Conversational Arabic Automatic Speech Recognition

Sarah Al-Shareef

Department of Computer Science
University of Sheffield
United Kingdom

This dissertation is submitted for the degree of
Doctor of Philosophy

May 2015

Dedication

I would like to dedicate this thesis to my loving parents Ahmad and Haya and my supporting brothers Mansour and Mashhour.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text. This dissertation contains 65,164 words excluding bibliography, footnotes, equations and tables and has 56 figures and 54 tables.

Acknowledgments

"All praise is due to Allah" and "whoever do not thank people do not thank Allah". Undertaking this PhD has been a truly life-changing experience for me and it would not have been possible without the support and guidance that I received from many people.

First and foremost I offer my sincerest gratitude to my supervisor, Professor Thomas Hain. It is an often used cliché, but in this case it is no overstatement to say that without the consistent guidance, support, knowledge, and encouragement of my supervisor, this thesis would never have existed. You went above and beyond to read every line of my draft in meticulous detail. I have learnt more from you than I can possibly put into words.

Secondly, I must say a special thank you to Professor Georg Struth who believed in me and offered me his complete support as a PG Tutor. Additionally, I have been blessed with a friendly and cheerful group who offered and shared many interesting discussions, ideas, thoughts and support, in particular: Amy Beeston, Erfan Loweimi, Chenhao Wu, Raymond Ng, and all former and present members of MINI and SpandH groups. Thank you Mark Summers for taking the time to proof read this thesis, your input was invaluable. Thanks to my examiners, Professor Phil Green and Professor Martin Russell for the valuable comments and an intellectually stimulating and enjoyable viva.

To my brother Mashhour, who have agreed to embark this journey with me, leaving his life back home and starting from scratch out side his comfort zone, just to be there for me. You are everything a sister could have asked for in a brother and more. My mother and father, Haya Al-Karimi and Ahmad Al-Shareef, thank you for ... everything. You made me into who I am. For my brother Mansour and his sons, Nasser and Hussain, thank you for enlightening my darkest days with your precious laughters.

To all my friends scattered around the globe, in particular: Nora Alkhaldi, Maryam AlRayyashi, Noura and Bedour Abouammoh, Manal AlGhamdi, Najwa AlGhamdi, Wadha AlMattar, Noof AlMulihan, Rana Zaini and Hanan Niaz, thank you for making Sheffield felt like home even when I'm thousands of miles away. Thank you for the memorable trips and chats, our "girlish" parties and for being the sisters I've never had.

Finally, despite my love for what I do, the work reported in this thesis would not have been possible without the financial support of Umm Al-Qura University and the Saudi Ministry of Education, for which I am grateful.

Abstract

Colloquial Arabic (CA) is the set of spoken variants of modern Arabic that exist in the form of regional dialects and are considered generally to be mother-tongues in those regions. CA has limited textual resource because it exists only as a spoken language and without a standardised written form. Normally the modern standard Arabic (MSA) writing convention is employed that has limitations in phonetically representing CA. Without phonetic dictionaries the pronunciation of CA words is ambiguous, and can only be obtained through word and/or sentence context. Moreover, CA inherits the MSA complex word structure where words can be created from attaching affixes to a word.

In automatic speech recognition (ASR), commonly used approaches to model acoustic, pronunciation and word variability are language independent. However, one can observe significant differences in performance between English and CA, with the latter yielding up to three times higher error rates.

This thesis investigates the main issues for the under-performance of CA ASR systems. The work focuses on two directions: first, the impact of limited lexical coverage, and insufficient training data for written CA on language modelling is investigated; second, obtaining better models for the acoustics and pronunciations by learning to transfer between written and spoken forms. Several original contributions result from each direction. Using data-driven classes from decomposed text are shown to reduce out-of-vocabulary rate. A novel colloquialisation system to import additional data is introduced; automatic diacritisation to restore the missing short vowels was found to yield good performance; and a new acoustic set for describing CA was defined. Using the proposed methods improved the ASR performance in terms of word error rate in a CA conversational telephone speech ASR task.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

1.1 Varieties of Arabic

Arabic is the official language for 22 countries with around 300 million native speakers. It has two main variants: standard and colloquial Arabic. Classical Arabic (CLA) has been codified by the Qur'an, the Islamic Holy book, and pre-Islamic poetry. A modernised version of CLA that retained its syntax but developed its vocabulary is known as Modern standard Arabic (MSA). MSA is taught in schools and used for formal written and oral communication and discussions such as lectures, public speeches, news, magazines and books. Learning MSA facilitates access to written media in the language system. In contrast to the written language, which has maintained its morphology and syntax throughout the years, varieties of modern Arabic have appeared in the spoken language as regional dialects that are significantly distinct from MSA phonologically, syntactically and morphologically. These dialects, collectively known as colloquial Arabic (CA), are defined as the mother-tongues for individuals in any Arabic country. CA is also referred to interchangeably as dialectical Arabic and conversational Arabic because it only exists in dialects and is used mainly for conversations.

CA introduced new phones to the Arabic phonetic system and some of the MSA phones are not used in some of the CA variants. The MSA phonetic system has 28 consonants,

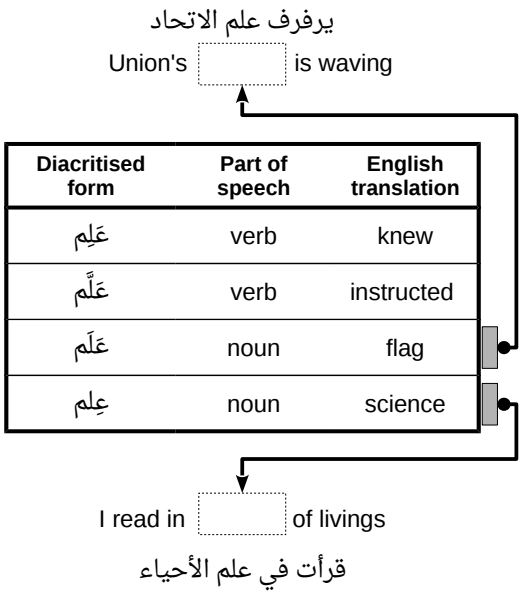


Figure 1.1: Choosing a diacritised form for the undiacritised word "علم" according to the given context.

three vowels and two diphthongs. Duration is a phonemic in Arabic generally, consequently, the phonemic inventory is expanded to include short and long variants of each phone. Arabic is written using two elements: letters, also known as graphemes, and diacritics. In MSA, most graphemes have a one-to-one mapping to a phone. Diacritics are small strokes that are added to the top or bottom of a grapheme to lengthen its duration or attach a short vowel to it. An Arabic sentence, written using graphemes and diacritics, provides the best possible representation for its phonetic realisation. Despite the significance of diacritics in describing the phonetic realisation, the use of diacritics is optional and they are actually rarely added. Consequently, an undiacritised word is ambiguous. Every grapheme can be diacritised using one of the six combinations of diacritics; however, not all possible diacritisations are valid. A conventional Arabic dictionary provides a list of valid diacritisations for each Arabic stem, without phonetic transcriptions as in an English dictionary. Each diacritised form carries its own meaning and the choice for a certain diacritised form is based on its surrounding context. Figure 1.1 illustrates an example of choosing a diacritised form for the Arabic word "علم" according to the context. In the top sentence, "يرفرف علم الاتحاد" (Union's flag is waving), the chosen diacritisation is the one suitable to fit in the context of the sentence.

CA increases the level of ambiguity by introducing additional phones. Because CA is a spoken language, it does not have a standardised writing system. No standard graphemes

are assigned for these new phones. Arabic writers improvise the spelling of a given CA word inconsistently. Writers are usually influenced by their knowledge of MSA, which makes them substitute a spoken CA word with its original MSA form even if the latter does not describe the spoken word phonetically.

Arabic in general has a complex word structure. An Arabic word can be created by applying one or more modification processes to an abstract root. These processes are known as morphological processes. Inflection, a morphological process, concatenates some linguistic particles, such as pronouns, to a verb or a noun. As a consequence, an Arabic word can be equivalent to a full sentence. For example, the word "علمته" is equivalent to the English sentence (I taught him). This property of Arabic allows freedom in the word order since each word almost encapsulates a self-sufficient meaning.

1.2 Colloquial Arabic automatic speech recognition

Automatic speech recognition (ASR) aims to convert an acoustic speech signal into written form. To achieve state-of-the-art performance, ASR systems require transcribed speech corpora, pronunciation dictionaries and a large volume of textual data to model the large variability in the acoustics, pronunciations and word sequences respectively. Collecting or constructing these resources is a costly task where linguistic expertise is necessary.

Over the past 50 years, ASR performance has improved steadily. Figure 1.2 shows an overview of the recognition performance of ASR in the NIST benchmark tests. For example, English systems for transcribing conversational speech achieved 14% word error rate in the NIST 2009 evaluation. In contrast, Egyptian CA systems in the NIST 2003 evaluation and Levantine CA systems in the NIST 2004 evaluation have significantly poorer performance with 37.5% and 46.5% word error rates respectively, three times worse performance than English systems. Reviewing more recent state-of-the-art ASR systems designed for CA, the performance range has not significantly changed. For instance, the recognition performance for Iraqi CA ASR systems achieved 32.1% WER (Afify et al., 2006) and Levantine CA ASR achieved 39.7% WER (Soltau et al., 2011).

The main reason for favouring MSA over CA in developing ASR systems is that MSA is shared by all native Arabic speakers and accepted as the formal Arabic language. As two CAs can widely differ from each other, speakers from these dialects use MSA to communicate. Hence, most language processing technology has targeted MSA instead of CA to access more users. Automatic speech recognition for Arabic, in particular for MSA, has been studied for two decades and, as for English, has improved steadily.

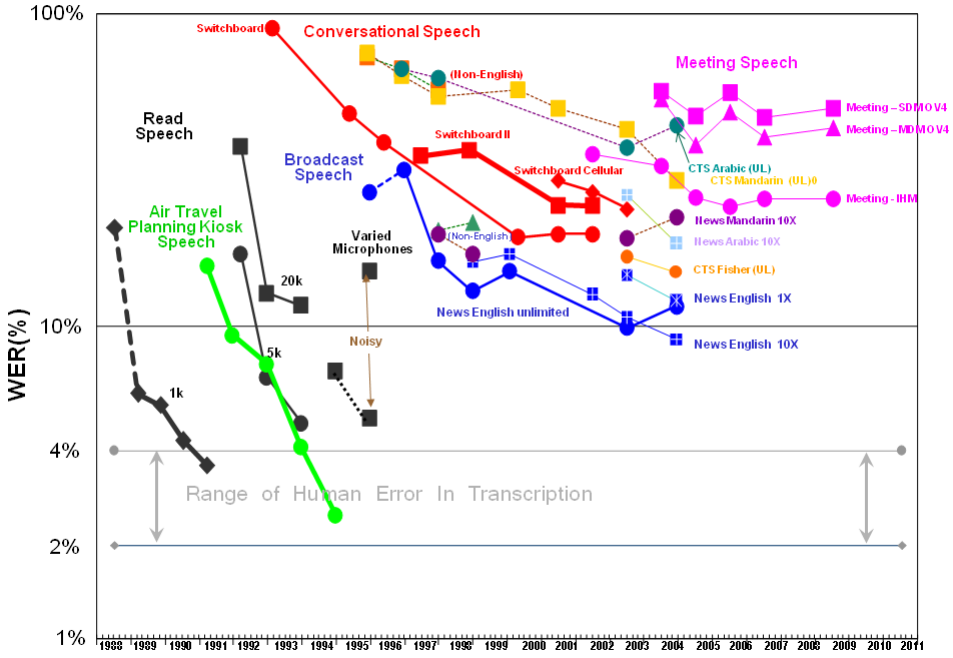


Figure 1.2: NIST STT benchmark test history 1988-2009. CA ASR performance is marked with green circles and MSA ASR performance is marked with light blue squares. Conversational English ASR performance is marked with orange and red markers. (NIST, 2009)[1].

1.3 Motivations and problem definition

MSA is only accessible for literate and educated members of the Arab society and only can be learned in schools. However, CA is accessible for all Arab society as their own mother-tongue where they learn it in their homes. An educated native Arabic speaker can switch between the two variants according to the situation. For example, news anchors use MSA for broadcasting the news on air and then switch to CA when speaking to their colleagues off the air. For written communication, MSA has been used traditionally for all written correspondence whilst CA has remained only a spoken language without a standard writing system. This is starting to change with the emergence of social media, which provide an informal environment for communication. Arabic participants begin to write as they speak and consequently, more CA words have started to be used in the digital media. Without a standard writing convention, native Arabic writers improvise their own spelling which is mainly derived from MSA writing conventions.

With the increased demand of native Arabic speakers for accessing information using their own mother-tongue, more technologies emerge to facilitate using CA. As outlined above, ASR systems for CA suffer from a significant under-performance in comparison to

[1]Copyright 2009 by NIST. Reprinted with permission.

both conversational English and MSA ASR systems. Such highlighted difference suggests the existence of underlying modelling issues so different approaches must be considered. Several potential sources for the poor performance in CA can be identified in the following.

Due to the rich morphology of Arabic, words are created from concatenating or inflecting, small particles at the beginning or at the end of a given word. These particles may represent pronouns and prepositions. For example, the inflected word from adding the particle "الـ" (the) at the beginning of the word "علم" is (flag) is "العلم" (the flag). Further inflection can be applied to create the word "بالعلم" (by/with the flag). Language models depend on counting word occurrences in a context to be able to estimate word probabilities in that context. This estimation is made for a selected word list which usually part of the seen vocabulary in the training corpus. If the two words: "علم" (flag) and "العلم" (the flag) occur in the same context, they will be counted separately even though they are essentially referring to the same word. As a consequence, more training samples are required in order to obtain reliable word probabilities for each inflected word. In an English case, including (flag) would be sufficient to cover the three sentences: (flag), (the flag) and (with the flag), whilst in the Arabic case, all three inflected words must be included in the chosen word list to have the same lexical coverage. Otherwise, these inflected words not included in the word list cannot be recognised. If a word is not included in the chosen word list, it is referred to as an out-of-vocabulary (OOV) word. The percentage of the total occurrence of OOVs in a given text is known as the OOV rate. It is expected for languages with rich inflectional morphology, such as Arabic, to have high OOV rates. For instance, a 60K lexicon has an OOV rate of less than 1% for North American English broadcast news (Rosenfield, 1995); however, for a similar task in rich morphological languages such as Finish this rate reached 15% for a 69K lexicon (Hirsimäki et al., 2006), 10% for Estonian with a 60K lexicon (Hirsimäki et al., 2006), 9.3% for Turkish with 50K lexicon (Arısoy et al., 2008) and 5.4% for MSA with 64K lexicon (Vergyri et al., 2008).

There is a large discrepancy between the written and spoken forms in MSA in general and in CA in particular. This is due to the omission of diacritics, which describe a large proportion of the phonetic realisation. English has an even larger difference between the written and spoken form; however, this is resolved by phonetic dictionaries. Phonetic dictionaries map the written form, in letters, with its spoken form, in phones. For example, the word "phone" is mapped to /foʊn/. Such dictionaries do not exist for Arabic. As aforementioned, a conventional Arabic dictionary provides a list of valid diacritisation variants for a given word without an explicit phonetic transcription. This implicitly shows an assumption that a fully diacritised variant is equivalent to a phonetic transcription. For a collection of symbols to be defined as a phoneset, i.e. a phonetic description alphabet, they must be phonetically separable. Choosing Arabic graphemes as the phonetic units

will not provide the desired phonetic separation because there exist multiple graphemes associated with the same phone. Moreover, CA has introduced additional phones which were associated inconsistently with some of the existing graphemes, hence, multiple phones are associated with the same grapheme. This invalidates the assumption about the optimality of diacritised form as a phonetic transcription.

1.4 Research objectives

Motivated by the above discussion, the work in this thesis is divided in to two directions: first, investigating linguistic issues (Objectives 1 and 2); second, narrowing the gap between the written and the spoken forms of CA (Objectives 3 a nd 4). The following objectives were chosen to address each of the identified issues. For each objective, a definition statement along with related research questions are presented.

Objective 1: Investigate word decomposition in colloquial Arabic language modelling

Arabic is a morphologically rich language so the rate of vocabulary growth increases as the amount of training data increases. This is reflected in high out-of-vocabulary (OOV) rate, and therefore a worse recognition performance. It has been reported that using morphological decomposition of words can limit vocabulary growth. For instance, the three inflected words: "علم" (flag), "العلم" (the flag) and "بالعلم" (by/with the flag) are decomposed as follows:

"علم" (flag)		"علم"
"العلم" (the flag)	$\xrightarrow{\text{decomposed into}}$	"الـ علم"
"بالعلم" (by/with the flag)		"بـ الـ علم"

Here, the word "علم" would have more reliable probability estimates. However, there has been limited work on this approach due to the lack of appropriate tools for morphologically analysing a CA sentence. To counter this, the first objective of this thesis is to investigate whether word decomposition can be effective in reducing the OOV rate and perplexity for an CA language model.

Objective 2: Investigate the use of MSA resources for CA language modelling

Increasing the amount of training data improves language models and their prediction ability. As discussed earlier, with the existence of inflected words, more training samples are necessary to obtain reliable word estimates and better prediction ability. Since CA is

only a spoken language, the lack of training data becomes a core obstacle to implementing language models with reliable word probabilities. However, there exists a considerable amount of textual resources in MSA that may be of some use. The second objective of this thesis is to investigate methods that use the available MSA textual resources to train a CA language model.

Objective 3: Narrow the gap between the spoken and written forms of CA

Most Arabic resources are written in undiacritised form where short vowels are not marked. A reader can infer these vowels and disambiguate words from their context. Although the diacritised form of a CA word might not equivalent to its phonetic transcription, with no doubt it describes a spoken CA word better than the undiacritised form. Therefore, retrieving these diacritics can be considered as a first step toward narrowing the gap between the spoken word and its common written form. All the available automatic diacritisation tools either rely on the local context or require a morphological analysis preprocessing stage, and almost all of them are designed for MSA. The third objective of this thesis is to develop an automatic diacritisation technique for colloquial Arabic using a longer context and without any prior linguistic preprocessing. This is taken further with an investigation of whether the inclusion of the phonological profile, such as the associated phone classes, and paralinguistic information, e.g. a speaker's gender and dialect, can be used as auxiliary information to improve diacritisation accuracy.

Objective 4: Find an appropriate acoustic inventory for CA

CA introduces several new phonemes to the Arabic acoustic inventory in addition to new mappings between the graphemic and phonemic systems. This creates a high level of ambiguity in generation of a pronunciation dictionary, which for Arabic is customarily derived directly from the fully diacritised written form. Thus there is a need to define a wider acoustic inventory. The fourth objective is to redefine the Arabic acoustic inventory to accommodate these new colloquial mappings, and to investigate if this new inventory has a clear mapping to the graphemic form where it can be learned in grapheme-to-phoneme techniques to generate new pronunciations to be used in speech recognition tasks.

1.5 Thesis overview

This thesis can be grouped into four parts, that are displayed in Figure 1.3. Here, each chapter is represented by a block and the dependencies between chapters are represented by arrows. Chapters are theoretically grouped with dashed lines. At first, background

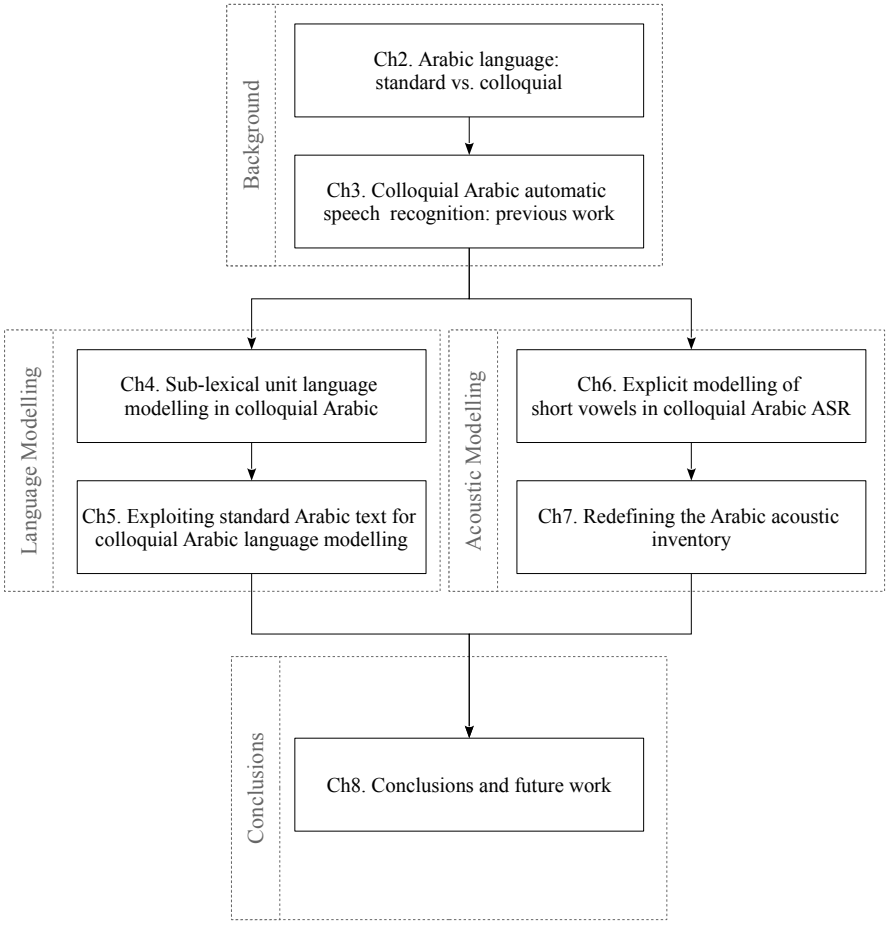


Figure 1.3: Thesis structure. Each chapter is represented by a block and the dependencies between chapters are represented by arrows. Chapters are grouped in parts within dashed lines.

is provided about the Arabic language (Chapter 2) and the tasks this thesis is concerned with (Chapter 3). The background from Chapters 2 and 3 is necessary for the two main contribution areas achieved in this thesis: language modelling (Chapters 4 and 5) and acoustic modelling (Chapters 6 and 7) for colloquial Arabic ASR. Chapter 8 presents conclusions and suggestions for future work. The following gives detail of the contents of each chapter.

Chapter 2 provides background information about the Arabic language and its variants. It gives a thorough comparison of CA to its standard counterpart phonologically, orthographically and morphologically. It also identifies the challenges that the language properties pose for developing an ASR system. Chapter 3 provides a concise introduction to the fundamentals of ASR along with an overview of the recent advancements and state-of-the-art technologies in developing an automatic speech recognition for CA. It identifies and summarises the current gaps in this area.

Chapter 4 discusses related research that employs word decomposition, especially for inflectional and morphologically rich languages such as Arabic, Turkish and German. As this chapter covers the intended work for Objective 1, the three methods of word decomposition are explained and discussed in terms of applying them to CA text.

The majority of the work in Chapter 5 is directed towards Objective 2 of this thesis. The chapter starts with a critical discussion of techniques for importing standard Arabic resources to build colloquial Arabic language models. Then, three different techniques are proposed and applied on word-level and morph-level.

As Chapters 4 and 5 focus on language modelling issues and pursue the first two objectives of the thesis, Chapter 6 and 7 shift the focus to the acoustic and pronunciation modelling issues in CA. Normally in written text, diacritics are optionally added because they can be inferred from the word and the context, thus a considerable part of the acoustic information is not written. Chapter 6 investigates retrieving the omitted diacritics from the textual data of CA, a task which derives from Objective 3 of the thesis. Two automatic diacritisation schemes that incorporate context information are proposed and empirically evaluated on a conversational telephone speech (CTS) ASR task. The context information considered for these tools is not only derived from the textual level but also from higher levels such as the extralinguistic level. As a fully diacritised form is retrieved, Chapter 7 concentrates on Objective 4 and discusses the issues of using graphemic units in acoustic modelling and addresses each of them to reduce the resulting ambiguity in the acoustic model. As a result, two acoustic inventories are introduced along with derivation algorithms for each inventory. The proposed derivation procedures are analysed and the resulting inventories are empirically evaluated in CTS ASR tasks.

Finally, the thesis is concluded in Chapter 8, which also summarises the main contributions of this thesis and suggests directions for future research.

Chapter 2

Arabic language: standard vs. colloquial

Contents

Arabic is the most popular living Semitic language and is the official language for 22 countries with around 300 million native speakers. In addition, Classical Arabic (which is one of the Arabic variants) is the liturgical language for Islam where it is learnt by 1.6 billion Muslims around the world. A variant similar to classical Arabic, known as Modern Standard Arabic (MSA), is used primarily in education and formal media and correspondence but does not exist as a native language. Arabic has changed with the

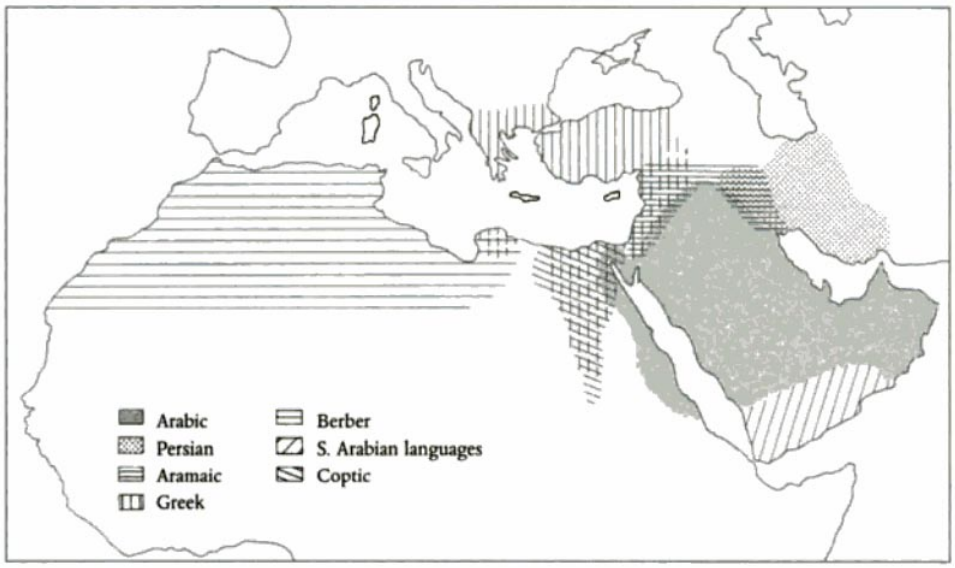


Figure 2.1: The language status at the beginning of Islamic conquests (Holes, 2004)[1].

exposure to other languages over the years, resulting in new variants of Arabic, namely dialects or colloquial Arabic (CA). In contrast to MSA which is taught at schools in both written and spoken forms, CA is used exclusively in informal daily conversations, is not written and is learnt at home as the native tongue.

This chapter gives a phonetic, orthographic and morphological overview of the Arabic language and its dialects. In addition, it outlines the main challenges to developing an ASR system for the language.

2.1 Varieties of Arabic

With the emergence of Islam in the fifth century, classical Arabic (CLA) was centralized in the Arabian Peninsula, and was codified by the Qur'an, the Islamic Holy book, and pre-Islamic poetry. The association with Islam made Arabic the official and religious language of Islam as it spread from the Arabian Peninsula to cover western Asia to North Africa and the south of Europe. At the beginning of the seventh century, Islamic conquests carried Arabic natives to the conquered territories and Arabic was used as the main means of communication between local residents and the incomers, who are usually Islamic officials (Holes, 2004). Figure 2.1 locates the boundaries of languages at the beginning of the Islamic conquests. Arabic in those regions went through number of developments in the following period until the twelfth century. Fearing the loss of the ability to understand Qur'an

[1]Copyright 2004 by Georgetown University Press. From Holes (2004). Reprinted with permission. www.press.georgetown.edu.

Table 2.1: Summary of differences between the high (H) and low (L) variants of the Arabic language in a diglossic speech community as discussed by Ferguson (1959).

Aspect	H variant (MSA)	L variant (CA)
Function	Formal.	Informal.
Prestige	Prestigious and beautiful.	Corrupted and not real.
Literary heritage	Highly esteemed with long history.	Less appreciated literature such as political cartoons.
Acquisition	Formal education.	At home as a mother tongue.
Standardization	Orthography is well established. Established norm for linguistic resources. Very limited variation in pronunciation.	No settled orthography. No norm for linguistic resources. Wide variation in pronunciation.
Stability	Stable.	Changing.
Grammar	Complex grammatical structure.	Simplified grammatical structure.
Lexicon	Technical terms and learned expressions.	Popular expressions.
Phonology	Considered as the basic phonological system.	Evolving.

in Arabic, elite and cultured people, such as Al-Khalil (Al-Farahidi, 1980) and Sibawayh (Sibawayh, 1983), codified norms and rules from their way of speaking to retain the Arabic language (Ibn Khaldoun, 2001).

