ” would cause the transcription to be a long vowel /uu0/ followed by a non-geminated /w/.

Table (see Table 12) shows the number of inserted, deleted and altered tags in the three batches used to evaluate the first stage of (flat start) forced alignment.

4.2 Boundary Types

Feedback from experts indicated that correcting certain boundary types was more difficult than others because of strong co-articulation between phones. This led to the idea of categorising boundary types based on the type of articulation of surrounding phones (fricative, stop, trill…). For example, the boundary between the phones /q/ and /l/ is labelled a “stop/approximate” boundary or “st/ap” boundary or more specifically a “voiceless-stop/approximate” or “vl-st/ap” boundary (the latter being a subset of the former). For stops and fricatives, both the voiced and voiceless subsets were included in the analysis. This means that the boundary types are not disjoint sets and some sets are subsets of others (the above being an example). Vowels were all grouped together under the same articulation category, “vowels” or “vo”.

The precision and shift metrics were calculated for each boundary type to show how accurately the forced alignment works for each type and the nature of shift happening in each type. This was inspired by feedback from the experts who found systematic shifts in the boundaries between the predicted and corrected timestamps. Also, it is already established that some boundary types, when realised in speech, correspond to abrupt changes in the acoustic features (intensity and spectrum) and hence could potentially be easier to detect by a machine (Yi 2003; Hosom 2009).

Boundary types used in this work are shown in the results available through the web link (Halabi 2015).

Next, the forced alignment parameters and HMM topology used in the evaluation are presented to make is easier to compare with other works.

4.3 HTK Parameters

HTK allows changing several parameters before running each of its components. It is not claimed here that all the choices for each of these parameters are optimal as, for some of these parameters, there haven't been experiments conducted showing performance for different values. Most of the chosen values for the parameters were based on the HTK segmentation scheme (Young et al. 1997). The parameters are the following:

1- Acoustic Features (MFCC, LPC…): MFCC were chosen with 36 features for each window. HTK allows changing more specific parameters when it comes to MFCC feature extraction such as the number of filterbanks. All the values for these parameters were set according to the HTK segmentation scheme (Young et al. 1997).

2- Pre-emphasis Parameter: Determines the extent to which the certain frequencies are boosted in the speech signal to decrease the effect of noise (Mporas et al. 2009; Young et al. 1997). The value used was 0.97 which is the one used in the HTK segmentation scheme (Young et al. 1997).

3- Hamming Window: A true or false value indicating whether to use a hamming window. This was set to true, which was based on the literature where Hamming windows were almost always used (Yuan et al. 2013; Young et al. 1997; Prahallad 2010; Mporas et al. 2009). Hamming windows is a window function which is zero or a very low value outside a certain range used to extract parts of a speech signal for analysis.

4- Window Size: This is the length of the Hamming window used. This determines how long the segments of speech used for MFCC coefficient extraction are (see figure). The default value in the HTK segmentation scheme (Young et al. 1997) was used.

5- Energy Normalisation: A true or false value indicating whether to normalise the log-energy (log-intensity) of the speech signal before extracting features. It was set to true which was the default value in the HTK segmentation scheme (Young et al. 1997).

6- Topology: The HMM models used had 3 states (in addition to dummy start and end states) which is the most common in the literature. Emission probabilities were modelled as a single Gaussian in 36 dimensions for each of the MFCC coefficients. The transition probabilities between states are multinomial. As future work, it is intended to use more Gaussians for the emission probabilities. There are many possibilities here to adjust the number of mixtures and the number of channels to which each of the MFCC coefficients belong.

7- Window Shift Rate: Called TARGETRATE in HTK. It is the shift applied to the window after each calculation of the MFCC coefficients. The default value in the HTK segmentation scheme (Young et al. 1997) was used.

4.4 Initial Evaluation (Flat Start)

4.4.1 Alignment quality:

Table 13 shows the precision values of all the metrics of the initial 3 batches of alignment. These contained 13166 boundaries (including boundaries with pauses) and 11311 phone boundaries (excluding boundaries with pauses). 1047 boundaries were skipped by the evaluation script out of the complete 13166 boundary set because of phone label mismatch between the automatically generated phonetic transcriptions and the experts' corrections. This decision was made because boundaries corresponding to incorrect phonetic transcript affect the precision of alignment and would skew the results as the aligner would try to align script to a non-matching speech signal. The goal of this evaluation is to calculate the precision of the alignment knowing that a certain percentage of phone boundaries and labels were mismatching (7.9 % of boundaries in this case). 12119 boundaries were left for analysis as shown in table (see Table 13). 68.49% of the predicated boundaries were within 20 milliseconds of the corrected boundaries. This is significantly lower than the precision achieved on the TIMIT corpus in previous work (Hosom 2009). The difference being that in this work, a different HMM topology was used, and the phonetic transcription was automatically generated by a rule based algorithm rather than depending on a human-generated pronunciation dictionary. This generated errors that affect the precision of the HMM forced alignment system.

To increase this precision further analysis of the common errors detected by the experts and retraining of the system based on the manually aligned subset is done (see Section 4.5).

Table 12 shows the number of insertions, deletions and updates the experts have done. The "Mismatching Boundaries" row does not simply equal the sum of the insertions, deletions and updates, because each one of these could either cause one or two mismatching boundaries. Overall, less than 8% of the boundaries were mismatching between the correction and automatically generated transcript and mostly due to recording errors or foreign words. This could be improved by adding a foreign-word pronunciation dictionary which does not exist for MSA.

Table 12. Correction Statistics for three batches

B^{+}	133 (~1.0%)
B^{-}	134 (~1.0%)
L^{c}	534 (~4.0%)
Mismatching boundaries	1047 (~7.9%)

There are no previous works on transcript corrections with which to compare these numbers. But it is important to note that there have been speech synthesis voices built on uncorrected automatically generated and aligned transcript in the past. In this work we attempt to find whether an uncorrected portion of the corpus, aligned by a system trained on a corrected portion of the corpus, would be suitable for speech synthesis in MSA using a listening test.

The rest of the results are available through Halabi (2015). They show the precision for different boundary types. It is easy to see that some boundary types correspond to significantly higher precision than others.

Table 13. Precision of Initial forced alignment for general boundary types. Blue shows this work's system. Red is the TIMIT Result by (Hosom 2009). "ph" stands for "phone", "pa" stands for "pause", "co" stands for "consonant" and "vo" stands for "vowel".

T	**<0.005**	**<0.010**	**<0.015**	**<0.020**	**<0.025**	**<0.030**	**>0.050**	D_B^*	N_B	D_B^+	D_B^-	D_B^σ
ph/ph	33.42	45.26	57.67	**68.49**	76.93	83.1	100	-0.00741	11311	2059	6534	0.002695
vo/co	28.48	38.22	50.41	63.01	73.48	80.87	100	-0.01181	4955	580	3325	0.002874
co/vo	37.66	52.06	64.91	74.21	80.69	85.89	100	-0.00231	5075	1277	2480	0.002782
co/co	35.63	45.42	57.01	67.03	75.25	80.58	100	-0.01063	1277	202	727	0.001472
Silence Boundaries												
pa/ph	28.07	29.82	32.89	40.79	57.46	75.44	100	0.002481	228	16	154	0.028729
ph/pa	22.22	37.2	46.38	57.97	67.63	76.33	100	-0.00188	207	41	129	0.015074
pa/co	28.07	29.82	32.89	40.79	57.46	75.44	100	0.002481	228	16	154	0.028729
co/pa	21.05	46.05	60.53	69.74	76.32	80.26	100	-0.00815	76	14	48	0.00044
vo/pa	22.9	32.06	38.17	51.15	62.6	74.05	100	0.001762	131	27	81	0.023527
Reported TIMIT precision (Hosom 2009)	48.42	79.30	89.49	**93.36**	95.38	96.74	100	-				

4.4.2 Expert Agreement:

Because the alignment process takes a long time, batches of 50 utterances were distributed to two experts (each expert receiving a different batch of utterances). A third expert later checked their work and corrected any errors left. The two experts at the beginning were trained together to make sure that their alignments were as similar as possible but it is useful to know how close their alignments were. This is referred to in the literature as expert agreement or inter-annotator agreement (Hosom 2009; Romportl 2010). We gave each expert 5 additional utterances that were part of the other expert's workload giving a total of 10 utterances aligned by both experts to conduct an expert agreement test.

To show how similar the alignments were between the experts. The same metrics were used as in the precision evaluation of the alignment – as found in the literature (Hosom 2009; Romportl 2010). The only difference is that the number of changes in phone labels was calculated. This number is the sum of the number of labels changed, the number segments added by the experts and the number of segments removed by the experts. If the resulting phone label sequence does not match, the analysis script skips the boundaries and does not included them in the agreement analysis shown in *Table 14*.

In this test, both of the experts had to correct the predicted boundaries resulting from forced alignment rather than correcting each other's. This is to estimate the agreement more accurately. Because if experts were given each other's alignments, it might be tempting not to change boundaries if the error is too small (smaller to that found in the forced alignment output).

Table 14 shows the results of comparing the alignment of 10 utterances between two of the experts. The 10 utterances contained a total of 981 phone boundaries (including ones with a pause) of which 47 had changes in identity (phone label) applied to their adjacent phones by either or both experts which lead to non-matching boundaries which were excluded from the analysis even if accurate. One of the experts inserted 7 new segments that they thought were missing which the other did not and the other expert inserted 7 segments which the first expert did not include. This resulted in the system ignoring 97 boundaries when calculating precision. 884 overall phone boundaries were left for agreement analysis. 827 of those boundaries were boundaries between two phones (no pause) and the remaining had at least one adjacent pause. Note that two consecutive pauses are possible.

84.28% of all boundaries were within 20 milliseconds of each other. As mentioned earlier, the 20 millisecond tolerance is the de facto standard as found in the literature for evaluating alignment precision (Hosom 2009; Stolcke et al. 2014; Yuan et al. 2013) but also it is the standard for evaluating expert agreement. The highest precision in previous work for expert agreement was on the TIMIT corpus with 93.49% of boundaries generated by the author within 20 milliseconds of corresponding boundaries in the TIMIT corpus (Hosom 2009). In the same work, Hosom reviews previous work which shows results in expert agreement. All the reviewed attempts reported agreement of over 90% which poses the question: why is the agreement in this work lower? Hosom excluded two types of boundaries from their evaluation because they proposed that they were subjective and shouldn't be in the precision analysis. No boundaries were excluded in this work. But still this leaves a significant difference in agreement which lead to a third expert running through the two experts' alignments (specially the points of disagreement) and normalising the alignment. This is not as laborious a task compared to the initial alignments as it only requires that the expert to review about 10% to 20% of the corpus. This mainly occurred at the boundaries that were not included in the analysis due to experts disagreeing in the segment's label or boundaries of a type corresponding to a lower precision score.

This stage helped identify misunderstandings in the labelling, segmentation and alignment processes by the experts, which they were informed about for more agreement in future manual alignment and corrections.

Table 14. Expert Agreement Analysis Results.