Until the nineteenth century, the original Arabic became exclusively written and was used only by the elite, literate society. Then, Arab countries were colonized by non-Arabic civilizations for more than fifty years, influencing the Arabic lexis and stylistics and resulting in the modernized version of the CLA known as Modern Standard Arabic (MSA) (Hawkins, 1983; Watson, 2002). Today, MSA is taught in schools and used for most of the written media, such as newspaper, magazines and books. Also, it is used in formal oral communication and discussions, such as in lectures, public speeches, broadcast news and most of talk shows. While the written language has maintained its morphology and syntax throughout the years, varieties of modern Arabic have developed along side it, appearing in the spoken language as regional dialects which are significantly distinct from MSA phonologically, syntactically and morphologically. Ferguson (1959) was the first to refer to this sociolinguistic phenomenon as diglossia and distinguished it from a situation where two dialects co-exist within a single speech community. In diglossic speech communities, there are two varieties of a language where each has its functionally exclusive domains: a highly valued variant, referred to as H, that is learnt at schools and is not used for daily conversations, and a low variant, referred to as L, that is used in informal

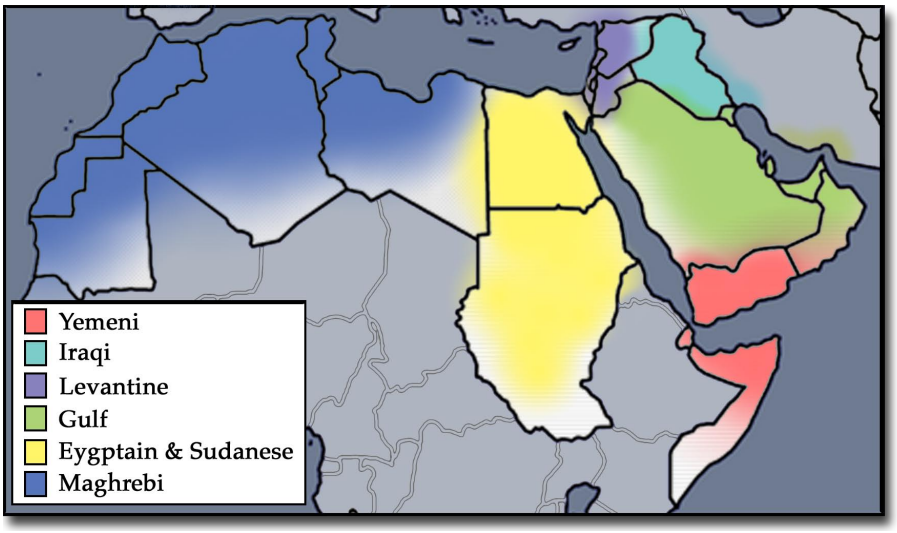


Figure 2.2: Map of the Arabic dialects as mentioned in Alghamdi (2000) and Watson (2002); The continuous spectrum of dialect variations does not take into account national borders.

conversations. According to Ferguson, H dialect is believed by its speakers to be the real language while L dialect is a corrupted form of the language. Table 2.1 summarises other differences between H and L varieties in Arabic.

There are about thirty different Arabic dialects (Lewis, 2009) which can be classified geographically, socially or phonologically. Linguists, such as Alghamdi (2000) and Watson (2002), have categorized those geographically into two major groups: Mashriqi (Eastern) and Maghribi (Western) Arabic. The former includes roughly all the dialects to the east of Sudan and Chad to the border of Oman; the latter includes all the dialects to the west of that line as far as Morocco. Additionally, these groups can be divided into six dialectical families based on their origins and characteristics: Magrebi, Egyptian, Levantine, Iraqi, Gulf and Yemeni (Maisel and Shoup, 2009). The approximate geographical spread of these dialects is shown in Figure 2.2. As stated by Watson (2002):

> *"Dialects of Arabic form a roughly continuous spectrum of variation, with the dialects spoken in the eastern and western extremes of the Arab speaking world being mutually unintelligible."*

Considering the aforementioned development of the dialects, Holes (2004) observed that socioeconomic lifestyle has a cross-cutting effect over the geographical distribution of phonological systems, especially in the Bedouin dialects. Bedouin dialects show a greater degree of similarity, in terms of their phonology and morphology, over geographical spread of the Arab world. Bedouin people reside far from the cities where interaction with other foreigners is limited. Holes accounted for the similarity across Bedouin dialects by the

slow effect of external influences, as compared with those who live in the cities, though this similarity was not observed among urban groups across national borders. For example, a Libyan and Saudi Bedouin dialects are more similar each other than their urban counterparts.

2.2 Arabic phonology

In general, by comparing the number of consonants against the number of vowels, Arabic language has a rich consonantal system compared with its limited range of vowels with twenty-eight consonants in total and only three vowels. Dialects add yet more flexibility to the vowelic system and limit some of its consonantal inventory.

This section summarizes the phonological similarities and differences between MSA and its colloquial varieties.

Consonants: there are twenty-eight distinct consonantal sounds. A set of the prominent acoustic sounds in Arabic are those articulated in the velar and postvelar regions of the vocal tract, heard in /k/, /q/, /ɣ/, /x/, /ħ/, /ʕ/, /ʔ/, and /h/[2]. Another evident feature of Arabic phonology is the emphatic consonant, articulated by retracting the tongue root. These are the emphatic alveolar plosives /tˤ/ and /dˤ/, the emphatic alveolar fricative /sˤ/ and the emphatic dental fricative /ðˤ/. All Arabic consonants are illustrated in Figure 2.3.

As a part of spoken dialect development, additional phonemes appeared in some dialects, such as /g/, /v/, /tʃ/ and /p/, and some of the existing MSA phonemes disappeared or have become rarely used. For example, the uvular plosive /q/ in MSA is replaced by the glottal stop /ʔ/ in Levantine and Egyptian dialects, while it was raised to the velar plosive /g/ in Gulf and Yemeni or /k/ in Jordanian and Palestinian and softened in Sudanese to the velar fricative /ɣ/. Another example is the retraction of the articulation location from the dental, as in /θ/ and /ð/, to alveolar location, to be /s/ and /z/ respectively. More of these different realizations are summarized in Table A.1 in Appendix A.

Vowels: there are three vowels, which are the open-low front unrounded vowel, /a/, the close-high back rounded vowel, /u/, and the close-high front unrounded vowel, /i/.

These vowels shift slightly to neighbouring vowels in CA and create allophones. Two

[2]Phonetic sounds are represented using IPA symbols (IPA, 1999) between two slashes (/).

	Bilabial	Labiodental	Dental	Alveolar	Postalveolar	Retroflex	Palatal	Velar	Uvular	Pharyngeal	Glottal
Plosive	b		t d tˤ dˤ					k g	q		ʔ
Nasal	m			n							
Trill				r							
Tap or Flap				ɾ							
Fricative		f v	θ ð ðˤ	s z sˤ zˤ	ʃ ʒ			x ɣ		ħ ʕ	h
Lateral fricative				ɬ							
Approximant							j				
Lateral approximant				l							

Where symbols appear in pairs, the one to the right represents a voiced consonant.
Where symbols appear in two rows, the bottom one represents emphatic consonants.
Shaded areas denote articulations judged impossible.

Additional sounds:
The labio-velar central approximate (w)

Figure 2.3: Arabic consonants shown in an IPA consonants chart (IPA, 1999); grey shading marks sounds that are used in colloquial Arabic only. Adapted from Alghamdi (2000) and Watson (2002)

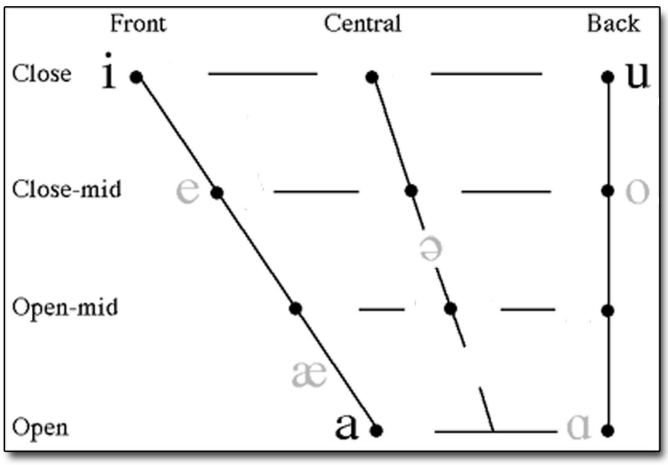


Figure 2.4: Arabic vowels shown in an IPA vowels chart (IPA, 1999); vowels in grey appear in CA only. Adapted from Alghamdi (2000) and Watson (2002).

phones are considered allophones if they are perceived as a single phoneme. Generally, the open-low front unrounded vowels /a/ may retract to the open-low back unrounded vowel /ɑ/ when it occurs with emphatic consonants while it may raise to the open-mid vowel /æ/ when it occurs with labial consonants. In contrast, both close vowels, /i/ and /u/, are lowered to the close-mid vowels /e/ and /o/, respectively, especially for loanwords such as /ʃokola:ta/ (chocolate). Some western Arabic dialects today merge two vowels together, either /i/ and /u/ or /a/ and /u/ into /ə/(Watson, 2002). Eastern Arabic dialects offer a more flexible vowel system than MSA. Figure 2.4 locates the Arabic vowels on the IPA schematic vowels chart (IPA, 1999) and shows in grey the vowels that are used in dialects but not in MSA.

Diphthongs: two diphthongs exist in MSA, which are: /aw/ and /aj/, but they are less frequently used in dialects and might be replaced with other vowels such as /o/ and /e/ respectively (Watson, 2002). For instance, the Arabic word for "sword" is pronounced as /sajf/ in MSA but as /se:f/ in CA, and the word for "turn" is pronounced as /dawr/ in MSA but as /do:r/ in CA.

Duration is phonemic in Arabic by which a phone is pronounced for an audibly longer time, usually twice the length of a normal phone (Alghamdi, 2000). Consequently, the basic Arabic phonemeic inventory is composed of 64 phonemes made of short and long phones. Longer consonants are called geminated consonants. For example, for the word /darasa/ (he studied), if the first vowel /a/ is pronounced longer, i.e. /da:rasa/, it means (he studied together with), whereas the word /dar:asa/ with a geminated /r/ means (he taught). Generally, phonetic duration is retained in most Arabic dialects except for dialects

that replace some short vowels with /ə/. In such dialects, long vowels become shorter while keeping their vowelic quality (Holes, 2004).

2.3 Arabic orthography

In this section, a brief overview of how words and sentences are written in the Arabic language will be given. The Arabic phonemic inventory is represented through script that is written from right to left using graphemes, or letters, along with diacritics above and below those letters. Letters are connected together horizontally from right to left to form words. The words follow each other and are separated by a blank or a white space to constitute a sentence.

2.3.1 Arabic graphemes

There are thirty-six different graphemes in the Arabic writing system, where each one of them alters its shape according to its position in the word, whether it is in an initial, middle, final or isolated position. The main goal for having these different shapes is to connect those letters together into one continuous unit that forms a word. However, not all letters can be connected from both sides. Figure 2.5 lists all different shapes for each Arabic graphemes along with their names. Some graphemes have restrictions on which position in a word they are allowed to appear at, shown as symbol "_" in Figure 2.5, for instance (ؤ) and (ئ) cannot begin a word while (ة) and (ى) can only appear as word endings.

2.3.2 Diacritics

The Arabic writing system uses eight diacritics to represent phonetic phenomena. Some of these diacritics are phonemic and some are morphemic and syntactic, i.e. they identify the word case or state but not the meaning. Three of these diacritics are used to represent the short vowels that can be written above the letter (/a/ and /u/) or below it (/i/) and another three diacritics mark nominal indefiniteness in MSA, known as nunation, which adds /n/ to the attached short vowel. There exists a special diacritic that is written above a consonant to represent its geminated version where it can be combined with any of the previous diacritics as shown in Figure 2.6. Finally, there is also a diacritic for marking the absence of vowels.

Lunar graphemes						
Buckwalter	Name		Final	Medial	Initial	Isolated
ا	alif	ألف	ـا	ـا	ا	ا
>			ـأ	ـأ	أ	أ
<			ـإ	ـإ	إ	إ
'	hamza	همزة	ء	ء	ء	ء
&			ـؤ	ـؤ	ـؤ	ؤ
}			ـئ	ـئ	ـئ	ئ
b	baa	باء	ـب	ـبـ	بـ	ب
j	jeem	جيم	ـج	ـجـ	جـ	ج
H	haa	حاء	ـح	ـحـ	حـ	ح
x	khaa	خاء	ـخ	ـخـ	خـ	خ
E	ayn	عين	ـع	ـعـ	عـ	ع
g	ghayn	غين	ـغ	ـغـ	غـ	غ
f	faa	فاء	ـف	ـفـ	فـ	ف
q	quaf	قاف	ـق	ـقـ	قـ	ق
k	kaf	كاف	ـك	ـكـ	كـ	ك
m	meem	ميم	ـم	ـمـ	مـ	م
h	haa	هاء	ـه	ـهـ	هـ	ه
p	taa marbwta	تاء مربوطة	ـة	—	—	ة

Solar graphemes						
Buckwalter	Name		Final	Medial	Initial	Isolated
t	taa	تاء	ـت	ـتـ	تـ	ت
v	thaa	ثاء	ـث	ـثـ	ثـ	ث
d	dal	دال	ـد	ـدـ	دـ	د
*	thal	ذال	ـذ	ـذـ	ذـ	ذ
r	raa	راء	ـر	ـرـ	رـ	ر
z	zain	زين	ـز	ـزـ	زـ	ز
s	seen	سين	ـس	ـسـ	سـ	س
$	shen	شين	ـش	ـشـ	شـ	ش
S	sad	صاد	ـص	ـصـ	صـ	ص
D	dhad	ضاد	ـض	ـضـ	ضـ	ض
T	taa	طاء	ـط	ـطـ	طـ	ط
Z	thaa	ظاء	ـظ	ـظـ	ظـ	ظ
l	lam	لام	ـل	ـلـ	لـ	ل
n	noon	نون	ـن	ـنـ	نـ	ن

Glides and long vowels graphemes						
Buckwalter	Name		Final	Medial	Initial	Isolated
A	alif	ألف	ـا	ـا	ا	ا
Y	alif maqswra	ألف مقصورة	ـى	—	—	ى
w	waw	واو	ـو	ـو	و	و

Figure 2.5: Writing system for the Arabic language. Graphemes are grouped into three sections: Lunar, Solar and Glides and long vowel graphemes. All possible shapes a grapheme can take according to its position within a word are shown along with its name written in Arabic script and Roman letters. The symbol "_" is shown when a grapheme is not allowed at a certain position within a given word. Buckwlater transliteration equivalent is shown for each grapheme.

Diacritics						
Name		Isolated	Without gemination or nunation	Gemination	Nunation	Gemination and Nunation
sukwn سكون	script	◌ْ	بْ	بّْ		
	IPA		/ b /	/ b: /		
	Buckwalter	o	bo	b~		
fatha فتحة	script	◌َ	بَ	بَّ	باً	بًّا
	IPA	/ a /	/ ba /	/ b:a /	/ ban /	/ b:an /
	Buckwalter	a	ba	b~a	baF	b~aF
kasra كسرة	script	◌ِ	بِ	بِّ	بٍ	بٍّ
	IPA	/ i /	/ bi /	/ b:i /	/ bin /	/ b:in /
	Buckwalter	i	bi	b~i	biK	b~iK
dama ضمة	script	◌ُ	بُ	بُّ	بٌ	بٌّ
	IPA	/ u /	/ bu /	/ b:u /	/ bun /	/ b:un /
	Buckwalter	u	bu	b~u	buN	b~uN

Figure 2.6: Arabic diacritics. The name written in Arabic and Roman scripts is shown for each diacritic, along with its pronunciation using IPA symbols and transliteration using Buckwlater scheme. The first column shows a diacritic on its own without being associated with any grapheme while the rest of the columns associate diacritics with the grapheme "b". The second column shows diacritics without gemination or nunation phenomena. Nunation usually comes at the end of the word and it can be combined with gemination and/or short vowel diacritics as well. The vowel-less diacritic *sukwn* can be omitted in fully diacritised writing.

2.3.3 Punctuations

Punctuation is a recent development in Arabic text and it is not used systematically yet. Today, most writers use punctuation marks in a non-standard way and some even write without using them (Van Mol, 2003). However, lack of punctuation does not lead to misunderstanding because word order in Arabic is linked to information organization and the context which can extend to sentences rather than the immediate context of a word (Holes, 2004).

2.3.4 Arabic transliteration

Roman-based alphabets can be encoded by computers today using ASCII encoding but non-Roman languages use a wider set known as Unicode (The Unicode Consortium, 2011). Owing to the difficulty of dealing with Unicode symbols in natural language processing tools, researchers carefully substitute these symbols with more suitable letters that preserve the orthographic features. This substitution is known generally as transliteration. If the chosen character set was formed from a roman-based system, the process is known as romanization.

A number of romanization standards are established to transliterate Arabic script, such as Qalam (Heddaya, 1985), ArabTeX (Lagally, 1992), etc. One of the most popular schemes was created by Tim Buckwalter (2002; 2004a). His is a case-sensitive scheme that maps each Unicode character in the Arabic alphabet and the diacritics to a unique single ASCII- based Roman character. The corresponding character is illustrated in Figures 2.5 and 2.6 for graphemes and diacritics respectively.

2.4 Relation between Arabic phonemes and graphemes

In linguistics, the relationship between the phonetic and orthographic systems in a language is described through its orthographic depth (Coulmas, 1996). A language is considered orthographically shallow if there is a simple grapheme-to-phoneme rule where a letter has only one sound. On the other hand, a language where a letter has more than one sound and more complicated grapheme-to-phoneme mappings is considered to be orthographically deeper.

In Arabic, most letters have a one-to-one relationship with their corresponding phonemes, as is shown in Figure 2.7. However, there are some cases of multiple phonemes being represented by the same grapheme, known as *homographemes*, and some cases where multiple graphemes are mapped to the same phoneme, known as *homophonemes*, while some graphemes become completely silent in certain conditions.

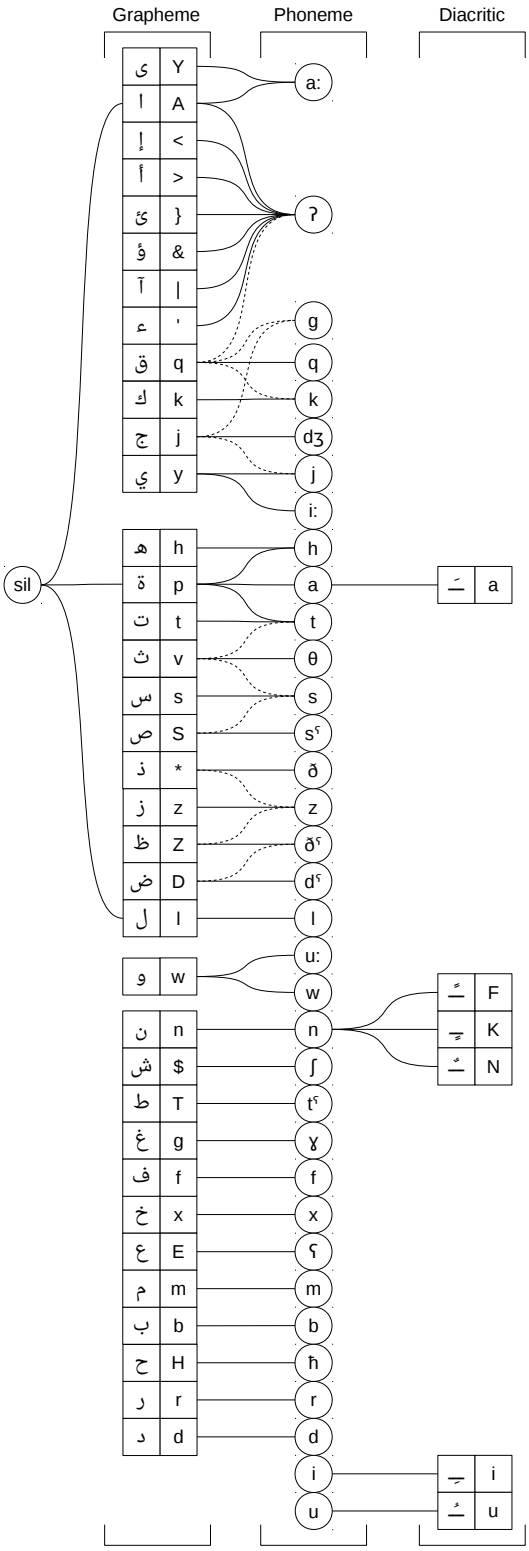


Figure 2.7: Grapheme-to-phoneme mapping in the Arabic language. Each grapheme and diacritic is shown in Arabic script (left) and Buckwalter transliteration (right) while phoneme are written in IPA. Mappings introduced by CA are shown in dashed lines.

Arabic homographemes (multiple phonemes mapped to one grapheme):

- Arabic letter "waw" (و): represents the labiovelar central approximate /w/ and the long version of the rounded close back vowel /u:/.
- Arabic letter "yaa" (ي): represents the voiced palatal approximate /j/ and the long version of the unrounded close front vowel /i:/.
- Arabic letter "lam" (ل): represents the alveolar lateral approximate /l/ except in the word الله "Allah" and its derivatives when it is pronounced as /ɬ/.
- Arabic letter "taa marbuwtah" (ة): a morphophonemic character that is used as a feminine marker, always comes as a word final and can be followed by a diacritic only. In MSA, it is pronounced as /t/ or, when it is at the end of the sentence, /h/.

Arabic homophonemes (one phoneme mapped to multiple graphemes):

- The glottal stop /ʔ/:
  - Arabic letter "hamza ala alif" (أ): represents the glottal stop /ʔ/ followed by the short vowel /a/. In addition, it is used to represent the glottal stop /ʔ/ followed by the short vowel /u/ when it is the word initial.
  - Arabic letter "hamza taht alif" (إ): represents the glottal stop /ʔ/ followed by the short vowel /i/.
  - Arabic letter "hamza ala nabira" (ئ): represents the glottal sound /yʔ/ and is used only in the middle or final position.
  - Arabic letter "hamza ala waw" (ؤ): represents the glottal stop /ʔ/ followed by the short vowel /u/ when it is in the middle or final position.
- The long vowel /a/:
  - Arabic letter "alif" (ا): represents the long version of the unrounded open front /a/.
  - Arabic letter "alif maqsowra" (ى): a derivation letter that is pronounced as the long version of the vowel /a/ and always comes as a word final.

Conditions of silent graphemes:

- Arabic letter "taa marbuwta" (ة): in dialectical Arabic it is either pronounced as /h/ or silent, whether it appears at the end of a sentence or not, and sometimes as /a:/.

Table 2.2: Different diacritised forms for the undiacritised word "درس", Buckwlater transliteration enclosed within ( ), where each diacritisation specifies different meaning for the word.

Undiacritised	Diacritised	Part of speech	English translation
درس (drs)	دَرساً (darsAF)	noun	lesson
	دَرسٍ (darsK)		
	دَرسٌ (darsN)		
	دَرَسَ (darasa)	active past verb	he learned
	دَرَّسَ (dar∼asa)		he taught
	دُرِسَ (durisa)	passive past verb	has been learned
	دُرِّسَ (dur∼isa)		has been taught

- Arabic letter "alif" (ا): in case of plural verbs ending with /u:/, this letter is written orthographically only and without any phonetic value. Sometimes it refers to the glottal stop /ʔ/, which is pronounced if the word is preceded by a pause, such as the beginning of the sentence; otherwise it is silent.

- Arabic letter "lam" (ل): it will be silent if it is used in the definite article "Al" (الـ) and followed by a dental (/θ/, /ð/ and /ðˤ/), alveolar (/t/, /tˤ/, /d/, /dˤ/, /z/, /s/ and /sˤ/) or post-alveolar (/l/) except /dʒ/. These letters are called "solar letters" and the rest are called "lunar letters", i.e. (/ʔ/, /b/, /dʒ/, /ħ/, /x/, /ɣ/, /ʕ/, /f/, /q/, /k/, /m/, /h/, /w/ and /y/), by which the letter lam is metaphorically depicted as a star that cannot be observed, in this case not pronounced, in the presence of the sun, whereas it can be seen, thus heard, in the presence of the moon.

Optional addition of diacritics: Although diacritics hold short vowel information, diacritisation is optional and used (if at all) only in ambiguous conditions when the reader cannot infer them from the context. For example, an undiacritised Arabic word "درس" can be written in seven diacritised forms, each diacritised form holds a different meaning as shown in Table 2.2. Not all diacritisations are valid, which can be obtained from a conventional Arabic dictionary. A diacritised form is chosen according to a given context because only one or a limited set of diacritisation variants is applicable in a certain context. For example if the word "درس" appears within a context, such as "هذا درس" where the word " هذا" means (this is), disambiguates diacritisation variants to only one option "دَرسٌ" because at this context only a noun is expected. By analogy, the two English sentences "this is their home" and "there is no vowels here" can be written in undiacritised/unvowelised form by removing all short vowels as "ths s thr hom" and "thr s no vwls her". The context allows disambiguating "thr" as "their" in the first sentence and as "there" in the second.

Table 2.3: An example of several pronunciations of the Arabic word "يقول" (he is saying) in CAs which is also written in different undiacritised forms as it is pronounced: MSA, Gulf (GCA), Levantine (LCA), Iraqi (ICA), Egyptian (ECA) and Magharbi (MCA). The third column shows the orthographic forms used by transcribers: left form is how is it commonly written by retrieving to MSA undiacritised form and right form is another written form describes how it is pronounced.

Dialect	Phonetic transcription	Orthographic forms (Buckwalter)	
MSA	/yaqu:l/	يقول (yqwl)	
GCA	/yugu:l/	يقول (yqwl)	
LCA	/y?o:l/	يقول (yqwl)	يئول (y}wl)
ICA	/yugul/	يقول (yqwl)	
ECA	/yu?o:l/	يقول (yqwl)	يئول (y}wl)
MCA	/yqul/	يقول (yqwl)	يقل (yql)

As a result of the optional use of diacritics, there are three orthographic systems: fully vowelised/diacritised, partially vowelised and unvowelised. For the first, all letters and diacritics are shown in the text. Together, they give precise information about the word pronunciation, displaying a shallow orthographic depth. In contrast, the latter two schemes lack all the short vowel information and other phonetic phenomena, such as gemination and nunation, making them orthographically deeper systems.

Using a fully vowelised writing system is almost always restricted to writing religious text and children's reading materials, unlike almost all other written materials which use the semi-vowelised and unvowelised systems.

The appearance of dialectical Arabic in written material is limited to certain situations where using dialects is part of the conveyed message, such as in newspaper cartoons, local poetry, plays and local media scripts. In these situations, words are spelled out phonetically regardless of how they are spelled originally in MSA, using unvowelized or semi-vowelized schemes (Holes, 2004). In some instances, people are beginning to use non-Arabic graphemes, which are borrowed originally from Urdu or Persian script, to represent a dialectal word with less ambiguity (Maisel and Shoup, 2009). This diversity in writing dialectical Arabic leads to the existence of multiple orthographic forms for a single word, despite the fact that some of these forms do not match the spoken word phonetically. Table 2.3 illustrates the word "يقول" (he says) and its multiple written forms according to the dialectical variations. It is noticeable in the example that even if the original MSA orthographic form does not match the dialectical pronunciation it is still considered one of its forms.

2.5 Arabic morphology

Arabic morphology concerns with the word structure by resolving two issues: first, how words are created; second, how they interact with the syntax. Theories that govern the first issues belong to derivational morphology and those governing the second issue are under inflectional morphology. By analogy, derivational morphology in English can analyses how words such as "unreal", "realness" and "realistic" are derived from the word "real". Inflectional morphology in English describes different rules for grammatical agreement in gender, time or number such as appending an "s" to present verbs. In Arabic, the line between derivational and inflectional morphology is not as clear as in English. Moreover, the boundary between inflectional morphology and syntax is not as clear as in English.

Generally, Arabic stems are created from morphological derivation. Afterwords, these stems are modified, using morphological inflection, if necessary to be used within a sentence.

2.5.1 Derivational morphology

All words in Arabic are derived from a semantic abstraction known as a root. It can be composed of two to five disconnected phonemes; only consonants and long vowels are allowed. These phonemes are known as radicals. A root is called a triliteral, a quadriliteral or a quinqueliteral if it has three, four or five radicals, respectively. The most common form of a root is constituted of three radicals which are denoted as $C_1C_2C_3$ (Ali, 1756). In the "Lisan Al-Arab" (Manzur, 1883), an Arabic dictionary, there are 6535 triliterals, 2548 quadriliteral and 187 quinqueliterals, summing in 9273 roots. Semantically related words can be derived from a root that relates to its concept by augmenting vowels and additional consonants into the root based on well-defined patterns. Two levels of morphological augmentation (or derivation) can be applied on a given root. First, creating an augmented root, not a word yet, by lengthening the middle consonant or inserting a vowel between root consonants, which is known as morphosemantic derivation. Second, obtaining a word by further applying a well-defined templates on the root (augmented or not) to derive verbs from verbal templates, or nouns from nominal templates. This process is known as morphosyntactic derivation. Both morphological derivations (morphosemantic and morphosyntactic) are used to apply semantic modifications to the abstract concept of the root.

In general, morphosemantic derivation can be applied by augmenting one, two or three phonemes to a triliteral but only one phoneme to a quadriliteral. The derivation rules and template change according to the whether all radicals are consonants and the position of the vowel in the root if it exists. For example, Table 2.4 lists all the patterns used this derivation from the consonantal triliteral "Elm" (learn concept) except for the pattern

Table 2.4: Morphosemantic patterns that can be on triliterals. The third column shows the semantic modification obtained when applying these patterns along with some examples in the fourth column and its English translation on the sematics in the fifth column. All examples used the triliteral "Elm", where C_1=E, C_2=l and C_3=m except Pattern IX where the triliteral "Hmr", where C_1=H, C_2=m and C_3=r was used.

Number	Pattern	Meaning	Example	English translation
I	$C_1C_2C_3$	stative	Elm	know
II	$C_1C_2C_2C_3$	intensive/causative	El~m	teach
III	C_1v:C_2C_3	make an effort to achieve	EAlm	explore
IV	?$C_1C_2C_3$	causative	>Elm	inform/mark
V	t$C_1C_2C_2C_3$	causative/reflective of II	tEl~m	become learned
VI	tC_1v:C_2C_3	causative/reflective of III	tEAlm	become explorer
VII	n$C_1C_2C_3$	passive	nElm	be informed/marked
VIII	C_1tC_2C_3	similar to V and VII	Etlm	be marked
IX	$C_1C_2C_3C_3$	become to this state*	Hmr~	become red
X	st$C_1C_2C_3$	benefactive	stElm	get information

Table 2.5: Examples of six patterns for augmented root with one phoneme (Pattern II) to define nouns and verbs. All patterned applied on the augmented root "El~m", where C_1=E, C_2=l and C_3=m.

Pattern	Description	Example	English translation
muC_1aC_2C_2iC_3	Actor noun	muEal~im	teacher
muC_1aC_2C_2aC_3	Patient noun	muEal~am	student (being taught)
taC_1C_2iC_3ap	Noun indicating an action happened once	taElimap	instruction
C_1aC_2C_2aC_3	Active verb - Past perfect	Eal~am	taught
yuC_1aC_2C_2iC_3	Active verb - Present imperfect	yuEal~im	he is teaching
C_1aC_2C_2iC_3	Imperative verb to masculine subject	Eal~im	teach!