T	**<0.005**	**<0.010**	**<0.015**	**<0.020**	**<0.025**	**<0.030**	**>0.050**	D_B^*	N_B	D_B^+	D_B^-	D_B^σ
ph/ph	42.63	59.81	73.93	84.53	90.86	95.01	100	0.010362	821	376	258	0.047647
vo/co	42.35	59.29	74.04	84.15	91.53	95.63	100	0.00566	366	199	84	0.000206
co/vo	43.24	61.54	76.13	86.74	93.1	96.82	100	9.33E-05	377	148	147	0.000176
co/co	41.56	54.55	63.64	76.62	77.92	84.42	100	0.001998	77	28	27	0.000349
vo/vo	0	0	0	0	0	0	0	0	0	0	0	0
Silence Boundaries												
pa/ph	18.75	31.25	37.5	37.5	50	68.75	100	0.019574	16	13	1	0.000204
ph/pa	7.14	14.29	42.86	57.14	64.29	64.29	100	-0.01578	14	2	11	0.000728
pa/co	13.33	26.67	33.33	33.33	46.67	66.67	100	0.020879	15	13	1	0.000191
pa/vo	0	0	0	0	0	0	0	0	0	0	0	0
co/pa	16.67	16.67	50	50	50	50	100	-0.01436	6	1	4	0.001432
vo/pa	0	12.5	37.5	62.5	75	75	100	-0.01685	8	1	7	0.000197
TIMIT Agreement Results	60.38	81.73	89.07	**93.49**	95.36	96.91	100	-				

4.5 HTK Bootstrapping

At each iteration and after the correction of 150 utterances was completed with a second revision, another automatic segmentation was conducted with the manually corrected data as input to bootstrap HMM models. HInit was used to initialise the parameters of the HMM model/s used for the different phones. HInit is an HTK tool that initialises the phone HMM parameters by using manual segmentations. For each phone, all the available segments for that phone in the training data were loaded and used to iteratively update the parameters of the phone's initial HMM using Viterbi training (Jurafsky & Martin 2008). Viterbi training works in a slightly different way than the Baum-Welch algorithm described in (see Section 3.3). In it, each of the phone's segments are divided equally between the states of the phone's HMM then these divisions are used to calculated each of the HMM's states' parameters. The new HMM model with the new parameters was utilised by the Viterbi algorithm to find the most likely sequence of states (under the new model) and the operation is repeated until convergence. These parameters include the means and variances of the Gaussians whose mix makes up the observation probability distributions and the transition matrixes which define the transition probabilities between states.

Then the training process continued in the same way as in stage 1 using the parameters estimated from HInit as a starting point instead of the output of HCompV. Note that HCompV was still run in this stage as it is required to produce the variance floors and HRest is iteratively run (Baum-Welch).

Table 15 shows the improved results after bootstrapping. A significant improvement from 68% to 82% is achieved. It is important to note that the results in *Table 15* are based on 50 utterances which is about one third of the amount used for the flat start evaluation. This included 3320 phone/phone boundaries. For the complete set of results please refer to the link (Halabi 2015).

Table 15. Alignment results after bootstrapping.

T	<0.005	<0.010	<0.015	<0.020	<0.025	<0.030	>0.050	D_B^*	N_B	D_B^+	D_B^-	D_B^σ
ph/ph	32.77	56.14	71.57	82.5	88.73	92.8	100	-0.00521	3320	961	1921	0.000267
vo/co	30.25	50.76	67.33	80.52	86.95	91.78	100	-0.00862	1448	293	948	0.000266
co/vo	35.78	62.35	77.1	86.32	91.52	94.59	100	-0.00146	1498	562	770	0.000233
co/co	30.38	52.15	65.59	74.73	84.41	89.52	100	-0.00706	372	105	203	0.000303
vo/vo	0	0	0	0	0	0	0	0	0	0	0	0
Silence Boundaries												
pa/ph	27.47	57.14	72.53	84.62	91.21	92.31	100	-0.00456	91	37	49	0.000337
ph/pa	15.29	41.18	51.76	67.06	72.94	77.65	100	-0.00862	85	27	55	0.001062
pa/co	27.47	57.14	72.53	84.62	91.21	92.31	100	-0.00456	91	37	49	0.000337
pa/vo	0	0	0	0	0	0	0	0	0	0	0	0
co/pa	13.33	40	46.67	70	73.33	80	100	-0.01547	30	7	22	0.001185
vo/pa	16.36	41.82	54.55	65.45	72.73	76.36	100	-0.00488	55	20	33	0.000955
TIMIT Agreement Results	48.42	79.30	89.49	**93.36**	95.38	96.74	100	-				

4.6 Precision Comparison

The next stage involves showing the precision of the system in this work relative to the highest precision systems in other works in the literature. As shown in *Table 16*, 93.36 P_{20} is the highest found in literature (Hosom 2009). The system used HMM/ANN (Hidden Markov Models paired with Neural Networks for feature extraction) as a baseline system to compare it with their modified HMM/ANN system which achieved the higher precision mentioned above by adding features on top of the MFCC feature set used in their work as well. These features included energy and burst detection features to help give areas of rapid acoustic feature changes more chance of being detected as boundaries.

Hosom 2009 also trained their system on part of the TIMIT corpus and did not perform any forced alignment. They claimed that regular HMM forced alignment system (similar to the one in this work) did not perform as well as theirs. They used two fifths of the dataset for evaluation and three fifths for training.

It is easy to see from *Table 16* that using an HMM/ANN system would improve the precision. It is also possible to infer this from the improvement HMM/ANN systems give to speech recognition relative to pure HMM (Hosom 2009). It was not possible to obtain or implement a version of it for this work but is suggested for use in future work (see Chapter 5:). Improving either the HMM forced alignment or the HMM/ANN system is not part of this work, it is used to demonstrate the correctness of the phone set and pronunciation rules produced by the automatic phonetic transcript generation system and the overall quality of the corpus.

As shown in *Table 16*, the system proposed in this work approaches the state of the art HMM forced alignment systems but still lags behind HMM/ANN. The difference in the evaluation setup of each system is detailed in this table.