"IX" which is derived from the triliteral "Hmr" (redden concept), because this patten can only be applied on words that imply physical appearance such as colours or shapes. As shown, each pattern modifies the semantic of the root, which is shown in the corresponding English translation. For instance, "Pattern I" does not augment any additional phonemes to the root so its meaning does not change from the abstract learning concept, but using "Pattern IV" changes the concept from know to deliver the knowledge, i.e. teach. The resulting roots from this process have not being defined as noun or verbs yet; they are still abstract meaning.

Using morphosyntactic derivation, the resulting root becomes either a verb or a noun. There are more 20 patterns for each augmented root that can create different tense and voice of verbs and nouns with different functionality such as actor, instrument, place and time, etc. For example, Table 2.5 shows six patterns to create three verbs and three nouns defined for "Pattern II"-augmented consonantal triliterals. These patterns create words by inserting additional consonants along with short and long vowels between the augmented radicals. For example, the noun "muEal~im" (teacher) is created from the augmented root "El~m" (teaching concept) by replacing radicals in the actor pattern "muC$_1$aC$_2$C$_2$iC$_3$"; where C$_1$=E, C$_2$=l and C$_3$=m.

2.5.2 Inflectional morphology

Most of the created words from morphological derivation are singular and masculine, unless the pattern stated otherwise. Gender, number, person and, in some cases time of the word can be modified by attaching prefixes and suffixes to the derived word. This process is known as inflectional morphology. Number in Arabic can be either singular, dual or plural and gender is ether feminine or masculine. Person can be either the speaking person (first), the spoken to (second) or the spoken about (third). First person is genderless and cannot be dual. For the others, each person can be either feminine or masculine and either singular, dual or plural. for instance, by attaching "p" to the end of a noun it refers to the feminine version of the noun such as "muEal~imp" (female teacher). Another example is by attaching "wA" to the end of an imperative verb to modify its number and gender to be masculine plural such as "Eal~imwA" (teach!-subject is masculine plural).

2.5.3 Difference between MSA and CA morphology

CA inherited the morphological structure of MSA with some modification towards simplifying the overall morphology. Such modification is restructuring the morphosemantic Pattern I and reducing the overall number of morphosyntactic patterns (Holes, 2004). In addition, more affixes are introduced for inflectional morphology and some of affixes in the

Table 2.6: Verb prefixes in both MSA and their alternatives in LCA, which specify the verb number, person and gender. Some of these prefixes require specific suffixes; optional affixes are surrounded by round brackets. The grey-shaded cells indicate the non-existence of the conditions. X should be replaced by a modified root.

Person	Gender	Number					
		Singular		Plural		Dual	
		MSA	LCA	MSA	LCA	MSA	LCA
first	-	ʔX	(b)aX	nX	(b)niX		
second	masculine	tX	(b)tiX	tXu:(na)	(b)tiXu	tXa:(ni)	
	feminine	tXi:(na)	(b)tiXi	tXna	(b)tiXu	tXa:(ni)	
third	masculine	yX	(b)yiX	yXu:(na)	(b)yiXu	yXa:(ni)	
	feminine	tX	(b)tiX	yXna	(b)yiXu	tXa:(ni)	

Table 2.7: Verb suffixes in MSA and their alternatives in LCA which specify the verb number, person and gender. The grey-shaded cells indicate the non-existence of the conditions. X should be replaced by a modified root.

Person	Gender	Number					
		Singular		Plural		Dual	
		MSA	LCA	MSA	LCA	MSA	LCA
first	-	Xtu	Xt	Xna:	Xna		
second	masculine	Xta	Xt	Xtum	Xtu	Xtuma	
	feminine	Xti	Xti	Xtunna	Xtu	Xtuma	
third	masculine	Xa	X	Xu:	Xu	Xu:	
	feminine	Xat	Xit	Xna	Xu	Xata:	

MSA are rarely used or abandoned (Holes, 2004), such as the dual number. Tables 2.6 and 2.7 compare verb prefixes and suffixes in MSA and the Levantine colloquial Arabic (LCA) variant. More flexibility emerges in the usage of gender affixes such as using one set of affixes for both plural feminine and plural masculine (Watson, 2002).

2.6 Arabic syntax

The word order in a sentence is described through the syntax of the language; however, the relationship between the sequence of the words (syntax) and the internal structure of each word (morphology) can be more complicated in Arabic (Holes, 2004). Generally, Arabic words can be classified into three categories (Abdur-Rasheed, 2008):

Noun: a word that describes a concept, a thing or a person. Based on its morphology, a noun could be classified by its number, gender, definition and case. Nouns includes participles, adverbs, circumstantial accusative, pronouns, relatives, interrogatives, and demonstratives.

Verb: a word that describes an action. Also, it can be classified by its number, gender and person. The verb class can be divided further by tense (past, present and future), mood (perfect, imperfect and imperative) and voice (active and passive).

Particle: a word that signifies meaning only in conjunction with other words, which could be verbs or nouns, or phrases. Particles are classified according to the following word, which could be a noun or verb and includes prepositions, conjunctions, interrogative particles, exceptions, and interjections, also a word or an affix added to a verb or a noun. A combination of a particle and a noun is called a prepositional phrase.

Words can be arranged in different orders to compose a sentence. The location of a word in the sentence determines its case, which is further indicated by the ending vowel i.e. case-ending. There are three main cases in Arabic: nominative, genitive and accusative. An Arabic sentence generally takes one of the following two forms:

Nominal sentence: The simplest structure is composed of the nominative subject, which could be a definite noun, proper noun or pronoun, and the predicate, which could be an indefinite noun, proper noun or adjective that agrees with the subject in number and gender. Moreover, the predicate could be a prepositional phrase. For instance, "ha*aA muEal~imunA" (This [is] our teacher) which composed of a demonstrative pronoun and noun phrase. It should be notice that "[is]" in the English translation is implied and there is no verb in the sentence.

Verbal sentence: Typically, an Arabic verbal sentence has a verb-subject-object structure. However, if the object is a pronoun suffix, the structure will be verb-object-subject. The verb can be intransitive, transitive and ditransitive and agrees with the subject in person and gender. For example, "yatEal~am" ([He] is learning) is a complete verbal sentence because the subject is incorporated within the verb as part of its inflection even if it was not explicitly mentioned. A subject can be explicitly specified as in "yatEal~am Ahmad" (Ahmad is learning).

More complicated structures can be composed using a hierarchy of nominal and verbal sentences. The basic word order for verbal sentences is verb-subject-object (VSO); however, this order might change to be in SVO or even VOS. For example,"yatEal~am Ahmad Al~ugap" (Ahmad is learning the language) is literally written (is learning Ahmad the language) in VSO and it can be written as "Ahmad yatEal~am All~ugap" (Ahmad is learning the language) in SVO order. The order VOS is usually used when verbs are inflected with object-pronoun, such as in "yatEal~amuhu Ahmad" (Ahmad is learning it), where the object-pronoun "-uhu" is attached at the end of the verb and preceded the subject in order, that is (is learning it Ahmad) literally.

2.7 Challenges in developing colloquial Arabic ASR

Arabic language poses significant challenges in developing automatic speech recognition (ASR) systems. These challenges include the complex morphology, especially the inflectional morphology, the absence of short vowel representation from the textual form, and the dependency of the case-ending on the syntax (Farghaly and Shaalan, 2009).

In addition to these challenges for developing an ASR system for MSA, more difficulties are introduced by the colloquial variants. First, very limited textual CA data exists in comparison to the volume of MSA text available. These textual resources were the outcome of the previous effort in developing linguistic tools for CA, such as that of (Maamouri et al., 2007) and (Appen, 2007) in Levantine CA. Another recent resource for written CA is found in social media, such as chats and forums, where informal environment for their users. Since CA is a spoken language, not written, no standard convention is universally utilised for transcribing these dialects (Watson, 2002; Holes, 2004). Consequently, native speakers are usually improvise the spelling of colloquial words, which is influenced by their knowledge of MSA. That makes those writers substitute a spoken word with its original MSA form, even if the latter does not describe the spoken word phonetically (Holes, 2004; Maisel and Shoup, 2009). Second, if annotators or users rely on the MSA undiacritised orthographic system which lacks some important phonemic information such as short vowels and gemination. Finally, CA has inherited the complex morphological form of MSA. Moreover, additional affixes are introduced informally and locally, which increases cross-dialectical differences.

2.8 Summary

In this chapter, the Arabic language was reviewed. A comparison between MSA and CA were made in terms of their phonology, orthography, morphology and syntax. A summary of the challenges posed by the nature of the language was listed, which will further discussed in details in the course of this work. Nevertheless, significant effort has been made to develop state-of-the-art ASR systems. A review of this work is presented in the next chapter.

Chapter 3

Colloquial Arabic automatic speech recognition: previous work

Contents

Automatic speech recognition (ASR) tries to replicate a part of human speech communication. As such technology, ASR is a task to which many different sciences contribute, in particular acoustics, linguistics, psychology, engineering, computer science, statistics and mathematics. The purpose of this chapter is to provide a sufficient background on the fundamentals of ASR, and to summarise previous work in CA ASR to date.

This chapter is organised as follows: Section 3.1 provides a concise introduction to the main components of ASR, along with the evaluation metrics for measuring of recognition performance. Section 3.2 summarises the techniques and outcomes of state-of-the-art CA ASR systems. Related work in modelling CA language and acoustics is discussed in Section 3.3 and Section 3.4, respectively. The chapter is concluded in Section 3.5.

3.1 Brief introduction to ASR

ASR is the process of choosing the sequence of words or phonemes that best match a given acoustic waveform. Any given choice is based on a criterion that accounts for acoustic and linguistic similarity between the written and the spoken forms. The scope of this chapter concentrates on statistical HMM-based ASR and related recent advances in the field of colloquial Arabic speech recognition. Information about other approaches for developing ASR systems in general is out of the scope of this thesis and will not be covered in this survey.

A speech signal can be represented using numeric vectors after applying front-end processing. These vectors are called acoustic feature vectors or frames. A statistical ASR system will recognise a sequence of T acoustic feature vectors $X = (x_1, x_2, \dots, x_T)$ to be the most probable sequence of n words $\hat{W} = (w_1, w_2, \dots, w_n)$ given the acoustics X and a set of parameters $\Theta = \{\Theta_a, \Theta_l\}$. The most probable sequence, $\hat{W}$, is given by:

$$\begin{aligned} \hat{W} &= \arg\max_W P(W|X,\Theta) \\ &= \arg\max_W \frac{p(X|W,\Theta_a)\,P(W|\Theta_l)}{p(X|\Theta)} \\ &= \arg\max_W p(X|W,\Theta_a)\,P(W|\Theta_l) \end{aligned} \tag{3.1}$$

In Equation 3.1, the likelihood $p(X|W,\Theta_a)$ is estimated by the acoustic model (AM), where Θ_a are the parameters, while $P(W|\Theta_l)$, which is independent of acoustic information, is the language model (LM), where Θ_l are its parameters (Jelinek, 1976). Before a system is able to perform any recognition task, the system parameters Θ must be learnt from a set of labelled training data. The main components of a standard ASR system are illustrated in Figure 3.1. The rest of this section briefly describes each of these components along with the evaluation metrics for the system performance.

3.1.1 Feature extraction

In solving a pattern recognition problem for a speech utterance only useful information should be considered. This information compromises a set of feature vectors, X. This process is called feature extraction, also known as a front-end component. Given the assumption that speech is stationary over a short period of time, short-term feature vectors are extracted using a sliding window, typically of 25 milliseconds long. As a result, a series of overlapping frames over the speech waveform is extracted. Most common techniques to extract these features are: cepstral-based, such as Mel frequency cepstral coefficients

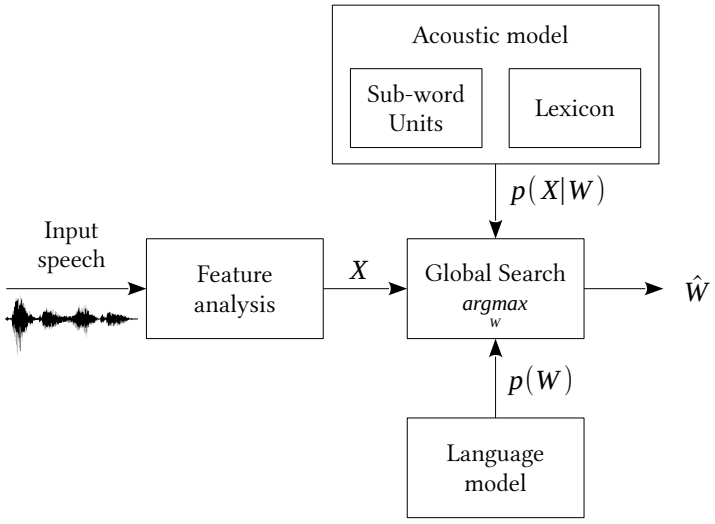


Figure 3.1: An overview of the main components in a standard ASR system. Adapted from Ney (1990)

(MFCC) (Davis and Mermelstein, 1980) and linear prediction based, such as perceptual linear prediction (PLP) (Hermansky, 1989).

3.1.2 Acoustic model and lexicon

The acoustic model represents the relationship between the extracted feature vectors and the word sequence. It finds the likelihood $p(X|W,\Theta_a)$, in Equation 3.1, where X is a sequence of feature vectors, and W is a sequence of words. This likelihood is derived from feeding the model with these acoustic features along with the actual textual transcriptions. As a result, for each time frame in X, a vector of likelihoods that this frame is generated by each possible word in W.

Increasing the amount of training samples improves the reliability of the estimates for each class or word that can be found in W. However, collecting transcribed speech data is expensive. Therefore, speech recognition systems use sub-word units classes as acoustic models since it is more likely for a sub-word unit to be observed in the data than a full word. If the sub-word level is used, a lexicon, also known as a pronunciation dictionary, is required to map a word to its corresponding sequence of sub-words. These sub-word units are usually context-dependent phones. In general, the chosen unit to be be used for acoustic modelling will be referred to as an acoustic unit.

Most state-of-the-art systems use hidden Markov models (HMM) (Baker, 1975; Jelinek, 1976) to capture the variability in the speech represented in the acoustic signal. Figure 3.2 shows an example of a left-to-right HMM model. Each state is associated with weighted

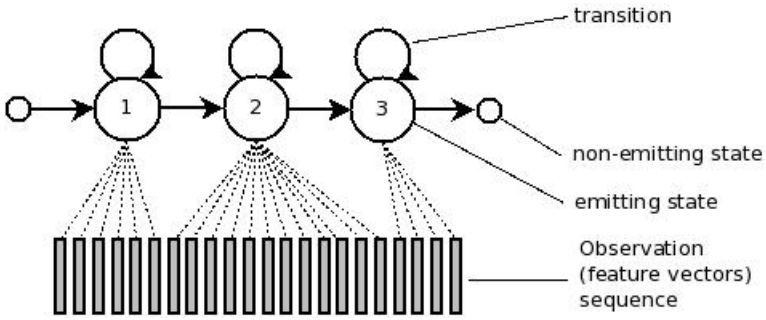


Figure 3.2: An HMM model that contains 3 emitting states and entry and exit non-emitting states. The transition between these states is strictly left-to-right. The produced sequence of states corresponding to the given sequence of observations is hidden.

mixtures of multivariate Gaussian distributions. A left-to-right HMM model containing N emitting states is described with the following elements:

- Transition probabilities, $A = [a_{ij}]$; $i, j \in \{1, 2, ..., N\}$, describe the probability of transiting from state i to state j, i.e. $P(s_j|s_i) = a_{ij}$.
- Observation probability distribution, $P(x_t|s_i)$. This is commonly described using a mixture of multivariate Gaussian distribution for each state i:

$$\begin{aligned} b_i(x_t) &= P(x_t|s_i) \\ &= \sum_{m=1}^{M} \frac{c_{i,m}}{(2\pi)^{(0.5D)}|\Sigma_{i,m}|^{0.5}} \exp\left[-0.5(o_t - \mu_{i,m}^{\top})\Sigma_{i,m}^{-}1(o_t, \mu_{i,m})\right] \end{aligned} \tag{3.2}$$

HMM is considered to have a strong performance in the ASR systems and they can be trained using some well-known algorithms, such as those of Viterbi (Forney, 1973) and Baum-Welch (Baum et al., 1970).

As the number of the mixtures in b_i increases, the number of parameters to be estimated increases. These parameters are estimated for each state independently on a subset of the acoustic features which were aligned with that state. Hence, a large amount of training data is required for reliable estimation. To counter this, state-tying is employed where two related states are tied together if they share the same Gaussians. The relation between states can be defined by the means of binary decision trees which are built based on answering a set of context-based binary questions (Young and Woodland, 1994). An example of these questions: "Is the previous phone a vowel?". Each of these questions separate all acoustic units into two sets, along with the amount of the available training data. The choice and order of these questions is based on the amount of the training data

left in each set and on the achieved likelihood improvement.

Recently, artificial neural networks (ANN) approaches, such as a multi-layer perceptron (MLP), recurrent neural network (RNN) and deep neural network (DNN), are combined with HMM models in architectures which can be either tandem (Hermansky et al., 2000; Weninger et al., 2011; Hinton et al., 2012) or hybrid (Bourlard and Morgan, 1993; Parveen and Green, 2002; Dahl et al., 2012). In hybrid architectures, statistics computed from HMMs are combined with the phoneme posterior probabilities generated from the output layer in the ANN. HMM parameters are optimised jointly with ANN parameters. In tandem architectures, HMM-GMMs are trained on top features generated from a relatively narrow intermediate layer within the ANN, known as a bottleneck layer.

3.1.3 Language model

The prior probability of the word sequence W, $P(W|\Theta_l)$ in Equation 3.1 represents the language syntax and semantics. As shown by its expression, it is independent of the acoustic observations. Thus, its parameters Θ_l, can be estimated from large textual information sources such as newspapers, journals and web content. For a given sentence, W, containing K words, the joint probability for the word sequence W is given as a product of conditional probabilities:

$$\begin{aligned} P(w_1, w_2, ..., w_K) &= P(w_1)P(w_2|w_1)P(w_3|w_1, w_2)...P(w_K|w_1, ..., w_{K-1}) \\ &= \prod_{i=1}^{K} P(w_i|w_1, ..., w_{i-1}) \end{aligned} \tag{3.3}$$

The most common type of language model used for speech recognition tasks is the n-gram LM (Bahl et al., 1983) which is based on the assumption that the probability of a word w_i depends on only the $n-1$ preceding words. Thus, the joint probability in Equation 3.3 becomes:

$$\begin{aligned} P(w_1, w_2, ..., w_K) &= \prod_{i=1}^{K} P(w_i|w_1, ..., w_{i-1}) \\ &\approx \prod_{i=1}^{K} P(w_i|w_{i-n+1}, ..., w_{i-1}) \end{aligned} \tag{3.4}$$

An n-gram is a sequence of n words and n is known as the order or depth of the LM. A trigram, bigram and unigram LMs are referred to LM of orders 3, 2 and 1 respectively.

The model parameters are estimated from counting the occurrences of word sequences

in a large volume of text. This is known as the maximum likelihood estimate, which is given by:

$$P(w_i|w_{i-1}, w_{i-2}) = \frac{\text{Count}(w_i, w_{i-1}, w_{i-2})}{\text{Count}(w_{i-1}, w_{i-2})} \tag{3.5}$$

However, some of the word sequences cannot be found in the training text, which makes their probabilities unknown and will be assigned zero probability. This presents a problem for an ASR because a word sequence with zero probability indicates that it would never be recognised. This has been addressed by smoothing methods such as discounting and back off (Jelinek, 1997). For example, modified Kneser-Ney smoothing (Chen and Goodman, 1996) applies several discount parameters where it gives greater significance to low-order LM only if the count of high-order n-gram is small. The smoothed word estimates for a given word, w_i, given its history is computed as:

$$P_{\text{KN}}(w_i|h) = \frac{\max(\text{Count}(h, w_i) - D, 0)}{\sum_{w'} \text{Count}(hw')} + \alpha(h)\ P_{\text{KN}}(w|h') \tag{3.6}$$

where h is the sequence of $n-1$ words preceding w_i and h' is the sequence of $n-2$ words preceding w_i. D is the discount coefficient and its depending on the observed counts and α is a normalisation constant to be applied for a lower order LM.

In practice, a background LM estimated on a large amount of general training data is adapted with another LM estimated from a smaller in-domain training data. Linear interpolation (Jelinek, 1980) can be employed for such adaptation. In this approach, a word probability given its history, $P(w_i|h)$, is given as a weighted average of the probabilities computed by each LM. Formally:

$$P(w_i|h) = (1-\lambda)\ P_{\text{back}}(w_i|h) + \lambda\ P_{\text{indomain}}(w_i|h) \tag{3.7}$$

where $P_{\text{back}}(w_i|h)$ and $P_{\text{indomain}}(w_i|h)$ are the assigned probabilities by the background LM and in-domain LM, respectively. λ is the interpolation weight, where $0 \leq \lambda \leq 1$, and it is chosen based on tuning the LM performance over a held-out in-domain development set.

3.1.4 Search algorithm

In order to solve the problem stated in Equation 3.1, the search component, also called the decoder, combines different knowledge resources (i.e. acoustic, pronunciation and language models) to find the most likely word sequence $\hat{W}$ (hypothesis transcription) given an acoustic observation sequence, X, in time-synchronous manner. Decoding can be efficiently undertaken using dynamic programming (Bellman, 1957) and Viterbi approximation (For-

ney, 1973) by expanding words, phonemes and HMM states and accumulating scores from different components to find the path with the highest likelihood. When the end of the acoustic observation sequence is reached, the path with the highest likelihood is traced back, yielding the best hypothesis transcription (1-best output). Due to the huge search space, pruning methods, e.g. beam pruning (Lowerre, 1976), are employed to discard unlikely branches and reduce the computational cost.

The previously described strategy is known as a single-pass decoding. However, some models and adaptation techniques cannot be applied directly to the original search space because of computational complexity. In such cases, decoding is performed in a multiple-pass fashion. In the first pass, decoding is used to generate multiple best hypotheses, instead of only one, using simple models. These hypotheses can be stored in the form of lattices or N-best list. A lattice is a directed acyclic graph representation and N-best list is a ranked list of the top N hypotheses. In the following pass(es), more complex and advanced models can be used to recompute the knowledge resources scores in a much smaller search space and subsequently generates the 1-best hypothesis transcription. This strategy is referred to as multi-pass decoding (Schwartz and Austin, 1991; Aubert and Ney, 1995; Richardson et al., 1995) or lattice/N-best list re-scoring in oppose to the formerly described single-pass decoding.

3.1.5 System evaluation

The resulting 1-best hypothesis transcription is compared to a reference transcription to evaluate the recognition performance. Word error rate (WER) is commonly used as metric for such evaluation. WER represents the minimum number of edit operations to transform the hypothesis into its reference transcription, and is computed as:

$$\text{WER} = \frac{S + D + I}{N} \times 100 \tag{3.8}$$

where S is the number of substitutions, D is the number of deletions, I is the number of insertions and N is the total number of words in the reference.

Other metrics that depend on the unit used to transcribe the reference are phoneme error rate (PER) and grapheme error rate (GER). PER (or GER) is computed by comparing the acoustic (or graphemic) units of the hypothesis transcriptions against those in the reference transcription using an equivalent same formula to Equation 3.8.

Language models can be evaluated independently from a full ASR system by computing perplexities on testing data (Bahl et al., 1977). Perplexity (ppl) is an information-theoretic metric that is the inverse of the geometric average probability assigned to each word in a testing dataset by the model. It can be interpreted as word average branching factor, that

can be computed as follows.

$$\begin{aligned} ppl &= (\prod_{i=1}^{N} P(w_i|w_1 w_2 \dots w_{i-1}))^{\frac{1}{N}}, \\ &\simeq (\prod_{i=1}^{N} P(w_i|w_{i-n} \dots w_{i-1}))^{\frac{1}{N}}, \\ &= \exp(-\frac{1}{N}\sum_{i=1}^{N} \log(P(w_i|w_{i-n} \dots w_{i-1}))), \end{aligned} \tag{3.9}$$

where N is the number of words in the testing dataset. Lower values indicate better prediction ability of the model. Perplexity can be viewed as a measurement of how close a given model is to the true model presented by the testing dataset. It also estimates the complexity of a given language (Brown et al., 1992). In order to compare different language models, perplexities must be computed on the same testing data, using language models which are estimated over the same vocabulary list.

3.2 State-of-the-art CA ASR systems

The first work in transcribing Arabic dialects was reported in a CallHome task within 1996/97 NIST benchmark evaluations framework. After that, more data was provided for the NIST evaluations (2003 and 2004); mainly in Egyptian and Levantine dialects. More data, especially in Iraqi dialect, was provided as resources for two main research programs: the Global Autonomous Language Exploitation (GALE) program (Olive et al., 2011) and the Spoken Language Communication and Translation System for Tactical use (TRANSTAC) program. GALE is a DARPA program to develop and apply computer software technologies to extract useful information from huge volumes of speech and text carried out between 2006 and 2009. TRANSTAC is a similar program which aimed to help an average US soldier to communicate with a person who cannot speak English, using a portable bidirectional translator. Unfortunately, these valuable resources are generally unavailable for public research, hence several researchers without access to such resources had defined their internal test sets. Table 3.1 listed the specifications of test sets used in the recent systems for CA ASR.

Most of these systems use a combination of ASR technologies to achieve the best recognition performance. Table 3.2 and Table 3.3 summarise these systems. The test set that is used for the evaluation is listed in the second column while the features used are listed in the third column and the combination of multiple features is denoted by a + sign. The notation for the forth column is in the format of APP:MODEL{DATA}+ADAPT,

Table 3.1: Description of test sets mentioned in the literature for colloquial Arabic ASR task. CTS: Conversational telephone speech, BC: Broadcast conversation, DI: Dialect identification.

Testset	Description	Task	Dialect	Size
eval'97	NIST RT-03 Evaluation	CTS	ECA	1.0 h
eval'03	NIST RT-03 Evaluation	CTS	ECA	1.0 h
ulm'10	ULM university recording	Read speech	ECA	385 utts
google11eg	Google voice search test set	Voice search	ECA	12.4 h
eval'04	NIST RT-04 Evaluation	CTS	LCA	1.5 h
galebclev	GALE broadcast conversation selected by DI	BC	LCA	4.0 h
ibm'06	IBM S2S translation	Role-play speech	ICA	1.5 h
ibm'07	IBM S2S translation	Role-play speech	ICA	
transtac'n08	TRANSTAC S2S translation (Nov'08)	Role-play speech	ICA	
transtac'j09	TRANSTAC S2S translation (Jun'09)	Role-play speech	ICA	16.0 h

where APP is the training approach used, such as maximum likelihood (ML), MODEL is either undiacritised grapheme (G), diacritised grapheme (D) or phoneme (P), {DATA} corresponds to the training data's dialect and +ADAPT is the adaptation used in the process, such as maximum likelihood linear regression (MLLR). The language model used is described in the fifth column and finally the reported WER is shown in the last column. It should be noted that both the system structure and decoding procedure, which are usually complex and composed of multiple components or passes, are not described in these tables.

The main tasks performed by these systems can be categorised into five classes as shown in Table 3.1, which are conversational telephone speech (CTS), broadcast conversation (BC), role-play speech, voice search and read speech. CTS is informal spontaneous speech which was taking place over a telephone channel between two speakers while BC can be between more than one participant and was in a more formal settings, hence, a large proportion of BC speech found to be spoken in MSA. For obtaining a role-play speech, participants were asked to act a certain role in a very specific scenario but without a scripted speech. Google's voice search speech is a recitation for a collection of common search queries but using conversational speech which is similar to read speech while participants were provided with scripted CA speech to read.

Generally, performance for a CTS ASR task is worse than that obtained for other speaking styles, such as voice search or read utterances. This can be accounted the additional challenges which posed by the conversational speech, such as disfluencies and high variability. Using adaptation techniques, such as maximum likelihood linear regression (MLLR) and speaker adaptive training (SAT), reduce the WER in most ASR systems. In the following sections, some of the systems mentioned in Table 3.2 and Table 3.3 are discussed in details either in terms of their language models (Section 3.3) or acoustic models (Section 3.4) or both.

3.3 Colloquial Arabic language modelling

The rich and complex morphology of the Arabic language, outlined previously in Section 2.5, causes a massive increase in lexicon size (Kirchhoff et al., 2003). As a result, the rate of unseen words in the training data, the out-of-vocabulary (OOV) rate, increases dramatically, thus reducing the reliability of the seen data against the unseen.

Many studies have addressed the vocabulary problem by using morphological and syntactic information with or instead of words. A word is decomposed into its morphological units, or morphemes, and each morpheme is assigned a part-of-speech (POS) tag. Table 3.4 lists an example of such analysis. Kirchhoff et al. (2003; 2006) suggested using

Table 3.2: Comparison of state-of-the-art ECA ASR systems performance. Features:[perceptual linear predictive (PLP), Mel frequency cepstral coefficients (MFCC), [heteroscedastic] linear discriminant analysis ([H]LDA), vocal tract length normalisation (VTLN), maximum likelihood linear transformation (MLLT)]; Acoustic modelling:[APP: maximum likelihood (ML), minimum phone frame error (MPFE); MODEL: grapheme (G), diacritised grapheme (D); DATA: modern standard Arabic (MSA); ADAPT: maximum likelihood linear regression (MLLR), conditional MLLR (CMLLR), speaker adaptive training (SAT), maximum a posteriori (MAP)]; Pronunciation probabilities (PP); Language modelling:[factored language model (FLM); word-based (word); Gaussian mixture (GMLM); Tied-mixture (TMLM)]

Publication	Task	Features	Acoustic Modelling APP:MODEL{DATA} +ADAPT	Language Modelling	WER
Kirchhoff et al. (2006)	eval'97	MFCC	MMI: D{ECA} +MLLR{ECA} +SAT	FLM word	56.6
Creutz et al. (2007a)		+HLDA		3-gram word	58.2
				3-gram morph	59.9
Kirchhoff and Vergyri (2004)	eval'03	MFCC	ML: D{ECA} +MAP{ECA} +MLLR{ECA}	2-gram word	42.7
			ML: D{ECA,MSA} +MAP{ECA} +MLLR{ECA}		43.0
			ML: D{ECA} +MAP{ECA} +MLLR{ECA} +CMLLR{ECA}	3-gram word	42.7
			ML: D{ECA,MSA} +MAP{ECA} +MLLR{ECA} +CMLLR{ECA}		42.6
Vergyri and Kirchhoff (2004)		MFCC +VTLN	ML: D{ECA}	3-gram diac words {ECA}	42.7
				3-gram diac words {ECA+MSA}	42.2
Elmahdy et al. (2010)	ulm'10	—	ML: D{ECA}	2-gram word	35.1
			ML: D{MSA}		48.4
			ML: D{MSA} +MLLR{ECA} +MAP{ECA}		29.1
			ML: G{ECA}		42.2
			ML: G{MSA}		64.8
			ML: G{MSA} +MLLR{ECA} +MAP{ECA}		36.1
Biadsy et al. (2012)	google11eg	PLP+LDA	BMMI:P{ECA} +CMLLR{ECA}	5-gram word	24.6
				5-gram diac word	29.3

Table 3.3: Comparison of state-of-the-art LCA and ICA ASR systems performance. Features:[perceptual linear predictive (PLP), Mel frequency cepstral coefficients (MFCC), [heteroscedastic] linear discriminant analysis ([H]LDA), vocal tract length normalisation (VTLN), maximum likelihood linear transformation (MLLT)]; Acoustic modelling:[APP: maximum likelihood (ML), minimum phone frame error (MPFE); MODEL: grapheme (G), diacritised grapheme (D); DATA: modern standard Arabic (MSA); ADAPT: maximum likelihood linear regression (MLLR), conditional MLLR (CMLLR), speaker adaptive training (SAT), maximum a posteriori (MAP)]; Pronunciation probabilities (PP); Language modelling:[factored language model (FLM); word-based (word); Gaussian mixture (GMLM); Tied-mixture (TMLM)]

Publication	Task	Features	Acoustic Modelling APP:MODEL{DATA} +ADAPT	Language Modelling	WER
Vergyri et al. (2005)	eval'04	PLP+MFCC +HLDA	MPFE: G{LCA} +SAT +adaptation	FLM word	46.9
		+VTLN	MPFE: G{LCA} +SAT +adaptation +autoVowel		46.5
Stolcke et al. (2006)		MFCC +HLDA	MPFE: G{LCA}	FLM word	47.3
			MPFE: G{LCA} +genericVowel		46.9
			MPFE: G{LCA} +autoVowel		46.5
Soltau et al. (2011)	galebclev	PLP+LDA +VTLN	ML: G{MSA} +BMLLR{LCA} +fBMLLR{LCA}	4-gram word	39.7
			ML: P{MSA} +BMLLR{LCA} +fBMLLR{LCA}		40.8
Afify et al. (2006)	ibm'06	MFCC+LDA +MLLT	MPE:G{ICA}	3-gram word	36.3
				3-gram morph	32.1
Afify et al. (2007)				GMLM word	33.1
Sarikaya et al. (2009)	ibm'07		MPFE:G{ICA} +fMLLR{ICA}	3-gram word	32.9
				TMLM word	32.5
Tsakalidis et al. (2009)	transtac'j07	PLP	ML:G{ICA}	3-gram word	39.7
			ML:D-M{ICA}		33.8
			ML:D-M{ICA} + 2gPP		33.4
			ML:D-B{ICA}		41.5
			ML:D-B{ICA} + 2gPP		40.8
Al-Haj et al. (2009)	transtac'n08	MFCC+LDA	D-M{ICA}	3-gram word	35.7
			D-M{ICA} + PP		34.8

morphological and word-level information in the text interchangeably in a multi-stream model, namely a factored language model (FLM) (Bilmes and Kirchhoff, 2003). In the FLM, a word is viewed as a vector of k factors, where a single factor could be any of the word characteristics such as its stem or morphological class. The strength of FLM is when it backs off to other factors in case a word n-gram sequence is not found in the training data. In their experiment, morphemes were derived from a manually-annotated lexicon for Egyptian colloquial Arabic (ECA) by Kilany et al. (2002), and the word sequence estimates were used to re-score word-based lattices. FLMs outperformed a standard word-based LM slightly in an ECA recognition task by 0.7% absolute. However, less improvement was observed in the absence of true morphological information. Vergyri et al. (2005) used Buckwalter Arabic morphological analyser (BAMA) (Buckwalter, 2002; 2004a) to generate morphological profiles for Levantine colloquial Arabic (LCA) data. If a word could not be analysed by BAMA, an automatic analyser was used. That yielded 0.1% absolute reduction in WER for an LCA recognition tasks. As shown in Table 3.4, the meaning of an isolated Arabic undiacritised word is uncertain and is only disambiguated through diacritics. This motivated some researchers to include diacritics in language modelling, whether at the word-level (Vergyri and Kirchhoff, 2004) or morpheme-level (El-Desoky et al., 2009). The objective was to reduce word ambiguity and ultimately improve LM perplexity. The gain in CA ASR performance was less than that reported in MSA experiments, due to the limited data available where diacritisation breaks down a word probability into multiple weaker diacritised word estimations.

The improvement in MSA recognition performance was more prominent when the decomposition process was restricted by defining the set of affixes to be decomposed and excluding frequent words in the vocabulary from decomposition (Xiang et al., 2006; Lamel et al., 2008; Nguyen et al., 2009; El-Desoky et al., 2009; Diehl et al., 2009b; Kuo et al., 2010; El-Desoky et al., 2010), especially for medium size vocabulary (fewer than 64 words) (Choueiter et al., 2006). The WER reduction ranges between 1.9-13.3% and correlates with the quality of the morphological analysis incorporated in the LM training. These outcomes and the inability of MSA morphological analysers to produce precise information for CA utterances led to the use of lightly supervised approaches for morphological decomposition (rather than relying on MSA tools). For example, Afify et al. (2006) implemented a morpheme-based language model for Iraqi collquial Arabic (ICA). Here, words were decomposed by applying blind segmentation over predefined affixes. The resulting word was then tested against a minimum stem length and other linguistic constraints. Performance was increased significantly by 4.3% absolute in the ICA recognition task. In addition, they found that adding contextual information by interpolating their model with a word-based language model improved the ASR performance by an additional 1% abso-

Table 3.4: An example of a morphological profile of the word "وسيدرسونها".

undiacritised	**diacritised / translation**	**morphes**	**tags**	**translation**
wsydrswnhA	wasayadrusuwnahA (and they will learn it)	wa+	conjunction	and
		sa+	verb prefix (future)	will
		ya+	verb prefix (imperfect)	–
		drusu	verb FormI imperfect active	learn
		+wna	Pronoun indicative 3rd person masculine plural	they
		+hA	Pronoun direct object 3rd person feminine singular	it
	wasayudar~isuwnahaA (and they will teach it)	wa+	conjunction	and
		sa+	verb prefix (future)	will
		yu+	verb prefix (imperfect)	–
		dar~isu	verb FormII imperfect active	teach
		+wna	Pronoun indicative 3rd person masculine plural	they
		+hA	Pronoun direct object 3rd person feminine singular	it

lute. In contrast, Creutz et al. (2007b) decomposed the words in the training data using a language-independent unsupervised statistical toolkit, Morfessor (Creutz and Lagus, 2005), which split words into segments with less linguistic sense, known as morphs, based on a minimum description length (MDL) criterion and classified these segments into prefixes, stems and suffixes using maximum a posteriori (MAP) estimation. Using a morph-based LM, estimated from ECA decomposed data, did not yield any improvement in recognition performance in comparison to word-based LM. The authors accounted for this by stating that the inflectional morphology of ECA cannot be captured by Morfessor.

A different approach was devised by Sarikaya et al. (2009). The authors deployed a maximum entropy (MaxEnt) based language model (Rosenfeld, 1996) with morphological and lexical features, as first proposed in previous work (Afify et al., 2007). Their work did not reduce the WER significantly. However, when their LM was interpolated with a standard trigram LM, WER improved significantly, by 1% absolute.

Limited efforts have been reported in terms of using MSA resources to train colloquial Arabic LM, whether by adding MSA text to the training corpus (Kirchhoff et al., 2002b), or by searching for parallel colloquial Arabic text in the MSA corpus (Chiang et al., 2005). In these studies, either no WER reduction was reported, or there was a small improvement.

3.4 Colloquial Arabic acoustic and pronunciation modelling

One of the important design decisions for ASR is choosing the speech units that each HMM will represent. Two models are commonly used for acoustic modelling: phoneme-based models and grapheme-based models. In the former, each model represents only one phoneme; therefore, a lexicon that maps an orthographic word, or baseform, to its fully phonetic description is required. Conversely, the grapheme-based models are based on the orthographic presentation of the data. Each model represents a letter that may be realised with multiple phones depending on the mapping between a grapheme to phoneme in the language.

Generally, Arabic transcriptions lack short vowels and gemination information. Many studies of Arabic speech recognition have used grapheme-based modelling such as those of Afify et al. (2006), Choueiter et al. (2006) and Billa et al. (2002). When undiacritised graphemic representation is employed, each grapheme can represent up to eight phonemic values: the phoneme associated with that letter alone; the geminated version of the phoneme; the phoneme followed by each of the three short vowels; and the geminated phoneme followed by each of the three short vowels. Figure 3.3 shows the phonetic values of the grapheme "ب".

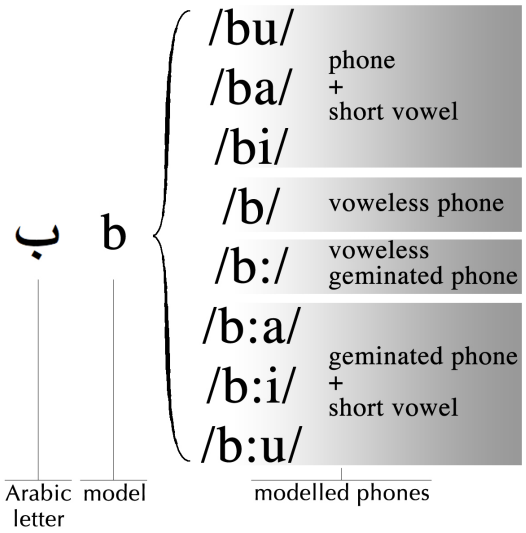


Figure 3.3: An example of modelling an Arabic letter "ب" as a grapheme model and all phonetic values that should be modeled within.

Kirchhoff et al. (2003) found that using a more detailed graphemic representation through explicit inclusion of diacritics outperformed systems using undiacritised graphemes as acoustic models in ECA ASR tasks. Manually fully diacritised transcriptions reduced the WER by 6.6% relative, on the NIST 1996 Arabic evaluation set. Similarly Afify et al. (2005) showed that modelling short vowels explicitly improved recognition performance in an MSA broadcast news ASR task, even if the recogniser outcomes did not include diacritics. These findings correspond to the fact that the Arabic language has a shallow relationship between the fully diacritised orthographic form and its corresponding phonemic representation. However, short vowels are optionally added and generally omitted from the written form. Thus, predicting these missing vowels is crucial when modelling them in the pronunciation.

As mentioned in Section 2.5, most of the words in MSA are results of applying a derivational morphological process. In this process, a vowelic template is applied to a root, for instance, the word "yadrusu" (he is studying) is obtained from applying the template "yaC_1C_2uC_3u" over the root "d r s" (learning concept) and the word "yudrasu" (it is learnt) from applying the template "yuC_1C_2aC_3u" over the same root. Since these templates are well-defined in the language, the vowelised form can be derived from its morphological components and templates. For instance, there are five phonemic vowelised forms for the word "ydrs", e.g. "yadrusu" and "yudrasu"; however, if the word was suffixed with a "h" to be "ydrsh", there is only two possible vowelised forms which can be further reduced to one vowelised form when the surrounding context is considered. Therefore, most of the state-of-the-art systems use a morphological analyser, such as BAMA. BAMA uses a

context-free rule-based approach with the support of an extensive dictionary of stems and affixes in MSA. BAMA has been the basic building block for subsequent tools, such as the morphological analysis and disambiguation for Arabic (MADA) tool (Habash and Rambow, 2005). MADA disambiguates word candidates derived from BAMA using support vector machine classifiers for individual morphological features which are pre-trained on fully annotated MSA data, and which incorporates likelihood scores based on context.

However, these morphological analysis tools are able to handle only MSA words, but not names, dialectical or foreign words, which are considered to be out-of-vocabulary words. In these cases, pronunciation can be generated either manually or automatically, where the latter can use knowledge-based or data-driven techniques. Since manual diacritising is an expensive task, Vergyri and Kirchhoff (2004) tried to automatically diacritise Egyptian colloquial Arabic training text. The authors employed BAMA to generate all the possible diacritised forms with the aim of constructing a pronunciation network. This network was then scored using a statistical tagger, that was trained on sequences of morphological tags derived from MSA text. An acoustic model was then trained on the automatically diacritised text, and achieved modest but significant improvements in the ASR task in comparison to a grapheme-based model trained on undiacritised text. The authors accounted for this modest improvement by the inclusion of more vowelised variants of the words, which increased the search space during decoding. Al-Haj et al. (2009) overcame this issue by including weighted pronunciation probabilities in computing the word sequence probability. First, they used a CART-based statistical letter-to-sound model trained on a manual fully diacritised Iraqi dictionary. Then, multiple pronunciations were derived from the augmented dictionary and their probabilities were estimated based on the raw frequency in the training data after using forced alignment. This approach reduced WER in the ICA ASR task in comparison to a standard unweighted diacritised grapheme-based system, by 7.5% relative.

Since Arabic readers rely on context for choosing the appropriate pronunciation, Tsakalidis et al. (2009) incorporated contextual information using a pronunciation model. The pronunciation model mapped the acoustic models' sequences to their corresponding words, and was more effective if the word had more than one pronunciation. A bigram pronunciation model was used and smoothed with context-independent pronunciation probabilities to overcome data sparsity. The vowelised pronunciation dictionary contains 57,000 words with an average of 4.8 pronunciations per word. BAMA was used to generate all possible vowelisation variants for 89.5% of the words with the rest of the words being derived from a manually vowelised dictionary. Their approach reduced WER in phoneme-based Iraqi Arabic by 0.7% absolute. However, they expected it to outperform a standard grapheme-based system in the presence of more training resources, since their results were worse by

1.1% absolute.

Owing to the enormous difference between MSA and CA, they can be considered as totally different languages. Based on this idea, Kirchhoff and Vergyri (2005) proposed a cross-lingual acoustic model. In their approach, they used pooled MSA and ECA resources as training data which led to a modest improvement in recognition performance by 3% relative. However, Elmahdy et al. (2010) accounted this minor improvement to the use of an ECA corpus that was small in relation to the MSA data, which made the model biased toward the latter, their study used a combination of MLLR and MAP adaptation by which a 10.2% relative reduction in WER of grapheme-based ASR in ECA ASR task was observed.

3.5 Summary

This chapter provides an introduction to the fundamentals of ASR systems along with an overview of the current state-of-the-art CA ASR systems and discussed in further details some of these systems' novel technologies in language, acoustic and pronunciation modellings.

As word decomposition improved the LM perplexity, no proper comparison has been carried out between different methods of word decomposition, language model quality and recognition performance in the CA ASR task. Moreover, in all previously cited work, there was no reported attempt to pre-process the MSA volume text before using it as an additional source in language model training for colloquial Arabic ASR.

Most of the aforementioned work in predicting the missing short vowels for a given Arabic word relies on the morphological analysis of the word, mostly by using BAMA, or adding some local contextual information in the prediction process. In addition, there have been very few attempts to address the actual mapping between the graphemic representation and its associated phonemic realisation in MSA generally and in colloquial Arabic specifically.

This thesis investigates some of these issues and research gaps within modelling for CA ASR. This work starts with an investigation on the linguistic issues which is followed by an investigation and a thorough discussion of the acoustic and pronunciation issues.

Chapter 4

Sub-lexical unit language modelling in colloquial Arabic

Contents

As discussed previously in Chapter 2, Arabic is a morphologically rich language where a word is generated by applying a combination of morphological processes. New words in Arabic can be easily created either by applying a template to a root (derivation), or by concatenating articles and prepositions to a word without changing the original word (agglutination) or by concatenating pronouns and applying changes to the original word

(inflection). As a result, the number of unique words in a given amount of text (vocabulary size) for Arabic is significantly higher than in other languages that lack such a rich morphology, such as English. For all languages, as the amount of a given text increases, so does the vocabulary size at a certain rate (known as vocabulary growth rate). This rate is considerably larger for languages with rich morphology.

Developing an ASR for morphologically rich languages should address two main issues which are raised by the nature of the language. First, the chosen vocabulary will have a limited coverage of the data, considering the aforementioned vocabulary growth, because modern ASR systems allow a dictionary with a limited vocabulary size (even when vocabulary size is large). As a consequence, high out-of-vocabulary (OOV) rates are observed for such languages. Second, due to the high number of inflected and agglutinated words in Arabic, the average frequency of a word in a given text is lower. For example, the sentence "mdrsy wmdrs Sdyqty hw nfs Almdrs" (my $\text{teacher}_{\text{m}}$[1] and my friend_{f}'s $\text{teacher}_{\text{m}}$ is the same $\text{teacher}_{\text{m}}$) has three inflected and agglutinated words sharing the same stem (stem is underlined): "mdrsy" (my $\text{teacher}_{\text{m}}$), "wmdrs" (and $\text{teacher}_{\text{m}}$), "Almdrs" (the $\text{teacher}_{\text{m}}$) are considered as three different words instead of three occurrences of what in English translation would be the same word. This causes estimated probabilities in the language model to be unreliable and generates high perplexities.

It has been reported in the literature that using word decomposition based on morphological analysis limits vocabulary growth and increases the average frequency for word particles which, in its turn, improves language model probability estimates. Since CA morphological structure is inherited from MSA, many researchers analysed CA words using MSA-based morphological analysis. In the absence of CA-based morphological analysers, word decomposition can be performed based on other non-linguistic measures. This chapter provides a detailed investigation of word decomposition in CA and its impact on the OOV rates and overall perplexities.

This chapter is organised as follows: Section 4.1 discusses the literature on using word decomposition in language modelling frameworks for morphologically rich languages in general, and CA in particular. This is followed by an overview of the investigation framework, which can be divided into two parts: First, word decomposition (Section 4.2) to be applied in CA text data; second, estimating language models probabilities based on the resulting decomposed text (Section 4.3). As an attempt to address the data sparsity issue, classes can be derived from the resulting sub-lexical units to be used as additional information in the prediction task (Section 4.4). Empirical results of this investigation are presented and discussed in Section 4.5. Finally, conclusions are drawn and the chapter is summarised in

[1] The subscript indicates gender of a word, i.e. $\text{teacher}_{\text{m}}$ is a male teacher while $\text{teacher}_{\text{f}}$ is a female teacher.

Section 4.6.

4.1 Related research

As discussed previously in Section 2.5, Arabic has a complex morphological system. A word can be composed of several sub-lexical units where each holds a morphological and syntactic meaning. Morphological analysis is the process of parsing a given word into its morphemes, identifying the part-of-speech (POS) for each morpheme and establishing the relationships between morphemes. For example, Table 4.1 lists the morphological analysis for the Arabic word "drshm" (their lesson / he studied them / he taught them / teach them) which can be decomposed into two sub-lexical units. These sub-lexical units are called morphemes because they are extracted based on morphological analysis; otherwise, they are called morphs[2]. As shown in Table 4.1, the outcome of the morphological analysis depends on the chosen diacritisation of the word. Because of that, Arabic morphological analysers find the legitimate diacritisation variants for the given word as part of the analysis process. Such morphological complexity appears in other languages to different degrees, such as Turkish, Czech, German, Finnish, Korean and Japanese.

In the literature discussed below, morphological analysis in language modelling has been employed using two different strategies. In the first strategy, morphological analysers were used for word decomposition solely by using the resulting morphemes as training material to estimate a *morpheme-based LM*. For the second strategy, the resulting morphological tags from the morphological analysis task were used as auxiliary information or additional input stream for training more complex language models. Frequently, the resulting LMs (morpheme-based LMs or complex LMs) were used in a multi-pass recognition framework to re-score an N-best list or lattices initially generated with a weaker LM (as discussed in Section 3.1.4).

Geutner (1995) was the first to use morphological analysis in order to decrease the OOV rate for a German ASR system. Using rule-based morphological decomposition, the author accomplished a relative reduction of 28% in vocabulary size but with a relative degradation of 1% in WER. Geutner pointed out that the short morphemes that were acoustically similar were the source of this degradation. Rather than decomposing all words within the vocabulary, Berton et al. (1996) used a more selective approach and included only infrequent words in the decomposition (those with fewer than 15 occurrences in the training data). This yielded less than 1% relative WER improvement in a German recognition task. Other attempts of using morpheme-based LMs and removing short morphemes by merging

[2]A morph is a lexical realisation of a sub-word unit while morpheme is the minimum morphologically meaningful sub-word unit.

Table 4.1: An example of a morphological profile of the undiacritised word "drshm" for each valid diacritised variant (diacritics are underlined). Resulting morphemes are shown in third column. Morphological tags for each morpheme are shown in the fourth column along with the English translation in the fifth column.

undiacritised	diacritised/translation	morphemes	tags	translation
drshm	darsahum	darsa	singular noun	lesson
	(their lesson)	+hum	possessive pronoun 3rd person masculine plural	their
	darsihum	darsi	singular noun	lesson
	(their lesson)	+hum	possessive pronoun 3rd person masculine plural	their
	darsuhum	darsu	singular noun	lesson
	(their lesson)	+hum	possessive pronoun 3rd person masculine plural	their
	darasahum	darasa	verb FormI perfect active masculine singular	he studied
	(he studied them)	+hum	direct pronoun object 3rd person masculine plural	them
	dar~isahum	dar~isa	verb FormII imperative masculine singular	teach
	(teach them)	+hum	direct pronoun object 3rd person masculine plural	them
	dar~asahum	dar~asa	verb FormII perfect active masculine singular	he taught
	(he taught them)	+hum	direct pronoun object 3rd person masculine plural	them

them based on perplexity improvement followed in Korean (Kiecza et al., 1999; Kwon, 2000), German (Adda-Decker and Adda, 2000; Larson et al., 2000), Turkish (Carki et al., 2000), Japanese (Kawahara et al., 2000), Czech (Byrne et al., 2000; Ircing et al., 2001) and Finnish (Kneissler and Klakow, 2001). Similar outcomes were observed: significant reduction in OOV rate but with limited improvement in recognition performance.

Regardless of the earlier work on Arabic morphological analysers, Beesley (1996), it was some time until attempts started to emerge that use morphological analysis and decomposition in Arabic ASR (both MSA and CA). This was due to the complexity of the task and the extensive development time needed to implement the earlier proposed algorithms (Darwish, 2002). In the meantime, other approaches were adopted for morphological decomposition where minimal linguistic knowledge (or supervision) is needed. Several researchers built morpheme-based LMs where words were segmented into (prefix)-stem-(suffix) sequences by restricting segmentation to a pre-defined affixes list. Such decomposition is referred to as semi-supervised morphological decomposition. Choueiter et al. (2006) used a morpheme generator for decomposition, based on an algorithm proposed earlier by Lee et al. (2003). They used a weighted finite state transducer (WFST) accepter to reject illegal sequences of morphemes. Their method yieldied a 0.7% relative reduction in WER for large vocabulary (more than 64k words) and 7.5% relative reduction in WER for medium vocabulary (fewer than 64k words) of an MSA broadcast news ASR task. Additional constraints were introduced in morphological decomposition by Xiang et al. (2006). Decomposition is applied only for infrequent words as long as the resulting stems from the morphological analysis were at least two letters in length and existed in a given dictionary. Using such constraints reduced the OOV rate by 29% relative and showed a 3.7% relative improvement in recognition performance of an MSA broadcast news ASR task.

With the introduction of BAMA (Buckwalter, 2002; 2004b) and its subsequent development MADA (Habash and Rambow, 2005), more work was performed in the CA LM. BAMA uses a context-free rule-based analysis with an aid of a morphological lexicon of MSA stems and affixes. Initially, BAMA segments a word greedily into three components: prefix, stem and suffix with prefix and suffix are possible to be null and stem with at least one letter long. For each segmentation, a lookup in the morphological tables of stems and affixes is performed and only a segmentation with all its components are existed in the lexicons is passed. Finally, a segmentation is considered a possible analysis only if its morphological tag sequence is allowed based on BAMA's compatibility tables. The outcome of BAMA is a list of all possible analyses for a given word. As an extension to BAMA, MADA uses trained SVMs to choose one candidate analysis result according to the context. Several studies used BAMA and MADA morphological analysis for word decomposition while controlling segmentation by adding constraints to the process. Afify

et al. (2006) applied the constraints defined by Xiang et al. (2006) along with a verification constraint using BAMA to morphologically decompose ICA transcriptions. The authors obtained a 66% relative reduction in OOV rate and a 8.7-10% relative reduction in WER for an ICA recognition task. Similarly, Lamel et al. (2008) and El-Desoky et al. (2009) applied the same strategy on an MSA broadcast news task with large vocabulary size (of 200k words) and gained 1.9% and 3% relative reduction in WER respectively using different sets of training data. MADA normalises the resulting morphemes to their original forms. For example, "Sdyqty" (my friend$_{\text{f}}$) becomes "Sdyqp +y" instead of "Sdyqt +y". In the latter, direct merging is straight forward to map sequence of morphemes back to sequence of words without any need for post-processing. Diehl et al. (2009a) proposed an SMT-based method to back-map morphemes to words. In an MSA broadcast news ASR task, they achieved at least 5.8% relative WER improvement using their mapping method, compared to less than 2% relative WER without. Other resources have been used apart from BAMA and MADA, such as Sakhr morphological analyser. That was used by Nguyen et al. (2009) for word decomposition along with a rule-based segmentation with an affixes set, achieving a 6.9% relative reduction in WER in an MSA broadcast news ASR task.

Besides word decomposition, morphological analysis was used as an additional input stream within more complex language modelling frameworks. Ghaoui et al. (2005) derived classes from morphological tagging in order to train a class LM (Brown et al., 1992). In their work, morphological analysis using a rule-based algorithm and predefined set of affixes achieved a relative reduction of 1.6% in the perplexity of MSA newswire data[3] over a standard trigram word-based LM. Kirchhoff et al. (2006) built a morpheme-based LM for ECA using a combination of stream LMs and class LMs to obtain almost 2% relative improvement on an ECA recognition task. In stream and class LMs, input streams are treated independently without any interaction during backing off to a lower order n-gram. In contrast, factored LMs (FLMs) use a more complex back off strategy, known as generalised parallel backoff (Bilmes and Kirchhoff, 2003), where the model backs off to another stream if that has a better estimate. Using FLMs with morphological information gave a significant reduction in WER on MSA broadcast news and conversations recognition tasks of up to 4% WER relative (El-Desoky et al., 2010; 2012), and up to 2% WER relative on ECA and LCA CTS recognition tasks (Kirchhoff et al., 2003; Vergyri and Kirchhoff, 2004; Vergyri et al., 2005; Kirchhoff et al., 2006). Kuo et al. (2009; 2010) compared different levels of expert-provided morphological and syntactic information that could be included in building a neural network LM. A 1.1% relative reduction in WER of an MSA broadcast news was achieved when only POS information was incorporated within the model. A further improvement of up to a 5.5% relative WER reduction was obtained when syntactic

[3]Newswire is a digital form of newspaper which usually delivered over the Internet.

information was included as well.

Sarikaya et al. (2007) used an alternative approach based on a continuous representation of the words using a joint morphological-lexical LM which was based on maximum entropy (ME) modelling. This gave the freedom to include more information such as morphological and syntactic as well as number and gender into the modelling process. The information was provided by means of a lexicon that was prepared by a language expert. Their method achieved a 9.3% relative reduction in WER on an ICA recognition task. Within the same modelling concept, El-Desoky et al. (2013) used feed-forward deep neural network LMs to achieve a relative improvement of 6.5% in an ECA recognition task. Again, morphological and syntactic information was provided by means of expert-prepared lexicons.

In the absence of any linguistic-based resources, Creutz et al. (2007b) estimated a LM based on training samples decomposed using non-linguistic metrics. They employed Morfessor (Creutz and Lagus, 2005). Based on their method, recognition performance on an ECA recognition task was degraded by a 4.2% relative compared to word-based LMs. The authors ascribed this degradation to the complex morphology of ECA which was not captured by Morfessor. Their results cannot be compared with the work discussed above because they used only single-pass recognition while all the studies discussed above used their LMs in multi-pass decoding approach.

Table 4.2 summarises related work in Arabic ASR with morpheme-based LMs. As can be seen, a reduction in WER relates to the quality of the morphological analysis technique used. A larger improvement is observed when morphological analysis is provided by language experts (including BAMA and MADA which include an internal lexicon) compared to when rule-based approaches are used.

4.1.1 Motivations

As discussed in Section 3.1.4, decoders solve the search problem in Equation 3.1 by expanding words, phonemes and HMM states, and by accumulating scores from acoustic and language models to find the path with the highest likelihood. This implies that the process of expansion is made on a closed set of words, i.e. a limited vocabulary. Therefore, the selected vocabulary should provide the best coverage possible for the targeted domain. For morphologically rich languages such as CA, a large vocabulary list is required to get the desired coverage due to the inflection and agglutination processes. Using large vocabularies requires more training data, which is a key issue for CA at the time being.

As discussed previously, word decomposition limits the impact of vocabulary growth. For example, Figure 4.1 shows that by a simple decomposing of the definitive *Al* in MSA and LCA text, a significant relative reduction in vocabulary size (20% for MSA and 11% for LCA) is obtained for both Arabic variants. As shown in Table 4.2, successfully applied

Table 4.2: A comparison between different strategies in modelling morpheme-based LMs and the achieved relative reduction in WER (WER Red%) in Arabic ASR. All results combined morpheme-based LM with a full word LM.