Table 16. Precision comparison

Metric	**Basic HMM forced alignment on MSA**	**Basic HMM forced alignment on MSA (with bootstrapping)**	**Baseline HMM/ANN forced alignment on TIMIT (Hosom 2009)**	**(Hosom 2009) Proposed System**
Feature used	MFCC (basic HTK setting) (Young et al. 1997)	MFCC (basic HTK setting) (Young et al. 1997)	MFCC with mel scale replaced by Bark scale	MFCC with mel scale replaced by Bark scale. With additional energy-based features
Model Architecture	1 Gaussian to model emition probabilities. Basic 3-state HMM architecture. (Young et al. 1997)	1 Gaussian to model emition probabilities. Basic 3-state HMM architecture. (Young et al. 1997)	HMM with ANN instead of Mixture of Gaussians	HMM with ANN instead of Mixture of Gaussians. With modifications on the state structure of the HMM.
Dataset used	Recorded as part of this work	Recorded as part of this work	TIMIT	TIMIT
Training data size	Unsupervised	150 utterances Approximately 25 minutes of speech	3696 files (3.145 hours of speech)	3696 files (3.145 hours of speech)
Evaluation data size	150 utterances Approximately 25 minutes of speech	50 utterances Approximately 6 minutes of speech	1344 “si” and “sx” file from TIMIT corpus	1344 “si” and “sx” file from TIMIT corpus
Language	MSA	MSA	English	English
Precision (P_{20})	68.49	**82.50**	91.48	**93.36**

Chapter 5: Conclusions and Future Work

A challenge remains to evaluate the contributions in this work. So far, the contributions could be listed as follows:

1- An MSA speech corpus for speech synthesis.
2- An MSA phoneme set for MSA spoken in a Levantine dialect.
3- An MSA Phonotactic Rule-set for converting MSA text to a phoneme sequence in Levantine accent.

Mainly, the contributions given in this work so far are in formalising MSA phonology for MSA speech corpus design and Arabic speech synthesis. The claim here is that this formalisation should help both in speeding up the corpus design and alignment process by:

1- Automatically phonetising MSA transcripts: The correctness of the automatic phonetiser produced in this work can only be verified by experts. This is because the pronunciation of MSA depends on the speaker's dialect. The iterative process described in this work for verifying the phonotactical rule-set was done by experts who are native speakers and teachers of Arabic language.
2- Using the phonetised script to choose the optimal subset of the script based on phonetic coverage for recording. We have shown the distribution of diphones and phonemes in the chosen transcript after reducing it to an optimal subset. As no work has been done before on corpus design in Arabic, a comparison to English corpus design was undertaken. A future work would be to attempt other optimisation methods and compare the results in Arabic.

Another claim is that the formalisation of MSA phonology would help in creating better front-ends for TTS systems. This is to be conducted in the future in this work. By building a speech synthesiser using this corpus and the phonetiser generated from this work, a standardised listening test will be conducted to evaluate the corpus in naturalness and intelligibility. It is also useful to add other metrics to differentiate between the quality issues or benefits resulting from recording, transcript, segmentation or alignment; or from the speech engine itself.

In short, the next stage of this work will include finding the methods and metrics to evaluate the quality of a TTS system built using the corpus, phonotactic rule-set and phoneme set produced in this work, and then conducting the evaluation based on these methods and metrics. If possible, a comparative study with previous attempts in MSA or other languages will also be conducted.

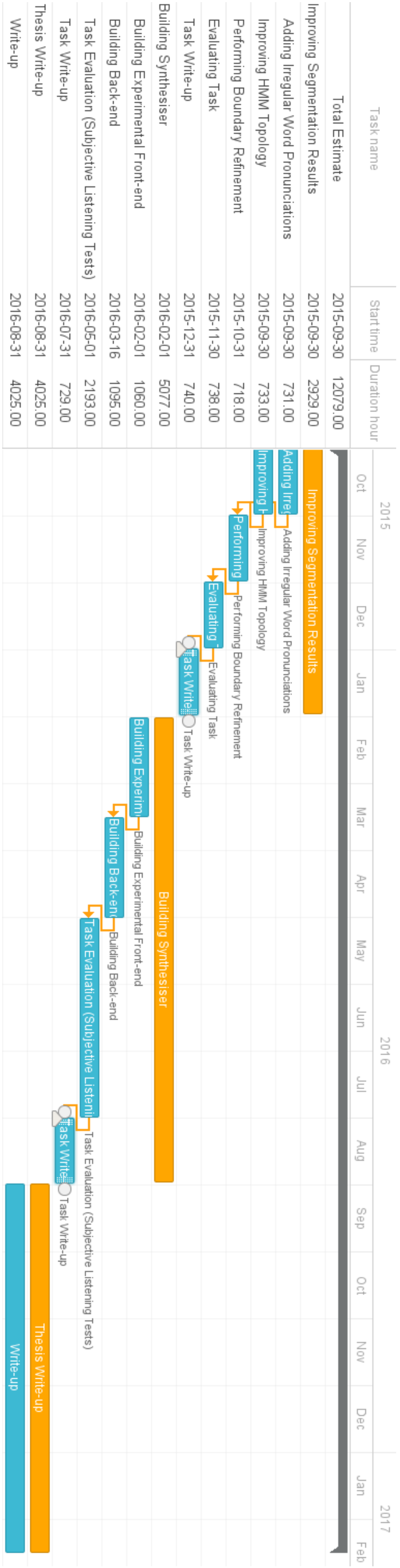
As shown in *Figure 4*, the next stage of this PhD work involves two main phases:

1- Improving Segmentation and Alignment results: The 82.5% precision within T_{20} is still significantly lower than the reviewed works. In this work, it is intended to try more complicated HMM topologies and employ the results from the different boundary types to perform boundary refinement.

   Segmentation and alignment quality is affected by mismatching annotations caused by foreign words. Foreign words imported into MSA script also require separate treatment when phonetising MSA script. This will be solved by creating a pronunciation dictionary for foreign words. If a foreign word is encountered which is not included in this dictionary, building a separate foreign word phonetiser would be a possibility.

2- Building Synthesiser: To evaluate the quality of the generated speech corpus, it is intended to build a speech synthesiser to conduct subjective listening tests. There are previously designed tests for English so a challenge here would be to map those to Arabic to be able to compare this work's corpus to others.

   It is important to note here that there are no intentions to produce contributions to the frond-end or back-end of the synthesiser. This stage is merely for evaluating the speech corpus.



Task name	Start time	Duration hour
Total Estimate	2015-09-30	12079.00
Improving Segmentation Results	2015-09-30	2929.00
Adding Irregular Word Pronunciations	2015-09-30	731.00
Improving HMM Topology	2015-09-30	733.00
Performing Boundary Refinement	2015-10-31	718.00
Evaluating Task	2015-11-30	738.00
Task Write-up	2015-12-31	740.00
Building Synthesiser	2016-02-01	5077.00
Building Experimental Front-end	2016-02-01	1060.00
Building Back-end	2016-03-16	1095.00
Task Evaluation (Subjective Listening Tests)	2016-05-01	2193.00
Task Write-up	2016-07-31	729.00
Thesis Write-up	2016-08-31	4025.00
Write-up	2016-08-31	4025.00

Figure 4. Future work's Gantt chart

Appendices

Appendix A Acoustic Features

All speech recognition and segmentation systems do not perform inference directly on the speech frames. There is always a layer that transforms the raw speech data to a sequence of feature vectors that are calculated from a window with a certain width and shifted by a certain amount to calculate the next frame. The window size and shift (offset) are always measured in milliseconds and typical they are 20-25 ms and 5-10 ms accordingly (Young et al. 1997).

The majority of the methods reviewed use mel frequency cepstral coefficients (MFCC) as acoustic features and often in combination with other features. These are extracted from the acoustic signal before any training or inference is done.

MFCC (Jurafsky et al.,2008) are a set of coefficients that have been found to have strong correlation with the vocal tract physical state of a human being and from this physical state it is possible to determine which phone is being uttered. This justifies the choice of MFCC as features because it enables the classification of phones from the vocal tract's physical state to then be classified from a correlated parameter that is MFCC.

MFCC are a representation of the speech signal that tries to ignore the unwanted information about the speech signal like speaker identity and the loudness of speech. In tasks like speech recognition we are not interested to know if the speaker is a male or a female (unless we are performing speaker recognition) or how loud they are speaking rather we want to know which phone they are uttering, so two speakers with different sound characteristics and possibly gender should generate similar MFCC when uttering the same phones.

There were attempts to improve MFCC precision in speech segmentation by using it alongside other features. (Hosom, 2009) tried adding additional features related to bursts or sudden increase in loudness and intensity of speech which could indicate occurrences of events such as phone boundaries. Even though MFCC does have loudness and intensity change detection characteristics, they have argued that their additions make the system more sensitive to those changes. The claimed that these feature additions have improved boundary detection for most boundary types.

Perceptual Linear prediction (PLP) (Hermansky, 1990) shares similarities with MFCC. It is also inspired by the human auditory system. The main difference in PLP (Honig et al., 2005) is that it performs Linear Predictive Coding (LPC) to the pre-emphasised Bark scale transformed power spectrum which generates an approximation of this spectrum. All this is before moving to the cepstral coefficients similar to MFCC. From the literature, no claim has been found that this linear coding step simulates any stage of hearing in the auditory system and it appears to be just a dimension reduction method.

Other feature extraction techniques have been encountered such as Discrete Cosine Transform Coefficients (DCTC) and LPC coefficients. The former is strongly related to MFCC and PLP as it estimates the cepstrum of the speech. The latter is strongly related to the human vocal tract state. (Karnjanadecha et al., 2012) conducted a comparison between different types MFCC, PLP and DCTC and showed that MFCC is best when using 39 cepstral and energy coefficients but they also showed that DCTC with 78 cepstral and energy coefficients out performed all the other methods but did not test for more coefficients for MFCC and PLP.

Appendix B Arabic Phonology

Arabic phonology is made up of 28 consonants each of which could be geminated. A geminated consonant was considered a separate phoneme in this work (see *Table 17*).

Gemination is linguistically defined as the doubling of a consonant. Phonetically, it usually involves lengthening part of the consonant making it approximately twice as long as the original. When describing syllables using “c” and “v” characters, geminated consonants are either symbolised as “C” or “cc”, the former is used in this work. The latter symbolisation is used to show that a geminated consonant’s parts belong to different syllables unless it occurs at the end of speech (de Jong & Zawaydeh 1999).

4 extra consonants where added to the table because they occurred in the corpus in foreign words but were not included in optimisation. Overall there are 56 consonant phonemes included in this work for optimisation.

7 of the Arabic consonants are considered strictly “emphatic” and the same goes for their gemination. 2 other consonants and their gemination are considered optionally emphatic (depending on context).

Emphasis propagates from emphasised consonants to adjacent vowels and optionally emphatic consonants causing them to become emphatic. 2 of the 7 emphatic consonants (/x/ and /g/) are only forward emphatic (emphasis from them only propagates to following vowels).

Table 17. Arabic consonant phonemes. IPA representation is enclosed in brackets if phoneme is found in foregin words only. Orange coloured consonants are strictly emphatic. Blue coloured consonants are optionally emphatic.