Reference	Language	LM	Word decomposition	LM streams	Decoding	WER Red%
Kirchhoff et al. (2003)	ECA	FLM	morphological lexicon + morpheme generator	morph, root, stem, pattern, POS	N-best list	<2%
Vergyri & Kirchhoff (2004)						
Kirchhoff et al. (2006)						
Vergyri et al. (2005)	LCA		rule-based with a predefined affixes set + BAMA			
El-Desoky et al. (2010)	MSA		rule-based with a predefined affixes set			2.7-13.3%
El-Desoky et al. (2012)			MADA			3.7%
Xiang et al. (2006)	MSA	n-gram	rule-based with a predefined affixes set	morph		3.7%
Choueiter et al. (2006)			morpheme generator + WFSA accepter			0.7-7.5%
Lamel et al. (2008)			BAMA			1.9%
Diehl et al. (2009a)			MADA	morph	CN	5.4-8.1%
Nguyen et al. (2009)			rule-based with a predefined affixes set+ Sakhr morph analyzer	morph	N-best list	6.9%
El-Desoky et al. (2009)			rule-based with a predefined affixes set			3%
Afify et al. (2006)	ICA		rule-based with a predefined affixes set + morphological lexicon			8.7-10%
Creutz et al. (2007b)	ECA		morfessor		1-pass	-4.2%
Sarikaya et al. (2007)	ICA	JMLLM	morphological lexicon	morph, POS, gender, number	N-best list	9.3%
Kuo et al. (2009; 2010)	MSA	NNLM	morphological lexicon	POS, morph		5.5%
El-Desoky et al. (2013)	ECA	DNNLM	MADA	stem, morph, pattern		6.5%

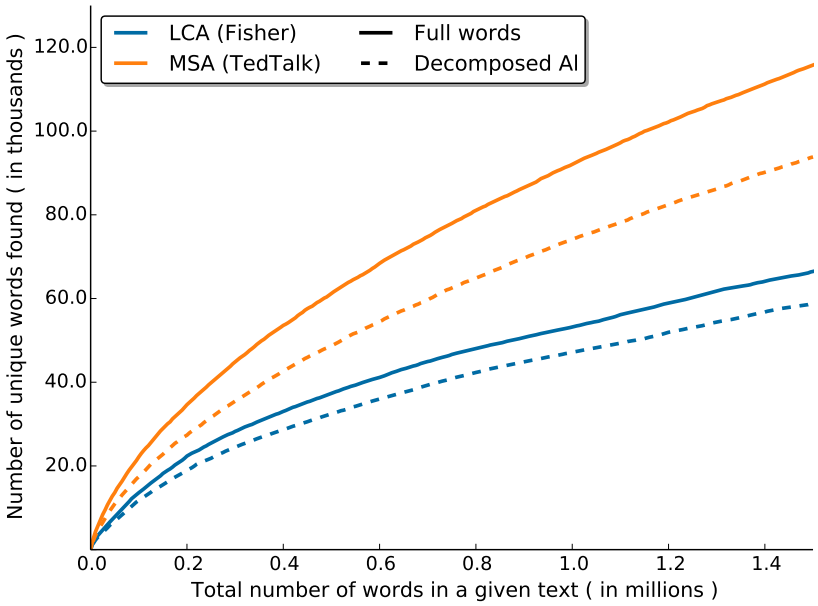


Figure 4.1: Vocabulary growth for two variants of Arabic when no decomposition is applied in comparison to when the definitive Al is decomposed.

word decomposition for CA was best observed when using morphological lexicons or a combination of rule-based segmentation and MSA-based morphological analysers (such as BAMA or Sakhr's). However, few unsuccessful attempts were reported in the literature that used word decomposition without any linguistic knowledge.

Therefore, a thorough investigation of word decomposition in CA in the absence of expert-based lexicons is presented in the following sections, where different levels of linguistic knowledge are incorporated. Linguistic knowledge ranges from extensive manual morphological analysis to no linguistic knowledge.

4.2 Word decomposition

Word decomposition methods can be grouped into two main approaches based on the amount of linguistic knowledge involved. In this context, linguistic knowledge is equivalent to supervision level. For the first approach, word decomposition is performed based on linguistic knowledge, either by using a full morphological analysis for the segmentation (Section 4.2.1), or by predefining a set of affixes along with a rule-based segmentation algorithm (Section 4.2.2). In second approach, no linguistic information is employed. Instead it relies completely on an information-theoretic approach (Section 4.2.3). Resulting word particles are called morphemes in the former and morphs in the latter.

To facilitate re-composing the word from its components, prefixes are appended by a '+' symbol and suffixes are preceded by a '+', whereas no symbols are attached to stems.

For example, the word "drshm" is decomposed to two morphemes "drs +hm" and the word "wsydrswnhA" is decomposed to six morphemes "w+ s+ y+ drs +wn +hA". It has been reported in many previous studies, such as of Geutner (1995), that having short morphemes reduces recognition accuracy due to the increase of acoustic similarity between them and their weak context modelling. Such artefacts can be limited by decomposing to longer morphemes. Therefore, a sequence of prefixes or suffixes can be merged into one prefix or suffix, for example, "w+ s+ y+ drs +wn +hA" has three prefixes and two suffixes which can each be merged into three morphemes (one prefix, one stem and one suffix) to be "wsy+ drs +wnhA".

4.2.1 Supervised decomposition

A full morphological profile (i.e. the output of a morphological analyser) should be provided for a given text in order to be used in the decomposition process. This profile is either derived from a morphological lexicon prepared by a language expert, or by an automatic analyser along with a seeding lexicon. Although using expert-made lexicons provided a morphological profile with the highest quality, it is expensive to develop and to expand for adoption of new words. Therefore, using an automatic analyser with a seeding lexicon is preferable, for cost and flexibility. However, because the morphological lexicons are mainly written by experts, they might be with a limited coverage where not all encountered words can be analysed. BAMA (Buckwalter, 2002; 2004b) is one of the most common morphological analysers for MSA. It is based on a rule-based algorithm with three morphological lexicons for stems, suffixes and prefixes. BAMA generates multiple analyses for a given word, especially if it is an undiacritised word.

The resulting decomposition is consistent and independent of the data volume, i.e. the same decomposition is generated for a given list of diacritised vocabulary, regardless of the amount of provided text. If no diacritisation is provided, multiple analyses are generated. An algorithm or a statistical model can be used to disambiguate which decomposition is chosen according to the provided context. MADA (Habash and Rambow, 2005) uses several SVM classifiers to rank a list of analyses list produced by BAMA according to the provided context. If a word cannot be analysed, e.g. loanwords or colloquial words, the word remains without any decomposition. Otherwise a word is replaced by a sequence of three components: prefixes, stem and suffixes.

4.2.2 Semi-supervised decomposition

As stated at the beginning of this chapter, CA words cannot be processed using MSA-based tools because of the introduction of new stems and affixes that are not part of MSA.

For example, the word "yqwlwA" (they are saying) exists in MSA and can be successfully analysed and decomposed using BAMA and MADA to "y+ qwl +wA". However, when this word is preceded by a colloquial prefix, such as "b+" as in "byqwlwA" (they are saying) or "H+" as in "HyqwlwA" (they will say), it becomes illegible as an MSA word and cannot be processed. Another case that cannot be processed by these tools is a colloquial stem with MSA affixes such as the word "$AfwA" (they saw). This is the word "$Af" (he saw) inflected with an MSA suffix "+wA" (they), but since the stem does not belong to MSA, it cannot be analysed.

For that, another approach is adopted for CA, especially in the absence of a morphological lexicon. Instead one predefines a set of affixes and uses a rule-based segmentation algorithm. Since the segmentation is done automatically and the only linguistic knowledge is the affix set, it is known as semi-supervised decomposition or blind segmentation (Afify et al., 2006).

Given that a word can contain more than one prefix and one suffix, the set of affixes to be defined must contain all possible combinations of prefixes and suffixes. For example, the following set is defined for decomposing LCA text:

Prefixes: Al+, hAl+, w+, wAl+, whAl+, b+, bAl+, bhAl+, wb+, wbAl+, wbhAl+, f+, fAl+, fhAl+, wf+, wfAl+, wfhAl+, $+, $Al+, $hAl+, w$+, w$Al+, w$hAl+, l+, ll+, lhAl+, wl+, wll+, EAl+, EhAl+, wEAl+, wEhAl+, kAl+, wkAl+.

Suffixes: +wA, +p, +whA, +whn, +whm, +wkm, +h, +wh, +hA, +hn, +hm, +y, +yp, +yn, +nA, +ny, +t, +At, +yAt, +yAthA, +tm, +yAthn, +yAthm, +k, +kn, +km.

Based on the reported literature (discussed in Section 4.1), very short morphemes (stems and affixes) should be avoided as much as possible. Moreover, the most frequent words should be kept intact for their positive impact on recognition performance.

4.2.3 Unsupervised decomposition

The problem of learning the orthographic structure from raw textual data with minimal supervision and intervention is known as unsupervised learning of morphology. This problem has been addressed in the last two decades with several strategies which have been described thoroughly in a recent survey by Hammarström and Borin (2011). One group of methods use criteria on the frequency of occurrences, for sub-lexical strings in order to find the boundaries between morphemes, similar to criteria used in data compression. Many authors of these methods employed the minimum description length (MDL) algorithm, e.g. Goldsmith (2001), Creutz and Lagus (2005) and Xanthos et al. (2006). MDL is an algorithm that seeks the smallest possible set of morphs to represent a given corpus with

the minimum coding length. Each morph in the given corpus is replaced with a positive numerical binary pointer. Frequent morphs are assigned shorter pointers, i.e. pointers have fewer bits, in order to minimise the overall length. Pointer length is computed as the negative binary logarithm of a morph likelihood within the corpus. These methods are mostly language-independent, but corpus dependent; i.e. the resulting set of morphs for the same language might change when they are extracted from two different corpora.

Morfessor is an open-source toolkit (Creutz and Lagus, 2005; Virpioja et al., 2013), which has been developed as three variants: Baseline, Categories-ML and Categories-MAP. Whilst Morfessor Baseline is a context-independent implementation of an MDL-based method, Categories-ML and Categories-MAP extend that to incorporate the context in redefining the morph list. Morfessor Categories-ML finds the optimal segmentation via maximum likelihood (ML) re-estimation. Morfessor Categories-MAP uses a maximum-a-posteriori (MAP) model with hierarchical structure for the vocabulary list to label each morph to be either a prefix, stem or suffix tag.

4.3 Sub-lexical unit LMs for CA

A model similar to word-based n-gram LM (see Section 3.1.3) is estimated from a decomposed corpus using one of the previous methods discussed in Section 4.2 to obtain either morpheme- or morph-based LMs.

Perplexity is used as a metric to measure the quality of an LM on a given test set. A comparison between LMs in terms of their computed perplexities on a given test set is meaningful only when these LMs share the same vocabulary. Morph-based LMs and word-based LMs differ in their linguistic units. The former uses sub-lexical units and the latter uses non-decomposed words as units. Their performance can be compared using an approximation for character-level based on the unit-level[4] perplexity. Given the unit-level perplexity ppl (stated in Equation 3.9), this approximation can be computed as follows:

$$\begin{aligned} ppl_c &= (ppl)^{\frac{N}{M}} \\ &= \left(\left(\prod_{i=1}^{N} P(w_i|w_{i-n}...w_{i-1})\right)^{-\frac{1}{N}}\right)^{-\frac{N}{M}} \\ &= \exp\left(\frac{-N}{M}\frac{-1}{N}\sum_{i=1}^{N}\log P(w_i|w_{i-n}...w_{i-1})\right) \end{aligned} \tag{4.1}$$

where N is the size of the corpus in words or morphs including sentence boundaries, and

[4]Word, morph or morpheme

M is the size of the corpus in characters, including word and sentence boundaries.

Beside perplexity, OOV rate is used to measure the coverage of the lexicon on a given test set. Again, OOV rate can be normalised by the average number of morphs per word in the corpus. Formally:

$$\%\text{OOV}_{\text{norm}} = \%\text{OOV} \times \frac{\text{testsize}_{\text{morphs}}}{\text{testsize}_{\text{words}}} \tag{4.2}$$

where %OOV is the OOV rate in terms of morphs while $\text{testsize}_{\text{morphs}}$ and $\text{testsize}_{\text{words}}$ are the size of the test set in terms of morphs and words respectively.

4.4 Incorporating sub-lexical unit classes

For under-resourced CA data sparsity, i.e. insufficient number of training samples, is an evident problem. Words that share a POS or a morphological feature, such as names and direct masculine pronouns, are expected to appear in a similar context. As a result of grouping similar words into clusters, a smaller number of classes is produced which reduces the effect of data sparsity. Consequently, the generalisation of language models is improved, allowing to accommodate unseen context and histories.

Several methods have been proposed to map words onto classes which can be either linguistic-based or data-driven. Linguistic-based methods use POS and morphological tags resulting from a syntactic parser or morphological analyser. Data-driven methods use greedy algorithms to group words into classes in order to minimise a given objective function computed from the training set. An example is probabilistic latent semantic analysis (Deng and Khudanpur, 2003). Some data-driven word-to-class mappings have been devised,for instance by using mutual information between successive words (Brown et al., 1992) or by using separate clustering for each word position (Emami and Jelinek, 2005).

In this section, two methods are described in order to incorporate morph or morpheme classes into language modelling without any linguistic-based class assignment. The first method is the class LM (Brown et al., 1992) which clusters words (Section 4.4.1). The second method uses Random Forest LMs (Xu and Jelinek, 2004) which groups histories instead of individual words (Section 4.4.2).

4.4.1 Class LM

The main idea in Class language model (CLM) is to cluster similar words into classes where all members of the same class are treated equally. Brown et al. (1992) proposed CLM which is an n-gram LM estimated over a sequence of classes where the word transition probability

is composed of a class transition probability and a word emission probability, formally:

$$P_{\text{CLM}}(w_i|w_{i-n}...w_{i-1}) = P(c_{w_i}|c_{w_{i-n+1}}...c_{w_{i-1}})P(w_i|c_{w_i}) \tag{4.3}$$

where c_w is the class of word w and $P(c_{w_i}|c_{w_{i-n}}...c_{w_{i-1}})$ is the class transition probability. $P(w_i|c_{w_i})$ is the word emission probability, which can be computed using:

$$P(w_i|c_{w_i}) = \frac{\text{Count}(w_i)}{\sum_{w \in c_{w_i}} \text{Count}(w)} \tag{4.4}$$

$\text{Count}(w)$ is the number of occurrences of the word w in the training corpus.

Word classes can be extracted efficiently using an agglomerative hierarchical clustering algorithm (Brown et al., 1992). This yields a binary tree with its internal nodes representing classes and its leaves representing words. Brown et al.'s word clustering algorithm uses a series of merges to optimise the perplexity over a given development set. It initialises with each word representing a class, then greedily finds the two classes where merging them achieves the best perplexity improvement. The merged classes are then represented by one class, and the search and merging process is recursively repeated until the desired number of classes is reached. Another variant, also introduced by Brown et al. (1992), assigns the most frequent words to their own classes to obtain the desired number of classes, then greedily adds words to each class to optimise perplexity on a development set. Both variants result in a hard word-to-class mapping where each word can belong to only one class.

Using CLMs generalises language model to unseen context in the training at the expense of weaker word predictive ability. Therefore, these models are linearly interpolated with word LMs in order to improve prediction performance.

4.4.2 Random Forest LM

For n-gram LM, a word w_i is predicted by its history, that is the sequence $w_{i-n+1}...w_{i-1}$. The building block for Random Forest LMs is a *decision tree* LM (DTLM). A DTLM (Bahl et al., 1989) uses a decision tree to group all observed histories into classes as the tree leaves. All histories within a class share the same distribution over predicted words. Starting from the root, each leaf (the equivalence class of histories) is reached by answering a series of questions about words in a specific position in the history, such as "is the word at position $i-2$ = \$w?". Figure 4.2 illustrates how a set of histories are clustered according to a given question, where the sets at the leaves will have smaller set of training data than at their parent nodes, as it shows in the counts. Word transition probabilities are computed using interpolated Kneser-Ney smoothing (discussed in Section 3.1.3) as follows:

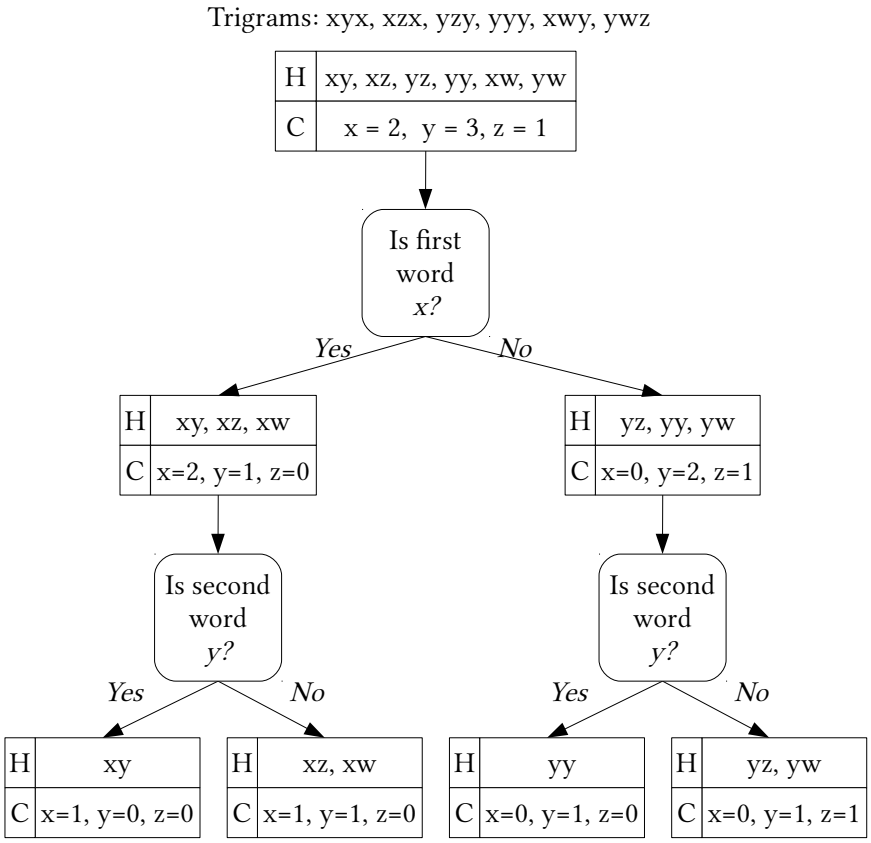


Figure 4.2: Clustering of histories in DTLMs. H denotes the observed histories in the training set, and C is the counts of words that follow the histories within a node.

$$
\begin{aligned}
P_{\text{DTLM}}(w_i|w_{i-n+1}...w_{i-1}) &= P_{\text{DTLM}}\left(w_i|\Phi_{\text{DT}}(w_{i-n+1}...w_{i-1})\right) \\
&\cong \frac{\max(\text{Count}(w_i, \Phi_{\text{DT}}(w_{i-n+1}...w_{i-1})) - D, 0)}{\text{Count}(\Phi_{\text{DT}}(w_{i-n+1}...w_{i-1}))} \\
&+ \lambda(\Phi_{\text{DT}}(w_{i-n+1}...w_{i-1}))P_{\text{KN}}(w_i|w_{i-n+2}...w_{i-1}),
\end{aligned}
\tag{4.5}
$$

where $\Phi_{\text{DT}}(w_{i-n+1}...w_{i-1})$ maps the input history to a class using the tree DT and $\text{Count}(w_i, \Phi_{\text{DT}}(w_{i-n+1}...w_{i-1}))$ is the number of occurrences of word w_i following the given history in the class of histories. D is a constant discount factor and P_{KN} is the Kneser-Nay back-off probability distribution (as in Equation 3.6). During the construction of a decision tree, the set of questions is automatically defined based on the training samples contained in a node; however, a node is split based on the improvement in the training data likelihood and the amount of training data within each new node. Then, these history classes are recursively refined while the tree grows. Each leaf can be split by asking questions about word identities at a specific position in observed histories until a stopping criterion is satisfied.

Random Forest LMs (RFLM) (Xu and Jelinek, 2004) are constructed by the use of a collection of M randomised DTLMs. The word probabilities are estimated by averaging the individual randomised DTLMs:

$$
P_{\text{RFLM}}(w_i|w_{i-n+1}...w_{i-1}) = \frac{1}{M}\sum_{j=1}^{M} P_{\text{DTLM}}(w_i|\Phi_{\text{DT}_j}(w_{i-n+1}...w_{i-1})) \tag{4.6}
$$

$P_{\text{DTLM}}(w_i|w_{i-n+1}...w_{i-1})$ describes the probability computed from Equation 4.5 using M decision trees. Each tree is constructed by randomising its initialisation and its growing question set (e.g the order of positions to ask about in the history).

4.5 Experiments

Several experiments were conducted to compare the impact of different levels of supervision within word decomposition in CA data (Section 4.5.1) and to empirically evaluate the performance of the estimated morpheme- and morph-based LMs from the decomposed data (Section 4.5.2). Finally, the impact of incorporating sub-lexical unit classes on sub-lexical unit language modelling using class-based (Section 4.5.3) and random forest (Section 4.5.4) was evaluated.

Table 4.3: Training and testing set characteristics.

	Sentences	Words	Vocabulary size
AppenLCA	57488	377944	31695
FisherLCA	375588	1528342	67195
total	433076	1906286	81636
testLCA	12051	53644	8762

In these experiments, two training sets from LCA were employed, FisherLCA and AppenLCA (more details in Appendix B), which are collections of telephone conversations in Levantine Arabic (LCA) transcribed to word-level. The two test sets from the same corpora were merged into one test set (*testLCA*). All sets were preprocessed, and all diacritics were removed, and the different graphemes of alif were mapped onto one grapheme "A". Moreover, all disfluency markers were removed, and all backchannel tags were mapped onto one tag instead. Table 4.3 summarises the characteristics of these three sets.

A vocabulary of 41688 words (referred to as *vocablist*) was chosen by keeping all non-singletons from AppenLCA and FisherLCA training sets. This has an OOV rate of 2.5% on the *testLCA*. As a baseline, word-based n-gram LMs of order 3 were estimated from each training set independently, and then linearly interpolated. The perplexities of *word-Appen* and *word-Fisher* LMs were 756 and 250 respectively. The linearly interpolated LM, *word-int*, gave a lower perplexity of 214.

4.5.1 Resulting morphemes and decomposed data

Four segmentation methods were employed for decomposing training and test sets in addition to the *vocablist*: supervised segmentation using MADA (*mada*); semi-supervised segmentation using a rule-based algorithm along with a predefined affix set (*blind*); and two unsupervised segmentations using Morfessor-Baseline (*morfbase*) and Morfessor-Categories-MAP (*morfcamap*). Two segmentations were performed based on each method. For the first, all observed words in both training sets, namely 81636 words, were included in the decomposition (*fv*). The most frequent 5000 words were excluded from the decomposition process in the second segmentation (*sv*). However, in both segmentations, words with fewer than five occurrences were excluded to avoid noisy and inconsistent results.

An increase in the training size was expected when using word decomposition because each word was decomposed into multiple units. In addition, a decrease in the number of unique units (i.e. vocabulary size) was expected because the morphological inflection and agglutination effect was reduced. The degree of change in the vocabulary size and training text depended on the chosen decomposition method. Figure 4.3 summarises the changes in the vocabulary size in *AppenLCA*, *FisherLCA* and *testLCA* and in the size of *vocablist*

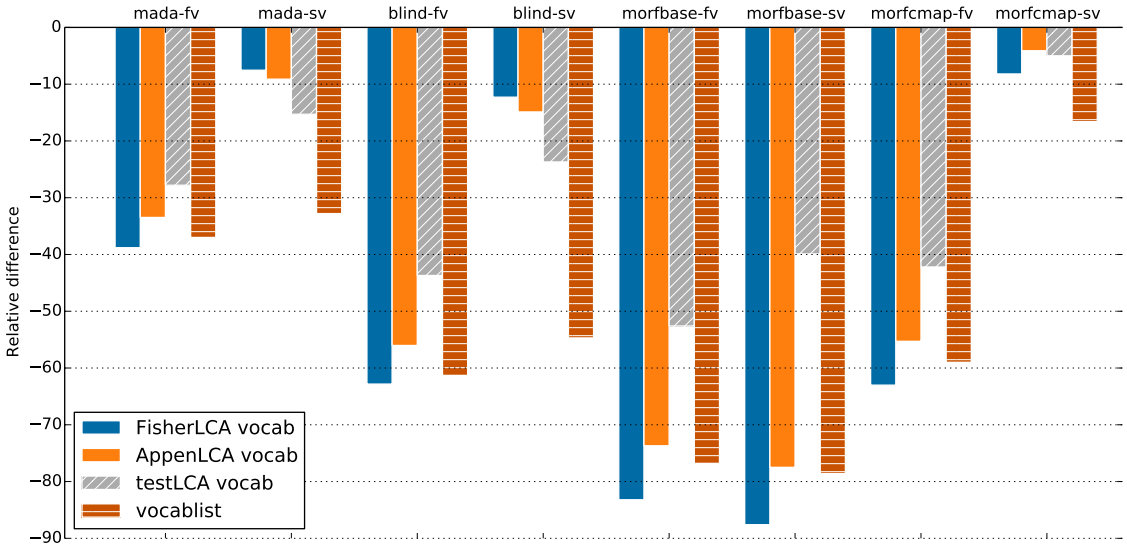


Figure 4.3: Relative difference in vocabulary size after applying different word decomposition approaches on training and testing sets and the vocabulary list. Each segmentation is shown in the format of (A-B), where A is the decomposition method and B is the included vocabulary.

when applying the aforementioned decompositions. Each segmentation is denoted by the format of A-B where A is the decomposition method (mada, blind, morfbase or morfcmap) and B is the included vocabulary in the decomposition (fv and sv). For example, mada-fv represents using MADA and the complete vocabulary was included in the decomposition process. Figure 4.4 shows the changes in the text size for the training and testing sets.

Almost a third of the overall vocabulary, representing 11.1% of the training data text, was left unprocessed using MADA because no MSA matches were found for the unprocessed words. When the full vocabulary was decomposed (*mada-fv*), an average decrease of 36.1% in the overall vocabulary size was obtained, whereas 34.9% increase in the training data volume was observed. Such a difference was not obtained when the 5000 most frequent words were excluded from the decomposition (*mada-sv*), giving a modest average vocabulary reduction of 8.3% and an increase in the overall training volume of 3.9%.

A rule-based algorithm was applied on the full vocabulary list after defining a set of affixes, as listed in Section 4.2.2. Because this affix set was defined based on knowledge of LCA, more words were decomposed, especially when full vocabulary was used (*blind-fv*). This gave an average reduction of 59.4% and an average increase in the text volume of 50.8%. Excluding the most frequent words from the decomposition (*blind-sv*) decreased this difference to get a 13.6% decrease in the vocabulary size and 7.9% average increase in training volume.

Two segmentation models were trained using Morfessor-Baseline on the pool of training

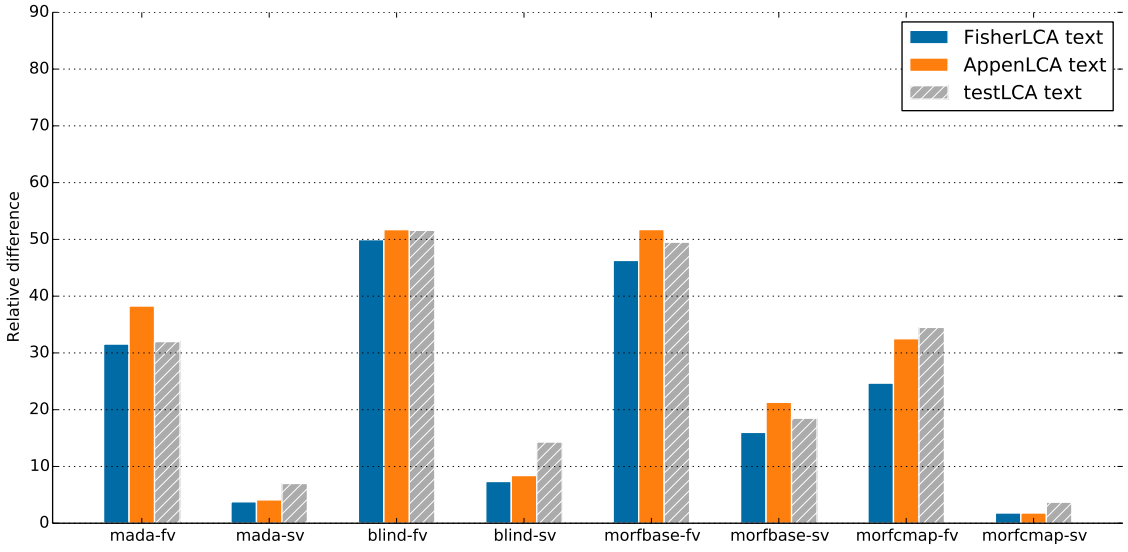


Figure 4.4: Relative difference in the total number of words after applying different word decomposition approaches on training and testing sets. Each segmentation is shown in the format of (A-B), where A is the decomposition method and B is the included vocabulary.

data. The first model was trained using the full vocabulary (*morfbase-fv*), and subsequently used in decomposing both training sets. The vocabulary size reduced significantly by 78.5%, and the overall training volume increased by 49.0%. Again, the 5000 most frequent words were excluded from the vocabulary (*morfbase-sv*) to reduce the vocabulary size by 82.5% and increased the training volume by 18.6%.

Similarly, two further segmentation models were trained using Morfessor-Categories-MAP with full vocabulary (*morfcmap-fv*) and shortened vocabulary (*morfcmap-sv*) to get an average vocabulary reduction of 56.2% and 6.2% respectively and an average increase in the training volume of 28.6% and 1.8% respectively.