		Labial	Libio-dental	Emphatic		Plain		Palato-alveolar	Palatal	Velar	Uvular	Pharyngeal/ Epiglottal	Glottal
				Dental	Alveolar	Alveolar	Dental						
Nasal (Always voiced)		m م				n ن							
Stop	voiceless	(p) پ		tˤ ط		t ة ت				k ك	q ق		ʔ ء أ إ ؤ ئ
	voiced	b ب		dˤ ض		d د				g ج			
Affricate									d͡ʒ ج				
Fricative	voiced		(v) ڤ	ðˤ~zˤ ظ		z ز	ð ذ	ʒ ج		ɣ~ʁ غ		ʕ~ʢ ع	
	voiceless		f ف		sˤ ص	s س	θ ث	ʃ ش		x~χ خ		ħ~ʜ ح	h هـ
Approximant		wو			ɫ~l ل				j ي				
Trill					rˤ~r ر								

In this work, 10 vowel phonemes in Arabic are used (see *Table 18*). There are two diphthongs which are considered to be a combination of a vowel and a consonant rather than a separate phoneme and were not included in the table.

Table 18. Arabic vowel phonemes

Vowel	/a/	/A/	/a:/	/A:/	/u/	/u:/	/i/	/i:/	/u1/	/i1/
Arabic Script	◌َ Possibly Pharyngealized		ا ى Possibly Pharyngealized		◌ُ	و	◌ِ	ي	◌ُ	◌ِ

References

Alghmadi, M.M., 2003. KACST Arabic Phonetics Database. In *The Fifteenth International Congress of Phonetics Science, Barcelona*. Barcelona, Spain, pp. 3109–3112. Available at: http://www.researchgate.net/publication/229049878_KACST_Arabic_Phonetics_Database [Accessed November 3, 2014].

Ali, H.K. & Ali, H.S., 2011. Epenthesis in English and Arabic A Contrastive Study. *Journal of Tikrit University for the Humanities*, 18(6), pp.648–660.

Aljazeera, 2015. Aljazeera Learn. Available at: http://learning.aljazeera.net/arabic [Accessed February 15, 2015].

Almeman, K., Lee, M. & Almiman, A.A., 2013. Multi dialect Arabic speech parallel corpora. In *2013 1st International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*. IEEE, pp. 1–6. Available at: http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6487288 [Accessed November 5, 2014].

Van Bael, C. et al., 2007. Automatic phonetic transcription of large speech corpora. *Computer Speech & Language*, 21(4), pp.652–668. Available at: http://www.sciencedirect.com/science/article/pii/S0885230807000228 [Accessed May 30, 2014].

Barros, M. & Möbius, B., 2011. *Human Language Technology. Challenges for Computer Science and Linguistics* Z. Vetulani, ed., Berlin, Heidelberg: Springer Berlin Heidelberg. Available at: http://www.springerlink.com/index/10.1007/978-3-642-20095-3 [Accessed January 1, 2015].

Biadsy, F. & Hirschberg, J.B., 2009. Using Prosody and Phonotactics in Arabic Dialect Identification. Available at: http://academiccommons.columbia.edu/catalog/ac:159968 [Accessed January 20, 2015].

Black, A.W., 2002. Perfect Synthesis For All Of The People All Of The Time. In *IEEE 2002 Workshop on Speech Synthesis*. Available at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.19.8631 [Accessed June 6, 2014].

Black, P.E., 2005. Greedy algorithms. Available at: http://www.nist.gov/dads/HTML/greedyalgo.html [Accessed March 15, 2015].

Boersma, P. & Weenink, D., 2015. Praat Software. Available at: http://www.praat.org/ [Accessed August 10, 2015].

Bonafonte, A. et al., 2008. Corpus and Voices for Catalan Speech Synthesis. In *LREC 2008*. Available at: http://aclweb.org/anthology/L08-1517.

Braunschweiler, N., 2006. The Prosodizer - Automatic Prosodic Annotations of Speech Synthesis Databases. In *Proc. Speech Prosody*. pp. PS5–27–76.

Brognaux, S. et al., 2012. Train & align: A new online tool for automatic phonetic alignment. In *Spoken Language Technology Workshop (SLT), 2012 IEEE*. pp. 416–421.

Buckwalter, T., 2002. Buckwalter Arabic Transliteration. Available at: http://www.qamus.org/transliteration.htm [Accessed June 27, 2013].

D. R. Van Niekerk, E.B., 2009. Phonetic alignement for speech synthesis in under-resourced languages. In *INTERSPEECH*. Brighton: ISCA, pp. 880–883. Available at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.384.1749 [Accessed September 10, 2014].

Elshafei, M.A., 1991. Toward an Arabic Text-to-speech System. *The Arabian Journal for Science and Engineering*, 16(4b), pp.565–583.

François, H. & Boëffard, O., 2002. The Greedy Algorithm and its Application to the Construction of a Continuous Speech Database. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*. European Language Resources Association (ELRA). Available at: http://aclweb.org/anthology/L02-1265.

Gadoua, A.H., 2000. Consonant Clusters In Quranic Arabic. *Cahiers Linguistiques d'Ottawa/Ottawa Papers in Linguistics*, 28, pp.59–85.

Ghahramani, Z., 2001. An introduction to hidden Markov models and Bayesian networks. , pp.9–42. Available at: http://dl.acm.org/citation.cfm?id=505741.505743 [Accessed August 12, 2015].

Gorman, K., Howell, J. & Wagner, M., 2011. Prosodylab-aligner: A tool for forced alignment of laboratory speech. *Canadian Acoustics*, 39(3), pp.192–193. Available at: http://jcaa.caa-aca.ca/index.php/jcaa/article/view/2476 [Accessed December 7, 2014].

Habash, N.Y., 2010. Introduction to Arabic Natural Language Processing. *Synthesis Lectures on Human Language Technologies*, 3(1), pp.1–187. Available at: http://www.morganclaypool.com/doi/abs/10.2200/S00277ED1V01Y201008HLT010 [Accessed January 13, 2015].

Halabi, N., 2015. MPhil Code and Results. Available at: https://www.dropbox.com/s/0tjdj4coy1n094g/MPHIL FILES.rar?dl=0.

Halpern, J., 2009. Word Stress and Vowel Neutralization in Modern Standard Arabic. In *2nd International Conference on Arabic Language Resources and Tools*. Cairo, Egypt, pp. 1–7.

Hoffmann, S. & Pfister, B., 2010. Fully automatic segmentation for prosodic speech corpora. In *INTERSPEECH*. Makuhari, Chiba, Japan, pp. 1389 – 1392.

Hosom, J.-P., 2009. Speaker-Independent Phoneme Alignment Using Transition-Dependent States. *Speech communication*, 51(4), pp.352–368. Available at: http://dl.acm.org/citation.cfm?id=1507768.1507931 [Accessed September 30, 2013].

Jakovljević, N. et al., 2012. Automatic Phonetic Segmentation for a Speech Corpus of Hebrew. *INFOTEH-JAHORINA*, 11, pp.742–745.

Jarifi, S., Pastor, D. & Rosec, O., 2008. A fusion approach for automatic speech segmentation of large corpora with application to speech synthesis. *Speech Communication*, 50(1), pp.67–80. Available at: http://www.sciencedirect.com/science/article/pii/S0167639307001215 [Accessed August 21, 2014].

John Alderete, S.A.F., 2009. Phonotactic Learning without A Priori Constraints: A Connectionist Analysis of Arabic Cooccurrence Restrictions. In *Proceedings of the 48th annual meeting of the Chicago Linguistics Society*. Chicago, Illinois. Available at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.387.5358 [Accessed February 9, 2015].

De Jong, K. & Zawaydeh, B.A., 1999. Stress, duration, and intonation in Arabic word-level prosody. *Journal of Phonetics*, 27(1), pp.3–22. Available at: http://www.sciencedirect.com/science/article/pii/S0095447098900882 [Accessed January 4, 2015].

Jurafsky, D. & Martin, J.H., 2008. *Speech and Language Processing*,

Kain, E., Miao, Q. & Santen, J.P.H. Van, 2007. Santen, “Spectral control in concatenative speech synthesis. In *6th ISCA Workshop on Speech Synthesis*. Bonn, Germany, pp. 11–16. Available at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.384.9067 [Accessed January 7, 2015].

Kawai, H. et al., 2000. A Design Method of Speech Corpus for Text-To-Speech Synthesis Taking Account of Prosody. In *Sixth International Conference on Spoken Language Processing (ICSLP 2000)*. Beijing, China, pp. 420–425.

Kawanami, H. et al., 2002. Designing speech database with prosodic variety for expressive TTS system. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC’02)*. European Language Resources Association (ELRA), pp. 2039–2042. Available at: http://aclweb.org/anthology/L02-1337.

Kelly, A.C. et al., 2006. Speech Technology for Minority Languages: the Case of Irish (Gaelic). In *INTERSPEECH*. Pittsburgh, Pennsylvania. Available at: http://www.tara.tcd.ie/handle/2262/39404 [Accessed January 1, 2015].

Kim, S., Kim, J. & Hahn, M., 2006. HMM-based Korean speech synthesis system for hand-held devices. *IEEE Transactions on Consumer Electronics*, 52(4), pp.1384–1390. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4050071 [Accessed August 19, 2013].

King, S., 2013. Measuring a decade of progress in Text-to-Speech. *Loquens*, 1(1), p.e006. Available at: http://loquens.revistas.csic.es/index.php/loquens/article/view/6/12 [Accessed November 3, 2014].

Kominek, J. & Black, A.W., 2003. CMU Arctic Databases for Speech Synthesis. Available at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.64.8827 [Accessed August 14, 2013].

Lamere, P. et al., 2003. The CMU SPHINX-4 Speech Recognition System. In *IEEE International Conference on Acoustics Speech and Signal Processing*. Hong Kong. Available at: http://citeseerx.ist.psu.edu/viewdoc/summary;jsessionid=F39A30BB1E9AF47B64D45B2AB0E3C27C?doi=10.1.1.406.8962 [Accessed June 11, 2015].

Laufer, A. & Baer, T., 1988. The emphatic and pharyngeal sounds in Hebrew and in Arabic. *Language and speech*, 31 (Pt 2), pp.181–205. Available at: http://www.ncbi.nlm.nih.gov/pubmed/3256772 [Accessed February 13, 2015].

Lenzo, K.A. & Black, A.W., 2000. Diphone Collection and Synthesis. In *ICASSP*. Beijing, China. Available at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.32.2556 [Accessed August 20, 2013].

Lu, H. et al., 2011. Building HMM based unit-selection speech synthesis system using synthetic speech naturalness evaluation score. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 5352–5355. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5947567 [Accessed August 19, 2013].

Maia, R. et al., 2007. An Excitation Model for HMM-Based Speech Synthesis Based on Residual Modeling. In *6th ISCA Workshop on Speech Synthesis*. Bonn, Germany: International Speech Communication Association, pp. 131–136. Available at: http://library.naist.jp/dspace/handle/10061/8269 [Accessed August 20, 2013].

Malfrère, F. et al., 2003. Phonetic alignment: speech synthesis-based vs. Viterbi-based. *Speech Communication*, 40(4), pp.503–515. Available at: http://www.sciencedirect.com/science/article/pii/S0167639302001310 [Accessed May 29, 2014].

Matoušek, J. & Romportl, J., 2007a. Recording and annotation of speech corpus for Czech unit selection speech synthesis. In *TSD'07 Proceedings of the 10th international conference on Text, speech and dialogue*. Springer-Verlag, pp. 326–333. Available at: http://dl.acm.org/citation.cfm?id=1776334.1776380 [Accessed August 19, 2013].