In general when decomposing all observed vocabulary, the unsupervised methods *morfbase* and *morfcmap* performed more decomposition than any supervised methods; consequently, they generated the smallest vocabulary lists. Because *morfcmap* relies primarily on frequency of words for decomposition, excluding the most frequent words has a significant impact and reduces the number of morphs generated compared to considering all vocabulary. For the knowledge-based methods, *mada* and *blind*, there was a similar difference between considering all vocabulary and excluding the most frequent words, indicating a stability in the decomposition as discussed before. However, using a rule-based algorithm with a tailored affix set for CA generates more morphemes than the MSA-based tool (MADA).

Each decomposition method generated a different number of morphs of various sizes, Table 4.4 compares the characteristics of the morphs resulting from each method described

Table 4.4: Characteristics of the resulting morphs when applying various word decomposition approaches. *morph/w* denotes to the average number of morphs per word. *Weighted morph/w* is the average number of morphs per word computed from the overall decomposed text, i.e. weighted by frequency. *char/morph* is the average morph length in characters.

	fv			*sv*		
	weighted morph/w	morph/w	char/morph	weighted morph/w	morph/w	char/morph
mada	1.4	1.6	6.2	1.0	1.6	6.2
blind	1.5	2.1	5.2	1.1	2.0	5.2
morfbase	1.5	2.0	5.4	1.2	2.5	5.6
morfcmap	1.3	1.8	5.6	1.0	1.3	6.3

above. Overall, the average number of morphs per word is lower when it is weighted by word frequency in the training data. This is more evident when the most frequent words are excluded from decomposition, as shown in *sv* side in Table 4.4. The rule-based algorithm (*blind*) generated the shortest morphs among all decomposition methods, which explained its high average of morphs per word.

4.5.2 Perplexity and OOV rate of morph-based LMs

As a baseline, two word-based 3-gram LMs with modified Kneser-Ney discounting were trained, using the SRILM toolkit (Stolcke, 2002), based on the described *vocablist* from the *FisherLCA* and the *AppenLCA* training sets. Then the two word-based LMs were linearly interpolated to obtain a word-level perplexity of 240 computed on the *testLCA* with bigrams and 214 with trigram LM.

Similarly, an interpolated morph-based LM was estimated when the decomposed *vocablist* based on one of the methods described above in Section 4.5.1 was used on its corresponding decomposed training sets. Then the perplexity of the resulting LM is computed from a variant of *testLCA* that was decomposed using the same decomposition method used for the *vocablist* and the training sets.

In order to compare the OOV rate of morph-based LMs and the word-based LM baseline (2.5%), all OOV rates were normalised using Equation 4.2. Table 4.5 lists all morph-level OOV rates and their corresponding normalised OOV rates along with the relative reduction obtained compared to the word-based LM OOV rate. Apart from the *morfcmap* method, similar normalised OOV rates were achieved regardless of excluding the most frequent word during the decomposition (*sv*) or not (*fv*). A large difference in normalised OOV rates between *sv* and *fv* settings for the *morfcmap* method was observed. The highest normalised OOV rate was obtained when using text decomposed by MADA. The normalised

Table 4.5: Relative reduction in normalised OOV rate for morph-based LMs compared to the baseline word-based LM (=2.5%). Models were estimated from decomposed text using different decomposition setting for each model.

LM		OOV%	norm OOV%	Red%
vocab	method			
fv	*mada*	1.2	1.6	-37.2
	blind	0.6	1.0	-61.2
	morfbase	0.1	0.1	-97.0
	morfcmap	0.6	0.7	-70.6
sv	*mada*	1.5	1.6	-37.1
	blind	0.8	1.0	-61.6
	morfbase	0.0	0.0	-100.0
	morfcmap	2.1	2.1	-14.6

OOV rate reduced as the amount of linguistic knowledge involved in the decomposition process reduced until it reached 0.0 OOV rate for morph-LMs estimated from completely unsupervised decomposed text.

Table 4.6 lists the morph-level perplexities based on using morph-based LMs of order 2 to 7. Comparing each column within the same row shows that the perplexity decreases with adding more context depth; however, this improvement becomes less obvious beyond order 4, especially when the applied decomposition excluded the most frequent words (*sv*). For instance, the perplexity gain for the *mada-fv* morph-based LM by increasing the order from 2 to 3 was 30% relative, and from 3 to 4 was 5% relative. However, perplexity improved by only 10% and 1.5% relative respectively for the *mada-sv* morph-based LM. Less than a 1% relative reduction was observed beyond using quadgrams for the *mada-fv* morph-based LM and beyond trigrams for the *mada-sv* morph-based LM. This relates to the average weighted number of morphs per word in Table 4.4 where a higher perplexities improvement is expected if the average weighted number of morphs per word is more than 1, as is the case for *mada-sv*, in contrast to *mada-fv*.

As discussed in Section 4.3, due to the difference in the linguistic units with different decomposition methods used for each LM, the perplexities shown in Table 4.6 are not comparable across decomposition methods. Hence, a normalised character-level perplexity is computed instead, as described in Equation 4.1. Figure 4.5 shows the corresponding character-level perplexities for all morph-based LMs in Table 4.6. As shown in the figure, the previous observation, that perplexity improves with the increase of context depth, holds true. Moreover, all morph-based LMs with full vocabulary decomposition (*fv*), apart from *morfbase*, have lower perplexity than word-based LMs by an average of 3% relative, where the latter has a character-level perplexity of 3.4 for trigram word-based LM and 3.5 for bigram word-based LM. In this case, the normalised character-level perplexity was

Table 4.6: Morph-based perplexity of morph-based LMs of order 2 to 7. Models were estimated from decomposed text using different decomposition setting for each model.

LM		order of morph-based LM					
vocab	method	2	3	4	5	6	7
fv	*mada*	113.1	85.7	80.3	79.1	78.8	78.7
	blind	78.9	57.9	53.0	51.9	51.6	51.5
	morfbase	94.9	63.3	57.2	55.9	55.5	55.3
	morfcmap	111.7	84.7	79.8	78.7	78.4	78.4
sv	*mada*	217.6	194.3	191.4	191.0	191.0	191.0
	blind	200.7	176.8	173.8	173.5	173.5	173.5
	morfbase	180.7	150.6	146.8	146.1	146.0	146.0
	morfcmap	228.9	206.4	203.6	203.2	203.2	203.2

related to the amount of constraints restricting the decomposition process. For instance, the highest perplexity was obtained from employing a completely unsupervised approach (*morfbase*). However, the perplexity was improved by adding more constraints, such as statistical constraints as in *morfcmap* or linguistic constraints as *mada*. Morph-based LMs trained on text without the decomposition of the most frequent words performed slightly worse than word-based LM.

4.5.3 Class-based morph-based LM

For each of the decomposed training sets, word classes were extracted using Brown et al.'s clustering algorithm. To choose the best performing number of classes in terms of perplexity, several bigram CLMs were trained with different numbers of classes, ranging from 50 to 1500. Figure 4.6 visualises the perplexity changing on the character-level to allow comparison across different decomposition methods. Perplexity improved as the number of classes increased for CLMs based on different decomposition methods. This improvement is considerable when increasing the number of classes from 50 to 500 classes. Beyond that point the improvement is less. CLMs based on *blind-fv*, *mada-fv* and *mada-sv* have an equivalent performance while CLMs based on *morfcmap-fv* perform slightly worse that them. None of these CLMs outperformed morph-based LMs (shown in Figure 4.5), even if they were interpolated with a corresponding morph-based trigram or quadram LM with the exception of *morpfbase-fv* and *blind-sv*. Figure 4.7 shows the character-level perplexity when interpolating CLMs with a corresponding morph-based quadram LM across different decomposition methods. Each decomposition methods is represented by two lines: a line with a yellow background that shows the morph-based LM performance without any interpolation and a line without any background where it shows the morph-based LM interpolated with its corresponding bigram CLM when using optimised interpolation

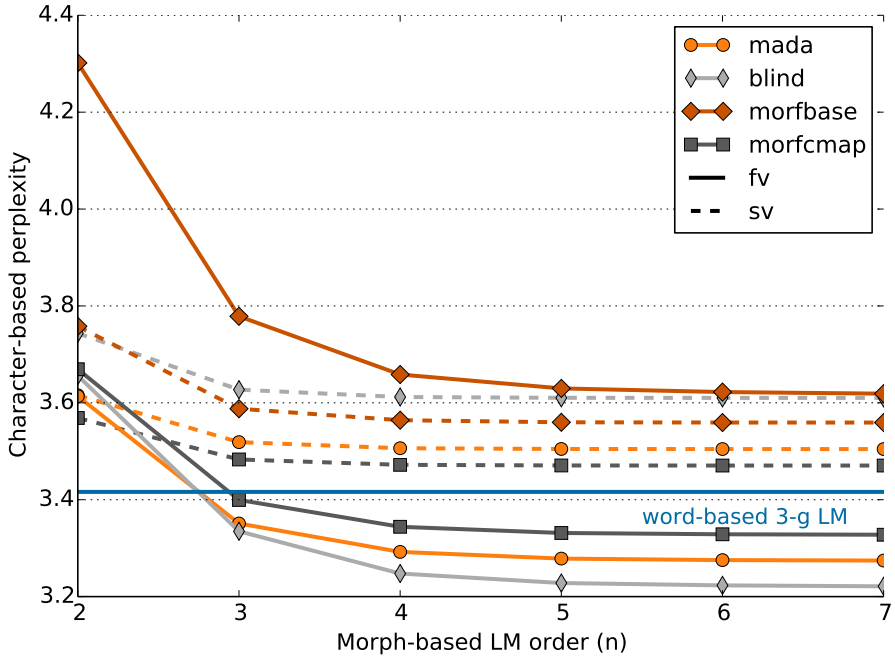


Figure 4.5: Character-level perplexity (y-axis) of morph-based LMs of order 2 to 7 (x-axis). Models were estimated from decomposed text using several decomposition methods (marker style) with either full vocabulary (solid line), or exclusion of the most frequent words from decomposition (dashed line). The perplexity from the word-based LM is represented by a solid line without markers.

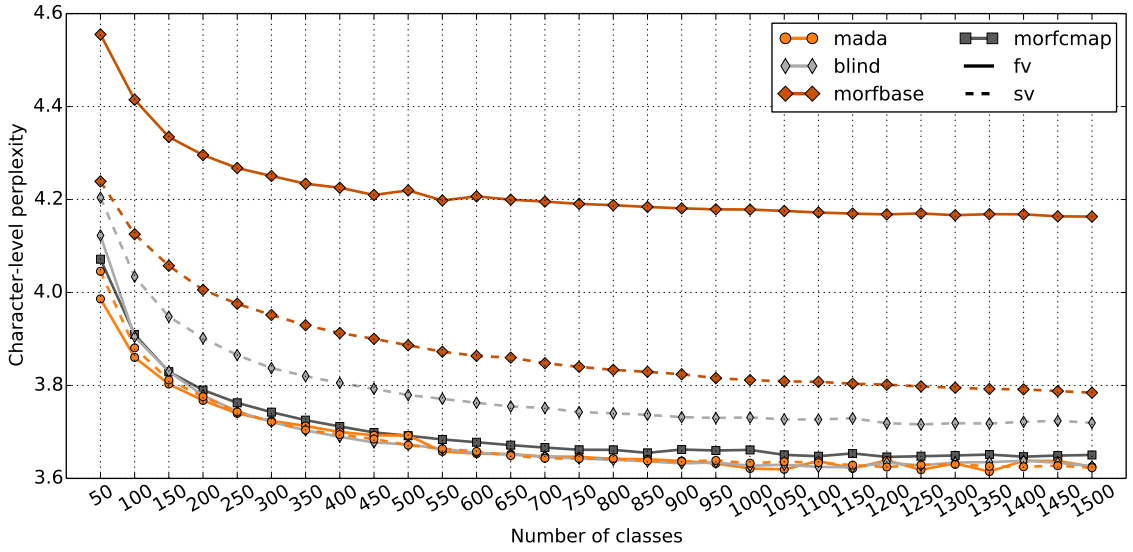


Figure 4.6: Character-level perplexity (y-axis) of bigram CLMs using different number of classes (x-axis). Models were estimated on decomposed training data using several decomposition methods (marker shape) with either full vocabulary (solid line) or excluding the most frequent words from decomposition (dashed line).

weights. Most of the interpolated CLM show the best performance when using 600-750 classes. Interpolating *sv* morph-based LMs with the corresponding CLMs marginally outperformed morph-based LMs by at least 0.8% relative; however, this was not the case for *fv* morph-based LMs although the latter still outperforming word-based LM.

4.5.4 Random forest morph-based LM

Using the same data sets, morph-based RFLMs were trained on decomposed training data. 100 DTs were generated from *AppenLCA* and 200 DTs from *FisherLCA*. Then the resulting RFLMs were linearly interpolated by optimising the morph-level perplexity on *testLCA*. Unlike CLMs, all estimated RFLMs outperformed their morph-based standard LM counterparts by at least 3.5% relative reduction in perplexity, as it is shown in Figure 4.8. In comparison to the word-based LM, the improvement is higher for RFLMs where the most frequent words were not decomposed (*sv*), where it ranges between 6.7% to 12.4% relative. The existence of these frequent words without decomposition helps in generating more diverse sequences of histories, especially in local context as in trigrams and quadgrams, which in turn improves the resulting DTs. The best performance was achieved by *morfcmap-sv* where perplexity was lower than that of the the word-based LM by 12.4% relative.

In order to investigate whether the number of the DTs is related to the quality of an RFLM, several morph-based RFLMs based on *morfcmap-sv* were estimated (since it

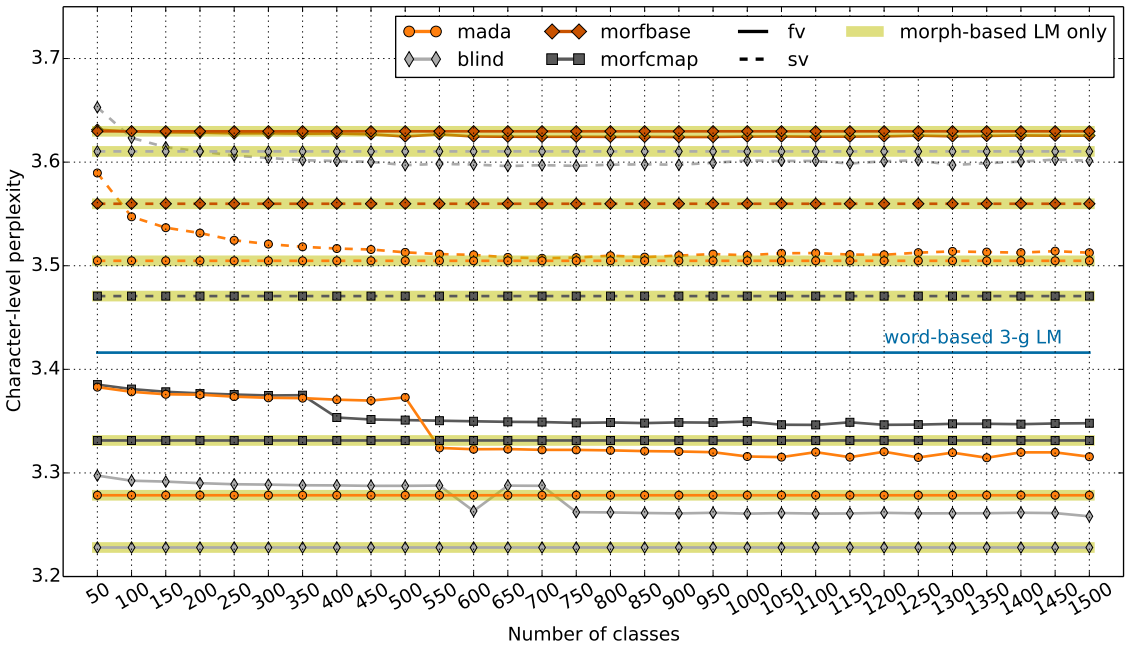


Figure 4.7: Character-level perplexity (y-axis) when linearly interpolating a 4-gram morph-based LM with a bigram CLM using different number of classes (x-axis). All models were estimated from decomposed training data using several decomposition methods (marker shape) with either full vocabulary (solid line) or excluding the most frequent words from decomposition (dashed line). Highlighted lines represent morph-based LMs without interpolation with CLM. Perplexity from the word-based LM is represented by a solid line without any markers.

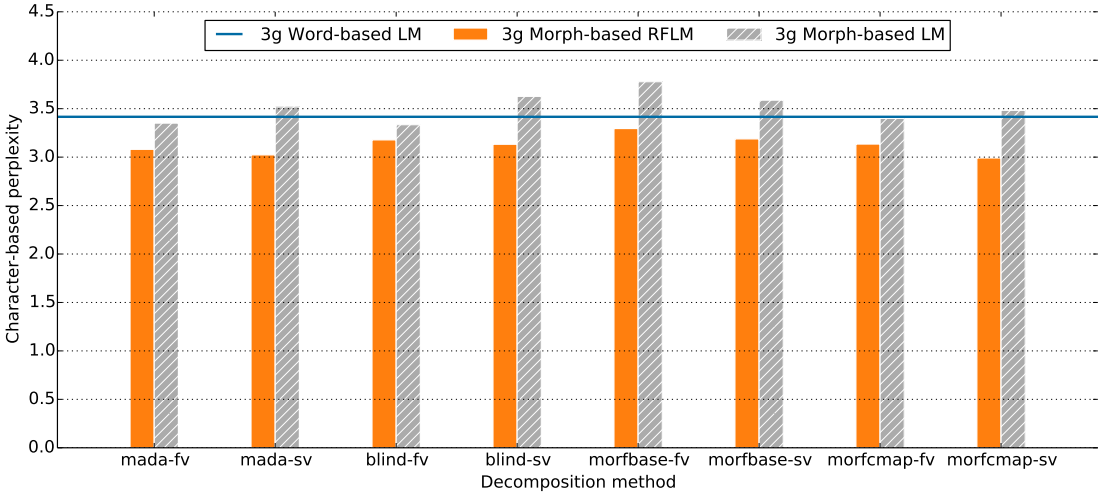
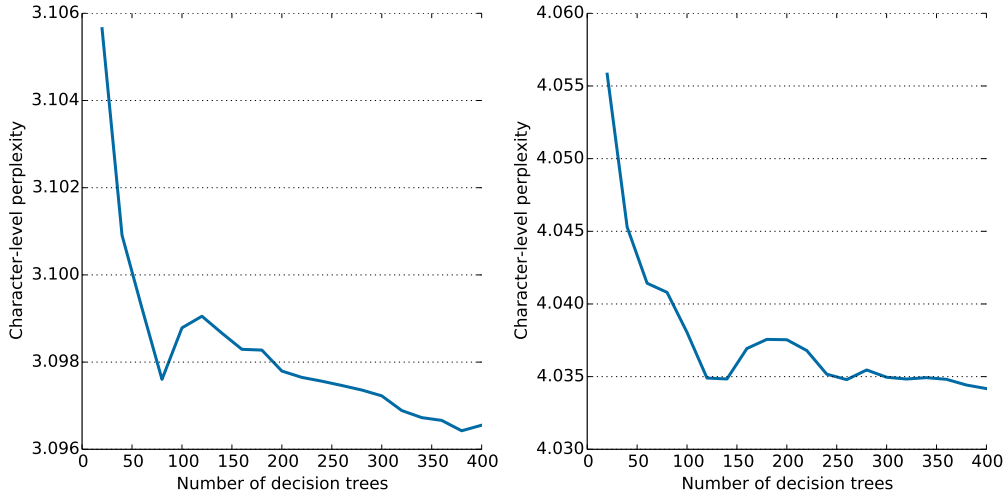


Figure 4.8: Character-level perplexity (y-axis) of trigram morph-based RFLMs and a corresponding morph-based LMs. Models were estimated on decomposed text using several decomposition methods (x-axis). Perplexity from the word-based LM is represented by a solid line without any markers.



(a) FisherLCA (b) AppenLCA

Figure 4.9: Character-level perplexity (y-axis) of trigram morph-based RFLMs estimated from (a) *FisherLCA* and (b) *AppendLCA* individually using several numbers of decision trees (x-axis).

gave the best performance so far). Figure 4.9 shows the effect on the character-level perplexity with a change in the number of DTs from 20 to 400, for two sets of RFLMs: one was estimated on *AppenLCA* and the other on *FisherLCA*. The change in character-level perplexity beyond using 100 DTs is insignificant, especially when the training set is large enough. In other words, the number of DTs makes little difference as long as it is equal or more than 100 trees.

4.6 Summary and conclusion

Arabic is a morphologically rich language with a high vocabulary growth rate. It has been shown that the vocabulary growth rate can be reduced by 30% with a simple morphological constraint such as decomposing the definitive *"Al"* using a rule-based algorithm. In line with Objective 1 of this thesis, more sophisticated word decomposition methods were investigated in this chapter with various degrees of linguistic knowledge involved in the segmentation process.

Without the existence of fully linguistic-based CA resources, such as manually generated morphological lexicons and analysers, the MSA tool, MADA, is considered the next best option. On the other extreme, two purely statistical and language-independent variants of Morfessor were included in this investigation where no linguistic knowledge was provided. Between these two extremes, a rule-based segmentation algorithm with a set of affixes to be decomposed was defined in order to address a specific dialect of CA. Gener-

ally, word decomposition limited the vocabulary growth and reduced the vocabulary size by different degrees depending on the decomposition method used, and whether parts of the vocabulary were excluded from the decomposition process. Regardless of the chosen method, word decomposition reached its best potential in reducing the vocabulary size when all observed vocabulary was included in the decomposition process.

Morph-based LMs, which are standard n-gram LMs estimated from decomposed text, with all vocabulary included in decomposition, outperformed word-based LMs that were estimated on unsegmented training data by an average of 3% relative character-level perplexity. As the level of CA-related linguistic constraints increased in the decomposition method, the performance improved in terms of character-level perplexities. For instance, a morph-based LM estimated from decomposed text using rule-based decomposition with manually defined CA affixes outperformed all its counterparts, followed by using MADA and then using Morfessor with MAP constraints, whereas completely unsupervised Morfessor without any constraints did not yield any improvements over word-based LM.

For all morph-based LMs, considerable reduction in OOV rates was observed. OOV rates decreased with the increase of linguistic knowledge or statistical constraints involved in the decomposition process. For example, normalised OOV rates were 37% for morph-based LMs estimated on decomposed text using MADA. This reduced to 0.0% for morph-based LMs estimated from completely unsupervised decomposed text.

As an attempt at addressing data sparseness in CA language modelling, classes were incorporated in the development of language models but at the morph-level instead of word-level. Two strategies were followed. In the first strategy, morphs were clustered into classes using Brown et al.'s word clustering algorithm along with its proposed class LM. In the second strategy, instead of clustering the morphs themselves, their histories were clustered using random fields LMs. CLMs resulting from the former strategy did not yield any improvement over their counterparts (standard morph-based LMs), unlike the RFLMs resulting from the second strategy where all of them achieved better performance than standard morph-based LMs and word-based LMs. The average improvement in character-level perplexity from RFLMs was 8% over word-based LM.

Unlike any of the previous studies, this work provided a detailed comparison of the use of sub-lexical unit LMs using several word decomposition methods. These methods ranged from linguistic-based, as in MADA, to completely unsupervised, as in Morfessor Baseline.

Based on these promising results, morph-based standard LMs and RFLMs, can be employed in a CA ASR within a multi-pass decoding where lattice or N-best list are rescored using these sub-lexical unit LMs.

Chapter 5

Exploiting standard Arabic data for colloquial Arabic language modelling

Contents

It was established in Section 2.7 that most of the challenges posed to the development of CA NLP tools in general and language modelling in particular can be summarised in two main issues: First, the high out-of-vocabulary (OOV) rate which results from the rich morphological nature of Arabic in general; second, insufficient amount of textual training data, also known as data sparsity, because CA is a spoken language with very limited written resources. The only available resources that is also accessible to public research were collected from previous research effort in CA linguistic tools such as a set of transcribed telephone conversations, which sums into less than 2.5 million words. Chapter 4 addressed the limited lexical coverage in CA and the use of sub-lexical units in order to limit vocabulary growth and consequently reduce the out-of-vocabulary (OOV) rate. This

chapter focuses on the data sparsity issue and explores how rich-resourced MSA can be exploited for use in CA LMs, namely Objective 2 of this thesis.

Section 5.1 starts with a discussion of reported attempts to use existing MSA textual data for development of a language model for CA. This is followed by a description of two strategies to exploit existing MSA textual resources for training CA LMs. The first strategy (Section 5.2) narrows the gap between MSA and CA through "colloquialisation", i.e. the rendering of MSA text as CA, within a statistical machine translation framework in order to generate more CA textual data. The second strategy (Section 5.3) casts MSA and CA as two different domains rather than two different dialects and tries to increase the context coverage by using paraphrastic language models. Both strategies are empirically evaluated in Section 5.4, using perplexity as evaluation metric. Finally, conclusions are drawn and the chapter is summarised in Section 5.5.

5.1 Related research

Aided by the existence of a large volume of MSA resources, several studies explored their use to enrich CA data for developing natural language processing tools. Much work explored using MSA data either directly, by finding a mapping between CA and MSA, or by parsing CA and MSA and using the syntactic and morphological level instead of or with the lexical level.

Pooling transcribed CA text with MSA data directly, such as Egyptian CA (ECA) with MSA (Kirchhoff et al., 2003) and Qatari CA (Elmahdy et al., 2013), yielded an insignificant (if any) reduction in the perplexity. Similar outcomes were observed when Kirchhoff et al. (2003) interpolated two LMs, one estimated from a small ECA training set and the other estimated from MSA data, with optimised weights even when the chosen MSA data were selected to be conversational in nature.

Alternatively, other studies attempted to transform CA into MSA in the context of statistical machine translation (SMT) due to the absence of CA-English parallel corpora. Motivated by the rich MSA-English machine translation resources, many researchers transformed CA-to-MSA (known as CA or dialect *normalisation*) in order to be able to use existing MSA resources (Bakr et al., 2008; Sawaf, 2010; Salloum and Habash, 2011; 2013; Aminian et al., 2014). For instance, Bakr et al. (2008) employed a hybrid normalisation approach to normalise ECA, which applied a combination of mapping rules and a statistical tokenising and tagging model trained on an ECA morphological lexicon. Another hybrid normalisation approach was proposed by Sawaf (2010). His normalisation method transferred CA words to MSA based on character- and morpheme-level mapping rules. Afterwords, an SMT system was used to translate from MSA to English. While Sawaf (2010)

normalised both affixes and stems to MSA vocabulary, Salloum and Habash (2011) only applied mapping rules on the affixes but also used morphological analysis information and dictionaries in addition to language models and allowed multiple morphological analyses in the form of lattices to be translated by an SMT system to English.

With the emergence of social media, more written CA can be observed where users use their own *mother-tongue*, namely CA, in conversational responses. Al-Sabbagh and Girju (2010) harvested the web for ECA and MSA lexicons while the COLABA project (Diab et al., 2010) constructed similar resources from web logs. Based on their experience, Elfardy and Diab (2012) composed a set of guidelines for constructing such resources with the aid of automatic dialect identifiers.

5.1.1 Motivation

As mentioned in Chapter 2, CA and MSA are considered two variants of the same language where each has its own functionally exclusive domain. CA is used primarily in informal daily conversations while MSA is used as the formal correspondence variant. Native Arabic speakers can easily switch between the two variants according to the situation and consequently can swap an utterance from one variant to the other. Transferring a given MSA utterance to a CA utterance is known as *colloquialisation* of MSA while the reverse process is called *normalisation* of CA.

Because MSA and CA are functionality exclusive, they could be considered two different domains or topics, such as politics and medical topics in English. The majority of the vocabulary is shared in addition to domain-specific vocabulary and syntax. Given this new analogy, rather than translating between two dialects, the main task of exploiting MSA resources for CA language modelling can be cast as closing the gap on the lexical level between two domains. Machine translation (Section 5.2) and lexical paraphrasing (Section 5.3) approaches can be employed in the colloquialisation process of MSA textual data.

5.2 Colloquialising MSA resources using SMT framework

As aforementioned, CA normalisation proved to be beneficial in machine translation tasks. In these tasks, MSA was used as a so called *pivoting* language: Normalising CA-to-MSA was performed first, which is then followed by standard MSA-to-English translation task. This approach allows using the rich resources of a MSA to train state-of-the-art SMT system.

Where a large volume of MSA textual resources exists, colloquialisation allows generating more matching data for CA language modelling by rendering existing MSA sentences

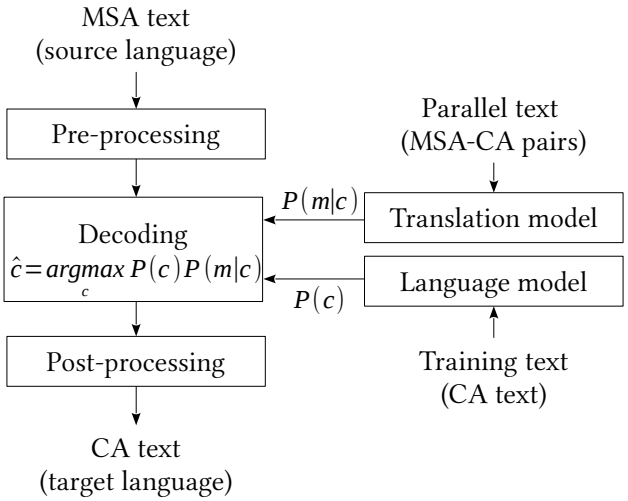


Figure 5.1: A statistical machine translation (SMT) framework for colloquialising MSA text to CA text.

into CA. Because colloquialisation is the transferring of a sentence from a source variant to a target variant, statistical machine translation (SMT) approaches can be employed for this purpose where MSA is considered as a source language and a CA dialect is the target language. In order to train an SMT model, a parallel corpus is required where each sentence is written in MSA along with its corresponding CA sentence which can be collected using manual annotation.

Statistical machine translation is based on and it tried to finds the most probable sentence in the target language, in this case in CA, given a sentence in the source language, in MSA. This problem can be defined using Bayes rule as follows:

$$\begin{aligned} \hat{c} &= \arg\max_{c} P(c|m) \\ &= \arg\max_{c} \frac{P(c)P(m|c)}{P(m)} \\ &= \arg\max_{c} P(c)P(m|c), \end{aligned} \tag{5.1}$$

where $\hat{c}$ is the most probable colloquial sentence, $P(c)$ is the language model of the CA and $P(s|c)$ is the translation model from a MSA sentence m to a CA sentence c and the whole process is known as decoding. Figure 5.1 illustrates a general SMT framework. As it shown, an SMT system is composed mainly of two main components. First component is the language model, $P(c)$, that assigns higher probability to sentences with better grammatical structure or word order. In order to estimate these probabilities, the LM parameters need

to be estimated using sentences from the target language, i.e. CA, and it is the same LM used in the ASR (Section 3.1.3). The second component is the translation model, $P(m|c)$, that assigns higher probability to CA sentences, c, which have a similar meaning as the MSA sentence, m. The parameters of a translation model requires a collection of parallel text, where each training sample is a pair of a sentence in the source language and its corresponding translation in the target language. Because there might be more than one valid word order, the translation model $P(m|c)$ is computed over all possible alignments as follows:

$$P(m|c) = \sum_{a} P(a, m|c), \tag{5.2}$$

where a represents an alignment between words in the source-target pair, and it is a hidden variable.

An SMT system is widely evaluated using the BiLingual Evaluation Understudy (BLEU) score (Papineni et al., 2002). An automatically translated text is considered better if it is close to a professional translator's outcome. Based on this idea, the BLEU score is computed from counting the number of n-grams between the translation hypothesis and one or more translation references with considering the variation in terms of word choice and ordering. Papineni et al. (2002) achieved this by using the modified n-grams precision. The modified n-gram precision, p_n , is calculated for each length n of n-grams, w^n, by summing over all matches for every translation hypothesis, S, in the whole corpora, $\mathcal{C}$; formally:

$$p_n = \frac{\sum_{S \in \mathcal{C}} \sum_{w^n \in S} \text{Count}_{\text{matched}}(w^n)}{\sum_{S \in \mathcal{C}} \sum_{w^n \in S} \text{Count}(w^n)}, \tag{5.3}$$

where $\text{Count}_{\text{matched}}(w^n)$ counts all the matched n-grams of length n between the hypothesis and some reference while $\text{Count}(w^n)$ counts all n-grams of length n in the hypothesis S. A translation hypothesis that is shorter than the reference sentences will have a higher precision score than those longer hypotheses. In order to compensate for having very short translation hypotheses, a brevity penalty, BP, is computed over the whole corpus, which is computed as follows:

$$\text{BP} = \begin{cases} 1 & \text{if } |\mathcal{C}| > r \\ e^{1-r/|\mathcal{C}|} & \text{if } |\mathcal{C}| \leq r \end{cases}, \tag{5.4}$$

where $|\mathcal{C}|$ is the length of the hypothesis corpus and r is the length of the closest reference sentences to the hypotheses. Then BLEU score is computed as follows:

Table 5.1: Example of normalising the colloquial sentence "mrp kwys" into five valid MSA equivalents.

CA sentence	MSA equivalence
mrp kwys (very good)	jyd jdA
	mmtAz jdA
	Hsn jdA
	Tyb jdA
	Hlw jdA

$$\text{BLEU} = \text{BP} \exp\left(\sum_{n=1}^{N} \lambda_n \log p_n\right), \tag{5.5}$$

where λ_n is a positive weights for each p_n such that $\sum_{n=1}^{N} \lambda_n = 1$ and it is commonly assigned to equal weights. BLEU ranges between 0 to 1 with higher scores indicating better translation hypotheses.

Unlike MSA, CA lacks standard conventions for writing colloquial words. Therefore, native Arabic writers usually improvise the spelling of such words and this leads to noisy and unreliable *colloquialised* MSA sentences. Hence, for creating a parallel CA-MSA corpus, colloquialisation of MSA data is replaced by normalisation of CA data for the consistency of annotation conventions. Normalisation of CA requires sentences in CA which can be drawn from any of the existing small CA speech transcriptions corpora (provided by LDC). A subset of these transcriptions can be selected to avoid repetitions and to insure more lexical coverage. Since CA and MSA share almost two thirds of their vocabulary (Habash and Rambow, 2006), only sentences including at least one non-MSA word are included in the chosen set. Usually, sentences in CTS corpora are short in length (between 4-6 words per sentence on average) and in undiacritised form. This imposes a challenge for the annotator to choose the corresponding MSA match that serves the intended meaning. Therefore, for an annotator to transfer a CA sentence, at least one preceding sentence and one subsequent sentence are presented as well to provide some semantic context. Because there might be more than one valid normalisation for a single CA sentence, more than one normalisation can be allowed to gain a much richer mapping between CA and MSA. For instance, the colloquial word "mrp kwys" (very good) can be rendered into five MSA sentences, as shown in Table 5.1.

Lately, crowdsourcing platforms, such as Amazon's Mechanical Turk (AMTurk), are used for collecting and annotating resources for computational linguistics (Sorokin and Forsyth, 2008; Kittur et al., 2008; Snow et al., 2008; Novotney and Callison-Burch, 2010; Paolacci et al., 2010; Kumar et al., 2014). Zaidan and Callison-Burch (2011) and Sabou et al. (2014) provided general guidelines for best practice in using such platforms in order

to obtain high quality NLP resources. Crowdsourcing allows annotation tasks to be distributed among several *non-professional* annotators by splitting them into smaller tasks, known as *mircotasks* or *human intelligent tasks* (HIT). An example of such microtask is annotating one or two sentences. A sentence can be annotated more than once by several annotators for the purpose of improving quality and consistency of the outcomes (Zaidan and Callison-Burch, 2011).

Unfortunately AMTurk is restricted for use by USA residents only; therefore, the Upwork platform was employed instead. Upwork, previously known as oDesk, is an international work platform to connect freelancers and work contractors together. Unlike AMTurk, Upwork does not scale easily to large numbers of annotators because each of them needs an individual contract before enrolling and performing any task. Nevertheless, the experience level of hired annotators is much higher in Upwork than in AMTurk. Moreover, the cost of performing the normalisation of CA using Upwork remains considerably lower than hiring professionals.

Several quality measures were used. First, enrolled annotators had to be native speakers of the presented CA dialect. Second, following the guidelines of Zaidan and Callison-Burch (2011), several control sentences for which their corresponding MSA pairs are known were presented to the annotators - at least three control sentences out of a total ten sentences were included in each job. An additional quality control procedure was to provide the source CA sentence and all its normalised variants resulting from a previous normalisation task so that any invalid normalisation variants would be rejected. If none of the normalisation variants survived, that CA sentence was flagged and returned to be normalised again.

The crowdsourced parallel corpus resulting from the normalisation and validation tasks was then used to train a colloquialisation system. The SMT model of that system was estimated from parallel text where MSA, i.e. normalised CA, was the source language and CA was the target language. Using the SMT framework, additional CA data for training n-gram LMs was generated by decoding MSA text via the estimated colloquialisation model.

5.3 The use of paraphrastic language modelling

Instead of considering CA and MSA as two different languages, one can treat them as two different domains instead as politics and medicine domains. CA is used in the conversational and informal domain while MSA is used in the formal domain where syntactic structure and some of the vocabulary differ across these domains. However, since both domains use the same language, some of the vocabulary and context are still shared across.

Such structure can be captured using Paraphrastic Language Models (ParaLMs), which

were proposed by Liu et al. (2012; 2014). In their model, multiple paraphrase variants were generated from a statistical model for each training sentence. Following the distributional theory (Harris, 1954), two sentences are considered paraphrase variants to each others if they appear in the same context, i.e. share the left and right context. Then, the ParaLM probabilities are estimated by maximising the marginal probability over all paraphrase variants. Formally, for a word sequence, $\mathcal{W}$, of length $L_{\mathcal{W}}$ words, which has a set of paraphrase variants $\{\mathcal{W}'\}$, the log probability of predicting the next word in the sequence is:

$$\mathcal{F}(\mathcal{W}) = \ln\left(\sum_{\psi,\psi',\mathcal{W}'} P(\mathcal{W}|\psi)P(\psi|\psi')P(\psi|\mathcal{W}')P_{\text{PLM}}(\mathcal{W}')\right) \tag{5.6}$$

where ψ is a phrase sequence, $P(\psi|\mathcal{W}')$ is a word to phrase segmentation model, which allows generating a single-word or multiword phrases from a single word, $P(\psi|\psi') = \prod_i P(v_i|v'_i)$ is a phrase to phrase paraphrase model to compute the probability of a phrase sequence. Here, $\psi' =< v'_1, v'_2, ..., v'_K >$ is a set of paraphrastic variants of the phrases $\psi =< v_1, v_2, ..., v_K >$, $P(\mathcal{W}|\psi)$ is a phrase to word segmentation model which converts a phrase ψ to a word sequence $\mathcal{W}$, and $P_{\text{PLM}}(\mathcal{W}')$ is the paraphrastic LM probability to be estimated. Liu et al. (2012) showed that sufficient statistics for an ML estimation of $P_{\text{PLM}}(\mathcal{W}')$ can be accumulated along each paraphrase word sequence $\mathcal{W}'$ and weighted by its posterior probability. In other words, the statistics, namely n-gram counts, that is required for predicting a word w following the history h can be computed as follows:

$$\text{Count}(h, w) = \sum_{\mathcal{W}'} P(\mathcal{W}'|\mathcal{W})\text{Count}_{\mathcal{W}'}(h, w) \tag{5.7}$$

where $\text{Count}_{\mathcal{W}'}(h, w)$ is the count of the sequence $< h, w >$ within the paraphrase $\mathcal{W}'$.

It is necessary to estimate the paraphrase model, $P(\psi|\psi')$, prior to Equation 5.6 which requires a large number of paraphrase variant pairs. These pairs are extracted statistically based on distributional similarity (Harris, 1954). Co-occurrence counts of two phrases of lengths ranging from l_{min} to l_{max} that share the same left and right context of specific length (l_{cxt}) are used to estimate the phrase paraphrase model. Based on these counts the paraphrase probabilities can be estimated as follows:

$$P(v'|v) = \frac{\text{Count}(v \rightarrow v')}{\sum_{\bar{v}} \text{Count}(v \rightarrow \bar{v})} \tag{5.8}$$

$\text{Count}(v \rightarrow v')$ is the number of occurrences where the phrase v shared the same left and right context of length l_{cxt} with the phrase v'. $\text{Count}(v \rightarrow \bar{v})$ is the number of occurrences where the phrase v shared the same left and right context of length l_{cxt} with any other

Table 5.2: Example of paraphrase variants along with their paraphrase probabilities. A colloquial word is marked by a subscript which indicates all dialects the word belongs to.

Paraphrase		English translation	$P(v'\|v)$
From, v	To, v'		
tZlwA bxyr (be safe)	xyr	good / alright	0.065
	tslmy	stay safe / thank you	0.033
	mwfq	be blessed / good luck	0.033
	bAltwfyq	with blessing / good luck	0.033
	$\text{bttwfq}_{\text{LCA}}$	wish you blessing / good luck	0.033
	$\text{Exyr}_{\text{LCA,GCA}}$	with safe / see you later	0.020
	$\text{bnfrH}_{\text{LCA,GCA}}$ lk	we are happy for you	0.020
	$\text{btqDy}_{\text{LCA,GCA}}$ rHlp mwfqp	have a safe journey	0.020
	$\text{ttnjHy}_{\text{LCA}}$	you will success	0.020
	$\text{yrbHwk}_{\text{LCA}}$	they will let you win	0.020
	Allh yHfZk	may Allah protect you	0.020
	$\text{btZlwA}_{\text{LCA}}$ sAlmyn	stay healthy	0.020
	klh xyr	all is good	0.020
	tstEmlyh bAlhnA	enjoy it with happiness	0.020

phrase that is not itself. As shown in Equation 5.8, the paraphrase probabilities, $P(v|v')$, are directed which allow the type of paraphrasing to be controlled by discarding out-of-domain target phrases v' from the paraphrase pairs prior to estimating paraphrasing probabilities. Table 5.2 lists some examples of paraphrase variants extracted from multi-dialect data for the phrase "tZlwA bxyr" (be safe), which is commonly used in CA as farewell wishes to end a conversation. Its many variants have a similar function even if exact translations are different.

Paraphrases are generated in the form of lattices where a word lattice $\mathcal{T}_{\mathcal{W}'}$ is created from each training sample $\mathcal{W}$. For efficiency, Liu et al. (2012) suggested employing a weighted finite transducer (WFST) (Mohri, 1997) framework where all the components in Equation 5.6 are represented as follows:

$$\mathcal{T}_{\mathcal{W}'} = \det\left(\pi_{\mathcal{W}'}\left(\mathcal{T}_{\mathcal{W}:\mathcal{W}} \circ \mathcal{T}_{\mathcal{W}:\psi} \circ \mathcal{T}_{\psi:\psi'} \circ \mathcal{T}_{\psi':\mathcal{W}'}\right)\right) \tag{5.9}$$

$\circ$, π and det are WFST composition, projection and determinisation operators. $\mathcal{T}_{\mathcal{W}:\mathcal{W}}$ is the original word sequence transducer, $\mathcal{T}_{\mathcal{W}:\psi}$ is the word to phrase segmentation transducer, $\mathcal{T}_{\psi:\psi'}$ is the phrase to phrase paraphrase transducer and $\mathcal{T}_{\psi':\mathcal{W}'}$ is the phrase to word transducer which is the inverse of $\mathcal{T}_{\mathcal{W}:\psi}$. Only phrases that exist in the extracted paraphrase variant set are produced and accepted by $\mathcal{T}_{\mathcal{W}:\psi}$ and $\mathcal{T}_{\psi:\mathcal{W}}$ respectively.

5.4 Experiments

In this section, the usefulness of MSA textual resources in the development of a CA LM using either a SMT-based colloquialisation system or a paraphrastic LM is evaluated in terms of word perplexity computed over a testing set. Two LCA corpora were used in most of these experiments, namely *FisherLCA* and *AppenLCA* training sets, which have been used in Section 4.5. The testing sets from both corpora are merged into one testing set *testLCA*. As for MSA resources, two training sets were used: *NW10* and *BC*, where the former is a newswire corpus based on a Lebanese newspaper while the latter is based on broadcast conversation data collected under the GALE project. All these sets are described in Appendix B.

5.4.1 Colloquialising MSA resources

Using the Upwork platform, 47 native LCA speakers were enrolled to normalise a subset from the LCA corpora, in addition to the *testLCA* set. The subset consisted of 18726 sentences with 77848 words drawn from *FisherLCA* and 11409 sentences with 77194 words drawn from *AppenLCA*. From this subset, 2000 sentences were randomly selected, rendered into MSA and kept in a control sentence pool. In addition to the control sentence pool, two sentence pools were prepared: a normalisation sentence pool and a validation sentence pool. Apart from those sentences in the control sentence pool, all sentences were initially assigned to the normalisation sentence pool.

Two tasks were prepared to be performed by the annotators. The first task was a normalisation task where an annotator was provided with a set of ten sentences in LCA to be rendered into MSA; three sentences were drawn from the control sentence pool while the rest were drawn from the normalisation sentence pool. For the second task, which was a validation task, an annotator was provided with a set of three LCA sentences along with their normalised variants and asked to reject invalid normalised variants. Sentences for the validation task were drawn from the validation sentence pool. Only sentences which were normalised by three annotators with different normalised variants were included. For a given validation sentence, if all normalised variants were rejected by an annotator, that LCA sentence was returned to the normalisation sentence pool.

In the normalisation task, annotators were asked to render each colloquial word into its MSA equivalent using an undiacritised form such that one LCA word can be rendered to a phrase of more than one word in MSA and vice versa. For instance,:

- the single-word CA phrase “ETfwltk” (on your childhood) is normalised to a multi-word MSA phrase “ElY Tfwltk”, and

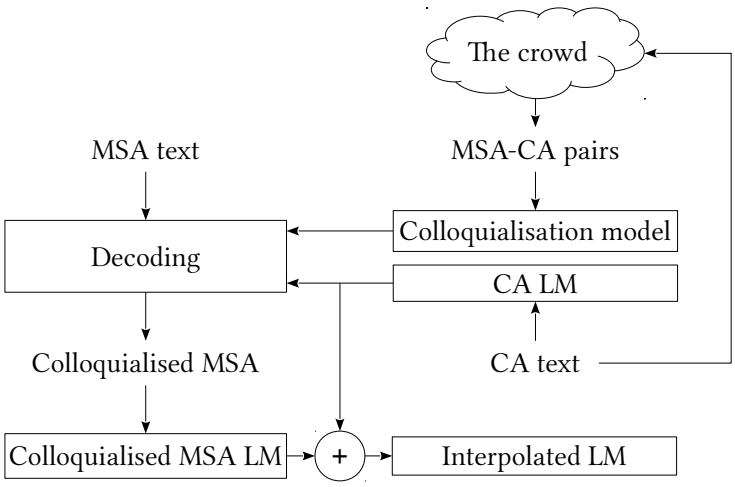


Figure 5.2: Schematic diagram for developing a language model based on colloquialised MSA text. The "+" sign indicate a linear interpolation between two LMs.

- the multi-word CA phrase "E$An hm" (because of them) is normalised to a single-word MSA phrase "l>nhm".

In addition, annotators were not encouraged to reorder the normalised phrase unless it was completely unacceptable in MSA, which is rare given the syntactic flexibility of MSA. Further details of the guidelines summary which was given to the annotators can be seen in Appendix C.

As a result, a parallel corpus of LCA-MSA data, which is a set of pairs of LCA sentences along with its normalised variants, was created. A translation model was estimated as a colloquialisation model based on the crowdsourced parallel corpus using an SMT framework, based on the Moses toolkit (Koehn et al., 2007). The source language was MSA (i.e. normalised variant) and the target language was LCA. The colloquialisation model obtained a BLEU score of 0.994 on *testLCA*.

After estimating the colloquialisation model, the MSA resources (*NW10* and *BC*) were colloquialised with the model and the Moses decoder. The resulting colloquialised MSA corpora were used to estimate a new trigram LM for each corpus. Figure 5.2 illustrates the development process for the colloquialised MSA-based LM.

The first three columns in Table 5.3a show the perplexity results on *testLCA* with one of the three LMs estimated on all LCA data, *BC* and *NW10*. Although *BC* is an MSA resource, its perplexity is equivalent to almost one tenth of that of *NW10*. This is mainly because the style of the *BC* dataset is conversational while *NW10* is intended for written media and thus has a much richer context than that of *BC*. Consequently, the *BC* LM was assigned a higher interpolation weight than the *NW10* LM when they were linearly interpolated with the *LCA* LM. The interpolated LM gave a relative perplexity

Table 5.3: Perplexity and relative difference in perplexity compared with the *LCA* trigram LM with interpolated LM with different combinations of LM components estimated on (a) MSA resources or (b) colloquialised MSA resources. If the interpolation weight is 1.0 that means there is no interpolation with any other component.

(a) LM components were estimated on MSA resources

LM component	Interpolation weights					
LCA	1.0			0.955	0.962	0.947
BC		1.0		0.005		0.028
NW10			1.0		0.038	0.026
Perplexity	213.3	2066.8	19474.4	209.2	206.9	206.0
%diff	0.0	+869.0	+9030.1	-1.9	-3.0	-3.4

(b) LM components were estimated on colloquialised MSA resources

LM component	Interpolation weights					
LCA	1.0			0.932	0.933	0.915
BC		1.0		0.068		0.033
NW10			1.0		0.067	0.052
Perplexity	213.3	1452.6	6304.7	206.5	200.4	199.5
%diff	0.0	+581.0	+2855.8	-3.2	-6.1	-6.5

Table 5.4: Relative difference in the number of n-grams (of order 1 to 3) between LMs estimated from MSA resources (baseline) and LMs estimated from colloquialised MSA resources.

Corpus	unigrams	bigrams	trigrams
BC	0.0	+1.7%	+3.6%
NW10	0.0	+6.1%	+4.9%

improvement of 1.9% and 3.0% respectively and 3.4% when both were included in the interpolation to reach a perplexity of only 206.

LMs estimated from colloquialised MSA resources showed a considerable reduction in the perplexity, especially for the *NW10* dataset, as shown in Table 5.3b. In comparison to the perplexity computed from *BC* and *NW10* (shown in Table 5.3a), a relative reduction of 30% and 68% respectively was obtained with colloquialised corpora instead of equivalent MSA data. Moreover, the obtained reduction in the perplexity resulting from interpolating the *LCA* LM and LMs estimated from colloquialised MSA resources was twice that of an interpolation with LMs estimated from MSA resources directly.

In order to further investigate the source of improvement in the LMs based on colloquialised MSA, a comparison between the number of n-grams counted in each model was

performed. Table 5.4 lists the relative difference in the number of n-grams of order 1 to 3 found in LMs estimated from MSA text and LMs estimated from colloquialised MSA text. As shown in the table, the number of both bigrams and trigrams were increased by at least 1.7% and 3.6% respectively depending on the size of the colloquialised dataset. This empirically proved that automatically colloquialised MSA text can be used as an additional resource for developing CA LMs.

5.4.2 Cross-dialect paraphrase variants

In order to evaluate the usefulness of out-of-domain resources for inducing paraphrase variants to phrases occurring in LCA data, two additional training sets were used: *GCA* and *ICA* (both are described in Appendix B). Both LCA training sets, *FisherLCA* and *AppenLCA*, were merged into the *LCA* training set to be used in the paraphrase variants induction as well as MSA training sets, *NW10* and *BC* into the *MSA* set. All words which are exclusive to CA sets, namely do not exist in *MSA* set, are considered colloquial words and were marked with the dialect they belong to. A colloquial word can be assigned to more than one CA dialect, for instance the colloquial word "Ally" (who), which appears in all CA sets but not *MSA*, was marked with *L*, *G* and *I* for LCA, GCA and ICA respectively to be "GILAlly". The chosen characters used for marking colloquial words, {L, G, I}, are not part of the Buckwlater transliteration set, and hence allow a straightforward retrieval of the original words. If all words in a phrase exist in the *MSA* set, i.e. the phrase does not have any colloquial words, it was considered to be an MSA phrase; however that does not mean it was extracted from the *MSA* data set.

Paraphrase variants were extracted from all training sets. Each variant pair shared left and right contexts of up to 3 words, i.e. $l_{\text{cxt}} = 3$. These variants could range between 1 and 4 words in length, i.e. $l_{\text{min}} = 1$ and $l_{\text{min}} = 4$. Based on these settings, a window of minimum length of 5 words, $l_{\text{cxt}} + l_{\text{min}} + l_{\text{cxt}}$, and maximum length of 10 words, $l_{\text{cxt}} + l_{\text{max}} + l_{\text{cxt}}$, was shifted over each training sentence to extract potential phrases. Table 5.5a lists the number of paraphrase variants extracted from each training set along with the distribution of the resulting variants among the included dialects. A paraphrase variant belongs to a certain dialect if there exists at least one word from that dialect in the phrase. For example, phrases from GCA (i.e. contains at least one colloquial word marked with *G*) between one to four words length was sharing their left and right contexts with 32090 other phrases from all four corpora. Of those 32090 phrases, 21.3% contain words only exist in the *GCA* set, 9.7% contain words only existing in the *ICA* set, 22.7% contain words only existing in the *LCA* set and 70.4% contain words existing in the *MSA* set. Words can belong to more than one dialect, and therefore extracted paraphrase variants can also be assigned to multiple dialects. Because of the overlap between the dialects, the

dialect percentages in each row in Table 5.5a do not sum to 100%. As shown in Table 5.5a, the majority of extracted paraphrase variants, 94.1% out of 797146, do not contain any colloquial words.

Conversational data sets included tags that correspond to non-lexical backchannel responses, such as "AhA". These non-lexical backchannel responses were produced by listeners as vocalised cues that the speaker still had their attention. Sometimes transcribers may mistakenly assign these tags to hesitation events which are considered to be speech disfluency. For that, paraphrastic extraction can be performed but all potential disfluency tags are not considered in the phrase extraction. However, they will still remain in the resulting variant set. For instance, the two phrases "lAzm nkwn mE" (we must be with) and "mvlA nqEd mE" (such as we are sitting with) cannot be extracted as paraphrase variants because they do not share left and right contexts in the following example when the word "AhA" is considered:

	left context						right context			
bs	AnA	wAnt	AhA	lAzm	nkwn	mE	AhA	bEDnA	AlbED	</s>
bs	AhA	AnA	wAnt	mvlA	nqEd	mE	bEDnA	AlbED	AhA	</s>

However, when all occurrences of the word "AhA" are discarded from the surrounding context, the two phrases are considered to be paraphrase variants, as they are supposed to be:

	left context						right context		
bs	AnA	wAnt	lAzm	nkwn	mE	bEDnA	AlbED	</s>
bs	AnA	wAnt	mvlA	nqEd	mE	bEDnA	AlbED	</s>

Table 5.5b shows the distribution of paraphrase variants by dialect when all potential disfluency tags are discarded. Although the results shown are similar to those of Table 5.5a, more than 2% of the extracted paraphrase variants differ after discarding backchannel and disfluency tags.

Since the main objective is to reduce the data sparsity problem by generating more matching data, only paraphrase variants targeting the desired dialect are kept. Tables 5.5c and 5.5d shows the distribution by dialect of paraphrase variants extracted from different source dialect but only targeting LCA dialect, when considering disfluency tags or discarding them in the extraction process respectively. In comparison to paraphrase variants extracted previously without targeting a specific dialect (Tables 5.5a and 5.5b), only 4.6% of the paraphrase variants were kept. Of that, 70% were paraphrased from phrases without colloquial words.

As described above, phrases of length between 5 to 10 words were extracted for their context (3 words on the left and 3 words on the right) to be compared against each

Table 5.5: Number of paraphrase variants, $|\{v'\}|$, induced from multiple corpora of different dialects, with different combinations of excluding disfluencies or not and paraphrasing to certain dialects only or not. Variants can have variable length between ($l_{\min}$=1) to ($l_{\max}$=4) shared a left and right context of (l_{cxt}=3) words. Also, the distribution of paraphrase variants by dialect.

(a) Including disfluencies and allowing paraphrasing to any CA dialect.

From, v	GCA%	ICA%	LCA%	MSA%	$\lvert\{v'\}\rvert$	% of Total
GCA	21.3	9.7	22.7	70.4	32,090	4.1
ICA	22.4	12.3	23.6	69.7	13,895	1.7
LCA	20.1	9.0	23.5	70.3	36,368	4.6
MSA	3.0	1.3	3.4	94.1	750,390	94.1
Any	4.0	1.7	4.6	94.1	797,146	100.0

(b) Excluding disfluencies and allowing paraphrasing to any CA dialect.

From, v	GCA%	ICA%	LCA%	MSA%	$\lvert\{v'\}\rvert$	% of Total
GCA	21.4	9.7	22.8	70.3	31,978	4.0
ICA	22.5	12.3	23.6	69.7	13,853	1.7
LCA	20.2	9.0	23.5	70.3	36,182	4.6
MSA	3.0	1.3	3.4	95.6	748,419	94.2
Total	4.0	1.7	4.6	94.2	794,962	100.0

(c) Including disfluencies and allowing paraphrasing to LCA dialect only.

From, v	GCA%	ICA%	LCA%	MSA%	$\lvert\{v'\}\rvert$	% of Total
GCA	66.8	35.8	100.0	0.0	7,296	20.1
ICA	70.2	44.5	100.0	0.0	3,274	9.0
LCA	61.6	32.7	100.0	0.0	8,562	23.5
MSA	62.6	31.9	100.0	0.0	25,565	70.3
Total	87.9	38.1	100.0	0.0	36,368	100.0

(d) Excluding disfluencies and allowing paraphrasing to LCA dialect only.

From, v	GCA%	ICA%	LCA%	MSA%	$\lvert\{v'\}\rvert$	% of Total
GCA	66.8	35.8	100.0	0.0	7,289	20.1
ICA	70.1	44.4	100.0	0.0	3,271	9.0
LCA	61.8	32.8	100.0	0.0	8,518	23.5
MSA	62.6	31.9	100.0	0.0	25,423	70.3
Any	62.5	32.1	100.0	0.0	36,182	100.0

other. Sentences drawn from spontaneous conversations tend to be shorter than those in broadcast news conversations or newswire data. For instance the average number of words for the *LCA*, *ICA* and *GCA* is between 4 to 6 words per sentence. The average is much higher in *MSA* sets at between 11 to 15 words per sentence. Therefore, Liu et al. (2014) suggested to use a shorter context for extracting more paraphrase variants whilst warning of reduced paraphrastic quality of these variants. Following their suggestion, similar paraphrase variant extractions to those shown in Tables 5.5 were performed, but with a shorter context ($l_{\text{cxt}} = 2$). These variants ranged between one to fours words in length. The number and dialect distribution of the resulting paraphrase variants are shown in Tables 5.6. When no specific dialect was targeted, the number of paraphrase variants extracted with shorter context increased by the factor of 2 compared to before. However, when targeting the LCA dialect, the number of the extracted variants with a shorter context was 30-fold more. Nevertheless, the dialect distribution of paraphrase variants remained similar across different context depths.

5.4.3 ParaLM using cross-dialect paraphrase variants

A subset was selected from the resulting paraphrase variants described in Section 5.4.2, where the paraphrase variants extraction was based on a longer left and right context of three words and excluding any potential disfluency tags. Only paraphrase variants that did not include any OOV words were kept to be used in the estimated paraphrase model using the Equation 5.8. Using a WFST-based tool implemented as part of the OpenFST library (Allauzen et al., 2007), a paraphrastic lattice was generated for each training sentence as described in Equation 5.9. Over all generated paraphrase variants, n-gram counts weighted by their posterior probabilities were accumulated, as described in Equation 5.7. Then the resulting statistics was normalised by removing all counts less than 0.001, and then the rest were rounded to the nearest integer after adding 1 to all counts. The final counts were then used to train a n-gram LM.

Based on this approach, two sets of ParaLMs were estimated individually from the previously introduced colloquial data sets (*LCA*, *GCA* and *ICA*) and MSA data sets (*BC* and *NW10*). One set only allowed paraphrasing to LCA dialect while the other did not enforce any restrictions. These will be referred to as *ParaLM.3-1-4.exH.2L* and *ParaLM.3-1-4.exH* respectively.