Matoušek, J. & Romportl, J., 2007b. *Text, Speech and Dialogue* V. Matoušek & P. Mautner, eds., Berlin, Heidelberg: Springer Berlin Heidelberg. Available at: http://www.springerlink.com/index/10.1007/978-3-540-74628-7 [Accessed December 31, 2014].

Matousek, J.R. & Psutka, J., 2001. Design of Speech Corpus for Text-to-Speech Synthesis. In *EUROSPEECH*. Aalborg, Denmark, pp. 2047–2050. Available at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.60.9719 [Accessed January 1, 2015].

Moreno, P.J. et al., 1998. A recursive algorithm for the forced alignment of very long audio segments. In *ICSLP*. ISCA. Available at: http://dblp.uni-trier.de/db/conf/interspeech/icslp1998.html#MorenoJTG98.

Mporas, I. et al., 2009. Using Hybrid HMM-Based Speech Segmentation to Improve Synthetic Speech Quality. In *Informatics, 2009. PCI '09. 13th Panhellenic Conference on*. pp. 118–122.

Murphy, K.P., 2012. *Machine Learning. A Probabilistic Perspective* 1st ed., Cambridge, Massachusetts: MIT Press.

Newman, D., 1986. The Phonetics of arabic. *Journal of the American Oriental Society*, pp.1–6.

Oliveira, L. et al., 2008. Methodologies for Designing and Recording Speech Databases for Corpus Based Synthesis. In *LREC 2008*. Available at: http://aclweb.org/anthology/L08-1484.

Peddinti, V. & Prahallad, K., 2011. Exploiting Phone-Class Specific Landmarks for Refinement of Segment Boundaries in TTS Databases. In *INTERSPEECH*. ISCA, pp. 429–432. Available at: http://dblp.uni-trier.de/db/conf/interspeech/interspeech2011.html#PeddintiP11.

Pereira, F.C.N. & Riley, M.D., 1996. Speech Recognition by Composition of Weighted Finite Automata. , p.24. Available at: http://arxiv.org/abs/cmp-lg/9603001.

Prahallad, K., 2010. Automatic building of synthetic voices from audio books. Available at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.169.7981 [Accessed October 14, 2013].

Qian, Y., Cao, H. & Soong, F.K., 2008. HMM-Based Mixed-Language (Mandarin-English) Speech Synthesis. In *2008 6th International Symposium on Chinese Spoken Language Processing*. IEEE, pp. 1–4. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4730269 [Accessed August 20, 2013].

Romportl, J., 2010. Automatic Prosodic Phrase Annotation in a Corpus for Speech Synthesis. Available at: http://www.kky.zcu.cz/cs/publications/JanRomportl_2010_AutomaticProsodic [Accessed March 10, 2015].

Stolcke, A. et al., 2014. Highly Accurate Phonetic Segmentation Using Boundary Correction Models and System Fusion. In *ICASSP*. Available at: http://research.microsoft.com/apps/pubs/default.aspx?id=209007.

Tao, J. et al., 2008. Design of Speech Corpus for Mandarin Text to Speech. In *The Blizzard Challenge 2008 workshop*. Brisbane, Australia.

Taylor, P., 2009. *Text-To-Speech Synthesis*, Cambridge University Press.

Tench, P., 2015. Consonants. Available at: http://www.cardiff.ac.uk/encap/contactsandpeople/academic/tench/consonants.html [Accessed March 15, 2015].

Thelwall, R. & Sa'Adeddin, M.A., 2009. Arabic. *Journal of the International Phonetic Association*, 20(02), p.37. Available at: http://journals.cambridge.org/abstract_S0025100300004266 [Accessed November 3, 2014].

Umbert, M. et al., 2006. Spanish Synthesis Corpora. In *Proceedings of the International Conference of Language Resources and Evaluation (LREC)*. Genoa, Italy, pp. 2102–2105.

Vetulani, Z., 2011. *Human Language Technology. Challenges for Computer Science and Linguistics*, Available at: http://books.google.com/books?id=l5FZxDY3yi0C.

Watson, J.C.E., 2007. *The Phonology and Morphology of Arabic*, Available at: http://books.google.com/books?hl=de&lr=&id=4RDIoDAF1e8C&pgis=1 [Accessed November 3, 2014].

Yi, J.R.-W., 2003. Corpus-based unit selection for natural-sounding speech synthesis. Available at: http://dspace.mit.edu/handle/1721.1/16944 [Accessed December 30, 2014].

Young, S. et al., 1997. *The HTK book*, Cambridge, Massachusetts: Cambridge University.

Yuan, J. et al., 2013. Automatic phonetic segmentation using boundary models. In F. Bimbot et al., eds. *INTERSPEECH*. ISCA, pp. 2306–2310. Available at: http://dblp.uni-trier.de/db/conf/interspeech/interspeech2013.html#YuanRLSMW13.

Zen, H. et al., 2007. The HMM-based speech synthesis system (HTS) version 2.0. In *6th ISCA Workshop on Speech Synthesis*. Bonn, Germany: International Speech Communication Association, pp. 294–299. Available at: http://www.researchgate.net/publication/228365542_The_HMM-based_speech_synthesis_system_(HTS)_version_2.0 [Accessed August 20, 2013].

Zen, H., Senior, A. & Schuster, M., 2013. Statistical Parametric Speech Synthesis Using Deep Neural Networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. pp. 7962–7966.

Zue, V.W. & Seneff, S., 1996. Transcription and Alignment of the TIMIT Database. In *Recent Research Towards Advanced Man-Machine Interface Through Spoken Language*. Elsevier, pp. 515–525. Available at: http://linkinghub.elsevier.com/retrieve/pii/B9780444816078500888.