As a baseline LM, a standard word-based trigram LM was chosen for each data source individually. Table 5.7 compares the perplexities on *testLCA* from all estimated LMs (standard and ParaLM). As shown in the table, using any out-of-domain data source, namely non-LCA, gave high perplexity. Perplexities ranged between 1519 and 2318 when data sources had a similar speaking style (i.e. conversational style) but a different dialect,

Table 5.6: Number of paraphrase variants, $|\{v'\}|$, induced from multiple corpora of different dialects, with different combinations of excluding disfluencies or not and paraphrasing to certain dialects only or not. Variants can have variable length between (l_{min}=1) to (l_{max}=4) shared a left and right context of (l_{cxt}=2) words. Also, the distribution of paraphrase variants by dialect.

(a) Including disfluencies and allowing paraphrasing to any CA dialect.

From, v	GCA%	ICA%	LCA%	MSA%	$\lvert\{v'\}\rvert$	% of Total
GCA	20.9	15.6	24.4	72.4	594,810	3.8
ICA	23.4	19.1	26.3	70.4	397,901	2.6
LCA	13.2	9.5	20.3	78.0	1,097,074	7.1
MSA	3.0	2.0	6.0	93.6	14,304,204	92.3
Any	3.8	2.6	7.1	92.4	15,489,216	100.0

(b) Excluding disfluencies and allowing paraphrasing to any CA dialect.

From, v	GCA%	ICA%	LCA%	MSA%	$\lvert\{v'\}\rvert$	% of Total
GCA	24.5	18.5	26.8	69.4	495,329	4.1
ICA	26.3	21.6	28.2	68.0	348,040	2.9
LCA	20.2	14.9	25.2	71.8	657,382	5.5
MSA	3.1	2.1	4.2	95.2	11,211,685	93.8
Any	4.1	2.9	5.5	93.8	11,957,828	100.0

(c) Including disfluencies and allowing paraphrasing to LCA dialect only.

From, v	GCA%	ICA%	LCA%	MSA%	$\lvert\{v'\}\rvert$	% of Total
GCA	74.8	59.7	100.0	0.0	144,843	13.2
ICA	79.3	67.4	100.0	0.0	104,512	9.5
LCA	58.0	44.1	100.0	0.0	222,146	20.2
MSA	44.4	30.3	100.0	0.0	855,321	78.0
Any	47.7	33.6	100.0	0.0	1,097,074	100.0

(d) Excluding disfluencies and allowing paraphrasing to LCA dialect only.

From, v	GCA%	ICA%	LCA%	MSA%	$\lvert\{v'\}\rvert$	% of Total
GCA	79.5	64.2	100.0	0.0	132,507	20.2
ICA	82.9	71.0	100.0	0.0	98,235	14.9
LCA	70.3	55.4	100.0	0.0	165,604	25.2
MSA	61.8	45.7	100.0	0.0	472,011	71.8
Any	64.4	48.5	100.0	0.0	657,382	100.0

Table 5.7: Perplexity of trigram standard LMs (3g LM) and ParaLMs estimated from CA (LCA, GCA and ICA) and MSA (BC and NW10) corpora. ParaLM.3-1-4.exH used paraphrase variants of length one to four words sharing left and right context of three words excluding disfluency tags. ParaLMs.3-1-4.exH.2L is similar to ParaLMs.3-1-4.exH but only targeting the LCA dialect. Relative difference in perplexity of ParaLM (%diff) is computed for each row by considering the perplexity of the 3g LM in that row as baseline.

		3g LM	3g ParaLM.3-1-4.exH		3g ParaLM.3-1-4.exH.2L	
Corpus	Size (words)	Perplexity	Perplexity	%diff	Perplexity	%diff
LCA	1,906,286	213.3	774.9	+263.9	353.9	+65.9
BC	1,433,932	2,066.8	2,652.9	+28.4	2,666.0	+30.0
NW10	15,779,447	18,824.9	11,721.4	-37.7	13,587.1	-27.8
GCA	344,383	1,518.7	1,545.8	+1.8	1,846.4	+21.6
ICA	167,263	2,318.3	2,000.2	-13.7	2,680.7	+15.6

Table 5.8: Perplexity of interpolating a standard trigram LM (wLCA) with a 3g ParaLM.3-1-4.exH. Each column represents an interpolation combination where an empty cell indicates that the ParaLM was not included in the interpolation. If the interpolation weight is 1.0 that means there is no interpolation with any other component. The bottom two rows show the perplexity and relative difference from the perplexity of the first column.

LM	Interpolation weights								
wLCA	1.000	0.981	0.971	0.976	0.959	0.968	0.975	0.956	0.943
pLCA		0.018			0.012			0.011	0.008
pBC			0.029		0.016				0.013
pNW10				0.024	0.014				0.012
pGCA						0.032		0.021	0.015
pICA							0.025	0.013	0.009
Perplexity	213.3	212.4	210.2	209.7	208.7	211.3	212.0	210.6	207.8
%diff	0.00	-0.44	-1.46	-1.69	-2.14	-0.92	-0.63	-1.28	-2.56

and reached up to 18825 when data sources differed in style (i.e. written style) and dialect as well (MSA). Using the ParaLM did not help to improve the perplexity for the conversational-based LMs whereas it showed an improvement of at least 28% relative for the written-based LM.

An improvement in perplexity can be observed when ParaLMs were interpolated with a standard LM. Table 5.8 compares the perplexity on *testLCA* of the models from interpolating the standard trigram LM (*wLCA*) and a combination of ParaLMs with different weights estimated using *ParaLM.3-1-4.exH* configurations. In comparison to no interpolation ($\lambda_i = 1$), a relative improvement in the perplexity of different degrees was observed. When training ParaLM without any restrictions on the targeted dialect, interpolating with any MSA-based ParaLM (*pBC* or *pNW10*) achieved the lowest perplexity among interpo-

Table 5.9: Perplexity of interpolating a standard trigram LM (wLCA) with a 3g ParaLM.3-1-4.exH.2L. Each column represents an interpolation combination where an empty cell indicates that the ParaLM was not included in the interpolation. If the interpolation weight is 1.0 that means there is no interpolation with any other component. The bottom two rows show the perplexity and relative difference from the perplexity of the first column.

LM	Interpolations weights								
wLCA	1.0	0.981	0.967	0.972	0.928	0.969	0.978	0.931	0.924
pLCA		0.018			0.041			0.046	0.031
pBC			0.033		0.014				0.013
pNW10				0.028	0.017				0.016
pGCA						0.031		0.017	0.012
pICA							0.022	0.007	0.005
Perplexity	213.3	212.4	210.0	209.1	207.9	211.6	212.2	210.7	207.3
%diff	0.00	-0.44	-1.55	-1.98	-2.52	-0.80	-0.53	-1.21	-2.82

Table 5.10: Relative difference in the number of n-grams found across several ParaLMs when paraphrase variants targeting LCA and when no specific dialect is targeted. All these ParaLMs were estimated using the same vocabulary list.

ParaLM	%diff when use LCA-targeted ParaLM		
	unigrams	bigrams	trigrams
pLCA	0%	-47.2%	+79.4%
pBC	0%	-50.2%	+74.6%
pNW10	0%	-51.6%	+48.7%
pGCA	0%	-34.1%	+38.1%
pICA	0%	-32.9%	+32.5%

lating with single ParaLM with a relative improvement of 1.5% to 1.7% while interpolating with any colloquial ParaLM obtained between 0.4% to 0.9% relative reduction in perplexity. Interpolating both MSA-based ParaLMs achieved 2.14% improvement while interpolating all colloquial-based ParaLM obtained 1.3% relative reduction. The best result was achieved when all ParaLMs were interpolated with 2.6% relative perplexity reduction to reach a perplexity of 208.

When using LCA-targeted ParaLMs, further perplexity improvements were observed. Table 5.9 shows several combinations of LCA-targeted ParaLMs to be interpolated with standard word-based trigram LM (*wLCA*). The improvement was exclusive to MSA-based ParaLMs, between 1.6% and 2.0% relative individually and 2.5% relative when interpolating both *BC* and *NW10*. Colloquial-based ParaLMs did not yield any improvement. However, interpolating all ParaLMs achieved the best improvement with 2.8% relative. Although the paraphrase model for LCA-targeted ParaLMs was estimated from a small fraction of the overall extracted paraphrase variants, namely 5% as shown in Tables 5.6b

and 5.6d, they outperformed ParaLMs estimated from paraphrase variants without any restrictions on the targeted dialect. As shown previously in Table 5.5a, LCA paraphrase variants represents only 4.6% of all extracted variants. Because of that, LCA paraphrase variants had very low paraphrastic probability estimates (computed from Equation 5.8) as the majority of these variants are non-LCA. Consequently, paraphrase variants with very low paraphrastic weighted counts (less than 0.001 when computed from Equation 5.7) are considered infrequent and were discarded before training the language model. However, the majority of these counts were not filtered out when restricting paraphrase variants to target only LCA. Table 5.10 shows the relative difference between the number of n-grams of order 1 to 3 found in ParaLMs estimated using the same vocabulary list. As shown in the table, the number of trigrams generally increases by at least 33% for non-LCA colloquial ParaLMs and reached almost 80% for the LCA ParaLM. The exclusion of non-LCA-targeted paraphrase variants, which constituted 70% of all extracted paraphrase variants, affected the number of bigrams in the estimated ParaLMs because CA dialects mostly share low order n-grams with MSA, but not higher n-grams.

5.5 Summary and conclusion

MSA textual data have been used in previous studies in developing CA LMs using two main approaches: First, pooling CA text with MSA data to be used in estimating a CA LMs; second, an LM was estimated on each Arabic variant individually and then both LMs are linearly interpolated with optimised weights. Both approaches reduced perplexity only insignificantly. None of the approaches perform any kind of processing on the MSA data prior to using it in LM training.

As aforementioned in Section 3.1.5, perplexity can be interpreted as a measure on the average of the number of equally most probable words that can follow any given word in the language. Therefore, lower perplexity indicates better language models, in other words, less confused models. The main objective of this chapter was to investigate the exploitation of the large volume of existing MSA textual data for developing CA LMs, and to address the CA data sparsity issue in order to improve the overall perplexity of the language model, namely Objective 2 of this thesis. Two strategies were introduced to serve this purpose.

In the first strategy, MSA was colloquialised into CA using a colloquialisation model within an SMT framework. MSA was cast as a source language and CA as a target language. Such transforming from MSA to CA has not been employed before for language modelling purposes and only the reverse direction has been explored for machine translation purposes. As a translation model,the colloquialisation model had to be estimated using a

parallel corpus. For the consistency of MSA annotating convention, a parallel corpus was created by normalising a subset of a CTS transcription data set using a crowdsourcing platform. Following the suggested quality assurance guidelines in the literature, every CA sentence was normalised by more than one annotator, followed by a validation task in case of the existence of conflicts in the normalisation. Estimated from the resulting parallel corpus, the colloquialisation model was used with an SMT decoder to colloquialise (i.e. translate) MSA resources into CA. The colloquialised MSA data was then used to estimate standard trigram LMs. These outperformed LMs estimated from the MSA data with a perplexity reduction up to 68% relative. Moreover, the perplexity reduction obtained from interpolating these colloquialised MSA LMs was twice that obtained from interpolating MSA LMs to reach 6.5% relative in comparison to the baseline CA LM.

Rather than casting MSA and CA as two different dialects, the second strategy considered them as two different domains. Each domain uses a different syntactic structure, which was captured by ParaLMs. First, paraphrase pairs were extracted from different corpora of different dialects including MSA. For conversational speech, it was shown that excluding disfluency words (such as hesitation and backchannel markers) can improve the quality of the induced paraphrase pairs. A relative perplexity reduction of between 1.5% and 1.7% was obtained when using ParaLMs estimated from MSA resources which was outperformed by ParaLMs targeting specific CA dialects to reach a 2% relative perplexity reduction. Further improvement was achieved when interpolating ParaLMs with standard LMs, giving a 2.8% relative reduction.

Comparing the two strategies, language models which were developed based on colloquialised MSA performed better than those based on ParaLM. This can be accounted to the filtering process in the developing ParaLM which does not exist in the colloquialisation process and consequently the latter yielded more training samples.

In contrast to previous studies, using the proposed methods in this work allows the use of MSA textual resources to reduce the data sparsity issue and improves the performance of an estimated LM on a CA test set in terms of perplexity.

Chapter 6

Explicit modelling of short vowels in colloquial Arabic ASR

Contents

Chapter 4 and Chapter 5 pursued objectives related to language modelling while this chapter concentrates on the automatic generation of the missing diacritics, and pursues Objective 3 of this thesis.

In order to train acoustic models for ASR, speech segments along with their transcription are employed. The provided transcription must use the same units which are targeted to be modelled, such as words or phonemes, if the training is intended for word-based or

Table 6.1: An example of several diacritised forms for the Arabic word "drs" written in Buckwalter transliteration. The provided diacritisations exclude the last vowel because it is syntactically decided in MSA and essentially absent in CA.

Undiacritised form	Diacritised form	Phonetic transcription	Meaning	Part-of-speech
drs	dars	/dars/	lesson	indefinite noun
	daras	/daras/	he studied	active perfect verb
	dar~as	/dar:as/	he taught	active perfect verb
	duris	/duris/	has been studied	passive perfect verb
	dur~is	/dur:is/	has been taught	passive perfect verb
	dar~is	/dar:is/	teach	imperative verb

phoneme-based acoustic models respectively. If transcriptions are not written using the same units used for acoustic models, a pronunciation dictionary is required to map between the written form and the acoustic model form. The Arabic language has no pronunciation dictionary because a fully diacritised Arabic text encodes most of the phonetic values explicitly. However, one of the main issues in acoustic modelling of Arabic is the absence of diacritics, especially those representing short vowels, from written form. Most training resources are undiacritised and given that diacritics constitute around 30% of written units overall, this means that one third of the spoken sounds are not represented.

For the English language, a standard dictionary provides a phonetic transcription of a given word, unlike an Arabic dictionary which only provides acceptable diacritised variants for a given word where the appropriate variant is chosen depending on the context. These diacritised variants are widely considered to be phonetic transcriptions. Pronunciation is dependent on the diacritised form of a given word, and thus it is crucial to restore these missing vowels for an ASR task, especially for acoustic and pronunciation modelling. One way to restore missing vowels is by including all possible vowelised versions of a word in a dictionary as options; however, this will expand the pronunciation variations exponentially with the number of consonants, where not all diacritised forms are legitimate. For instance, the Arabic word "drs"[1] might have at least six acceptable diacritised forms, shown in Table 6.1, which define its exact meaning and pronunciation. However, in an extreme case, a 3-consonant word could have up to 64 diacritised variations[2]. In addition to their significance in pronunciation modelling, these diacritics proved to be of use within other natural language processing (NLP) tasks, such as morphological disambiguation (Habash and Rambow, 2005), information retrieval (Hammo et al., 2008) and machine translation

[1] All Arabic words are written in Buckwlater Arabic transliteration scheme (Buckwalter, 2002; 2004a).

[2] The number of diacritised variants is computed through permuting the seven vowel-related diacritics (a,i,u,~a,~i,~u,vowel-less) in two possible positions (excluding the last vowel since it is syntactically decided in MSA and almost essentially absent in CA, as discussed previously in Section 2.6).

(Diab et al., 2007).

This chapter is organised as follows. Section 6.1 describes previously proposed methods to restore missing short vowels in Arabic generally and CA specifically. Motivated by the Arabic consonantal and vowelic (CV) skeleton, a short vowel insertion model is introduced to represent a generic vowel model in Section 6.2 as an exploration of unsupervised short vowel modelling. This is followed by an investigation of two proposed methods for restoring short vowels in CA automatically. The first method casts the problem as a grapheme-to-phoneme conversion (Section 6.3). Speaker-related information is introduced as an additional stream of information to aid diacritics prediction in Section 6.4. A second novel approach is proposed in Section 6.5 to capture longer spans of context and to incorporate more auxiliary information to support the prediction process using a conditional random field (CRF) model. These methods were experimentally tested and their results are reported in Section 6.6, along with a comparison to an MSA disambiguation tool, MADA. Finally, conclusions are given in Section 6.7.

6.1 Related research

Recovering missing diacritics (short vowels, gemination and nunation) before acoustic model training has proved to be beneficial in several studies in Arabic ASR during the last decade. If recovery concentrates on retrieving short vowels only, the process is known as vowelisation or vowel restoration, otherwise it is known as diacritisation or diacritic restoration. Unfortunately, this distinction is not always clear within these studies as vowelisation and diacritisation are used interchangeability in the literature. Gemination has been neglected in most diacritisations for acoustic modelling purposes because it only represents a longer duration (which is implicitly modelled) for the associated phoneme without adding any different phoneme, unlike short vowel diacritics and nunations where they indicate the existence of a hidden acoustic value from the associated consonant. Moreover, most of the studies discussed below provide their results exclusive and inclusive of the last diacritic. This last diacritic indicates the case of the word and usually is decided syntactically. Since CA lost the case-ending vowel during its evolution, only results excluding the case-ending diacritic are discussed below.

Diacritisation can be either manual or automatic. Manual diacritisation is provided by language natives and theoretically should be considered to be accurate. Since fully diacritising Arabic text is a very uncommon and unnatural process, it is expensive in terms of time and labour. In addition, it is prone to errors especially for CA and loanwords where there are no clear standards agreed upon. Hence, the need for automatic diacritisation emerged. Automatic diacritisation has been investigated using several approaches: rule-,

morphologically-, example- and statistically-based methods. Rule-based methods have not been pursued in the recent studies because of their difficulties to transfer to other dialects and the expense of maintenance, so the focus on the other approaches has increased.

The literature discussed below has formulated the diacritisation problem as other NLP or machine learning problems, such as tagging, sequence labelling or machine translation (MT) tasks where the graphemic representation was considered as input in the process either on its own or incorporated with additional information such as morphological or syntactic analysis. In the literature, diacritisation was processed in three different modes: character, word and hybrid or hierarchical mode. The latter is merging character and word modes together. For the character-based mode, each character (consonant or long vowel) is considered as an input unit, whereas in word mode a whole undiacritised word is considered as an input unit. A hybrid mode combines the two, mostly starting with one mode and transfering to the other based on a certain condition.

One of the first attempts in diacritising MSA text used a rule-based diacritiser (El-Sadany and Hashish, 1988) with syntactic rules in order to diacritise a given stem. In their subsequent work, prefixes and suffixes used a fixed diacritised form (El-Sadany and Hashish, 1989). El-Imam (2004) provided grapheme-to-phoneme and phoneme-to-phone rules in the context of speech synthesis. In addition to the requirement of linguistic expertise to develop these rules, they are difficult to maintain and alter to cope with new dialects.

In contrast, Gal (2002) proposed a statistical approach using a bigram HMM where unvowelised words are the observations and vowelised words are the hidden states of the model. This model was tested using a fully diacritised Quran text and achieved 14% diacritised word error rate (DWER)[3]. This approach was successfully applied by Elshafei et al. (2006) on news and articles in MSA and with a wider context, up to quad-gram, in their subsequent work (Alghamdi and Muzaffar, 2007). Nelken and Shieber (2005) employed Gal's approach in a weighted finite state transducers (WFST) framework in diacritising a broadcast news Penn Arabic Treebank corpus (Maamouri et al., 2004). Their system was composed of multi-level diacritisers (word, morpheme and character) and yielded a DWER of 18.0% and DER of 5.5%. Recently, Hifny (2012a; 2012b; 2013) scored word sequences using an n-gram LM, which was trained on diacritised text, and built a word lattice from the generated sequences. Smoothing techniques such as Katz and modified Kneser-Ney were applied for the unseen sequences, then dynamic programming (DP) was used to find the most likely sequence in the lattice. This approach was tested on 1.9 million words using a LM trained on 5.25 million words (all harvested from the Internet) and yielded a

[3]Diacritised word error rate (DWER) is the percentage of words among a given test set which have at least one incorrectly predicted diacritic, while diacritic error rate (DER) is the percentage of predicting a diacritic incorrectly.

best performance of 3.4% DWER when using quad-gram LM and absolute discounting.

A hierarchical example-based approach was adopted by Emam and Fischer (2004) where a sentence is processed in a top-down manner to search for a matching sentence in a diacritised lexicon. If a sentence was not found, the search backed off to look for individual words. If a word was not found in the lexicon, a character-based n-gram model was used to diacritise each word. In similar fashion, Ananthakrishnan et al. (2005) included both an undiacritised word and all the diacritised candidates (retrieved from BAMA) in LM training. In the case of unseen words, the system backed off to a character-based LM which was trained on a fully diacritised text. Another hybrid system was developed by Rashwan et al. (2009; 2011) where a dictionary was used to find a diacritised form of a given undiacritised word and build a lattice of all possible variations of a given sentence. This was then disambiguated using a language model and A* lattice search to infer the most likely diacritised sequence. If a word was not found in the dictionary, a morphological analyser was employed to factorise a given word to its components (prefix, stem and suffix) and use them directly in the lattice before disambiguation.

Conversely, Zitouni et al. (2006) formulated Arabic diacritisation as a sequence labelling problem and proposed a maximum entropy (MaxEnt) framework solution. The model features were generated based on morphological and syntactical analysis and were used as contextual information in estimating the probability of a diacritic of a character, given the context. Habash and Rambow (2007) trained multiple support vector machines (SVM) taggers on richer morphological information to choose the best diacritised form from a set of candidates generated by a language model. Their method, known as MADA, outperformed that of Zitouni et al. by 30.4% relative on DWER and 12% relative on DER. In a similar multi-stage procedure, Shaalan et al. (2009) combined a lexicon search with a bigram estimation and an SVM classifier. The features for the SVM classifier were generated from automatic segmentation, chunk parsing and POS tagging. Shaalan et al. used a unique subset of the Penn Arabic Treebank. Therefore, it is not possible to compare their results with the work cited above. Like Zitouni et al. (2006), Schlippe et al. (2008) cast the diacritisation problem as one of sequence labelling. In their work, they employed a conditional random field (CRF) model to predict a sequence of diacritics based on the conditional probability of the sequence of consonants given a sequence of contextual and global features. Those features included POS tags and full words. Their system achieved between 9.4-8.4% DWER and 2.2-1.9% DER on the Penn Arabic Treebank corpus where prediction accuracy improved as the amount of context considered was increased.

Within the same work, Schlippe et al. (2008) compared their CRF-based system results with another approach that considered diacritisation as a statistical machine translation (SMT) problem, where the undiacritised text was considered as a source language and the

diacritised text as a target language. Their system involved translating monotonically on the word level, i.e. an undiacritised word to a diacritised word, or on character level, i.e. a consonant to a pair of consonant and diacritic. The SMT-based diacritiser outperformed their CRF-based method with a 21% relative improvement on DWER and a 15% relative improvement on DER. Hattab and Hussain (2012) combined an SMT-based diacritiser with an additional diacritiser that employed morphological and syntactical analysis information in the process when training the system on CLA. As would be expected, their results showed that testing on CLA words (matching the same training domain) outperformed testing on MSA words, derived from news and magazines, showing 60% relative DER improvement.

Most of these approaches were devised utilising the existence of proper resources such as annotated data, dictionaries and morphological analysers. However, when the same problem is presented in the absence of such resources, as in diacritising CA text, limited improvement on the diacritisation performance, if any, has been achieved. The first automatic diacritiser for CA was developed by Kirchhoff et al. (2003) where diacritisation rules were derived from paralleled ECA transcription in an example-based approach, achieving a 16.6% DWER. Vergyri and Kirchhoff (2004); Kirchhoff and Vergyri (2004) interpreted the diacritisation task as an unsupervised tagging problem. A trigram tagger was trained on morphological tags provided by BAMA on MSA broadcast news transcriptions. Then, a word network was constructed of all possible diacritised forms, again provided by BAMA, by which the scores produced by the tagger serve as transition weights. To obtain a diacritised transcription, acoustic information was incorporated by applying forced alignment using the word network and acoustic models trained on a diacritised ECA CallHome corpus. Using the word network without tagging weights achieved a significant improvement over sole acoustic alignment, with a DWER of 27.3% being reported for ECA. In Vergyri et al. (2005) a quad-gram character-based tagger was trained on a carefully selected and manually diacritised small subset of an LCA corpus, and was used to predict the missing diacritic. On a held-out set the DWER was 30%.

Instead of identifying the exact identity of a vowel, Lamel et al. (2007) suggested using generic vowels (GV) instead of true short vowels, for example the fully diacritised form of the word “drs” would be “d@r@s@” where the symbol @ denotes a GV model. To represent the condition of vowel absence, all possible permutations were generated from the inclusion of GV models at all positions to having no vowels at all. For example, {d@r@s@, d@r@s, d@rs@, dr@s@, d@rs, drs@, dr@s, drs} are all the possible diacritised variants for the word “drs” which were used as pronunciation variants in Lamel et al.’s work and obtained slightly worse performance that using true vowels models by 5.3% relative WER.

Table 6.2: An example of applying three different morphological templates to the consonantal Arabic root [d r s] shown in the first row, the second row shows the applied templates while the derived words and their meaning are shown in the third and fourth rows and their CV-skeleton equivalences in the last row.

Root	$C_1C_2C_3$ = d r s		
Pattern	C_1aC_2aC_3	C_1AC_2iC_3	maC_1C_2aC_3ap
Word	daras	dAris	madrasap
Meaning in English	learned/studied	learner/student	school
CV-skeleton	CVCVC	CVCVC	CVCCVCVC

6.1.1 Motivations

The simplicity in Lamel et al.'s generic vowel is compelling since it does not require additional linguistic resources such as diacritisers or pronunciation dictionaries. As the authors accounted this marginally lower recognition performance to the multiple pronunciation variants in the dictionary, a modified version of their generic vowel model is proposed and discussed in Section 6.2 by which only one pronunciation is generated for each undiacritised entry.

As aforementioned, diacritisation methods can be grouped into character-based, word-based or hybrid/hierarchical-based which merge character and word modes together. In most cases, diacritisation is started on a word level, while in case of data sparsity diacritisation is performed on character level. Based on this observation, it would be preferable to employ character-based diacritisers for CA due to its inconsistencies and limited available training resources. Using acoustic information in diacritisation significantly improves prediction performance (Vergyri and Kirchhoff, 2004; Kirchhoff and Vergyri, 2004). Given that most of the consonantal graphemes have a one-to-one relation to their phoneme, some of the acoustic properties can be represented through the phonological place-voice-manner (PVM) information and can be used as auxiliary information for the diacritisation process. In addition, none of the cited work incorporates any extralinguistic information in the diacritisation process, especially speaker-related characteristics, such as gender and dialect. The use of these last two characteristics is further investigated in Section 6.5 and compared with a G2P-based method (discussed in Section 6.3) empirically in Section 6.6.

6.2 Generic short vowel (GV) model

As previously discussed in Chapter 2, Arabic words are generally derived from a consonantal root by applying vowelic templates. Table 6.2 lists three examples for such a process and it shows, on the last row, that almost every consonant is followed by a vowel regardless of the vowel identity and duration (except the last consonant which is controlled by the

Table 6.3: Percentage of different arrangement of a consonant (C) and a vowel (V) in four different Arabic dialects: Gulf (GCA), Iraqi (ICA), Levantine (LCA) and standard Arabic (MSA). CV is a consonant followed by a vowel, VC is a vowel followed by a consonant, VV is two successive vowels and CC is two successive consonants.

Dialect	CV	VC	CC	VV
LCA	27.91	49.14	22.63	0.32
ICA	28.24	49.05	22.35	0.36
GCA	27.85	48.69	22.98	0.48
MSA	33.19	44.56	21.66	0.60
Average	29.30	47.86	22.41	0.44

case-ending in MSA and is mostly vowel-less in CA). To support this observation empirically, Table 6.3 compares the number of subsequent consonant and vowel permutations derived from telephone conversation transcriptions of at least 20 hours in four different Arabic dialects (GCA, ICA, LCA and MSA). As it shows, the overall average of different combination of consonant (C) and vowel (V), i.e. (CV%+VC%), is 78%. In other words, more than three-quarters of the fully diacritised Arabic text has a consonant followed by a vowel or vice versa. Motivated by this observation, a generic vowel can be inserted after each consonant within a given word to capture some of the vowelic essence in the language regardless of its real identity. However, there are some cases where a consonant is vowel-less, an average of around 22% (see Table 6.3), such as in the word "madrasap" shown in the last column of Table 6.2. To overcome this, this generic model should allow the possibility of being neglected, either explicitly by not including the model or implicitly within the model itself. The former solution had been explored by Lamel et al. (2007), where multiple pronunciation variants were generated by removing one generic vowel at a time. However, empirical results showed that using a single pronunciation is preferable in ASR lexicons (Hain, 2002), so implicit omission should be employed. Figure 6.1 illustrates the suggested HMM topology for this generic vowel model where a direct transition exists from the non-emitting start state to the non-emitting exit state. Another distinction between this and Lamel et al. (2007)'s work is that the GV model was initially trained on a small amount of diacritised data, then was used in unsupervised training on undiacritised data. Conversely, no prior training is performed for the GV model in this thesis for the acoustic mismatch between the manually diacritised and the undiacritised data sets where the former contains high level of background noise.

6.2.1 Results and discussion

Two sets of acoustic models are trained based on the standard recipe described in Appendix B, using the AppenLCA corpus. The first set, *diac*, used true vowel models whereas the

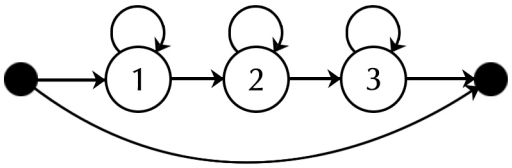


Figure 6.1: Generic vowel model (GV) topology. A standard 3-state HMM with a skip from the start non-emitting state to end non-emitting state to model the absence of the short vowels.

Table 6.4: Pronunciation entries for the words in the utterance "$w >xbArk" (How are you?) when using GV model, *gv*, or ture vowels, *diac*.

Baseform	Pronunciation	
	gv acoustic set	*diac* acoustic set
$w	sh @ w	sh u w
>xbArk	ea @ x @ b @ A @ r @ k	ea a x b a A r a k

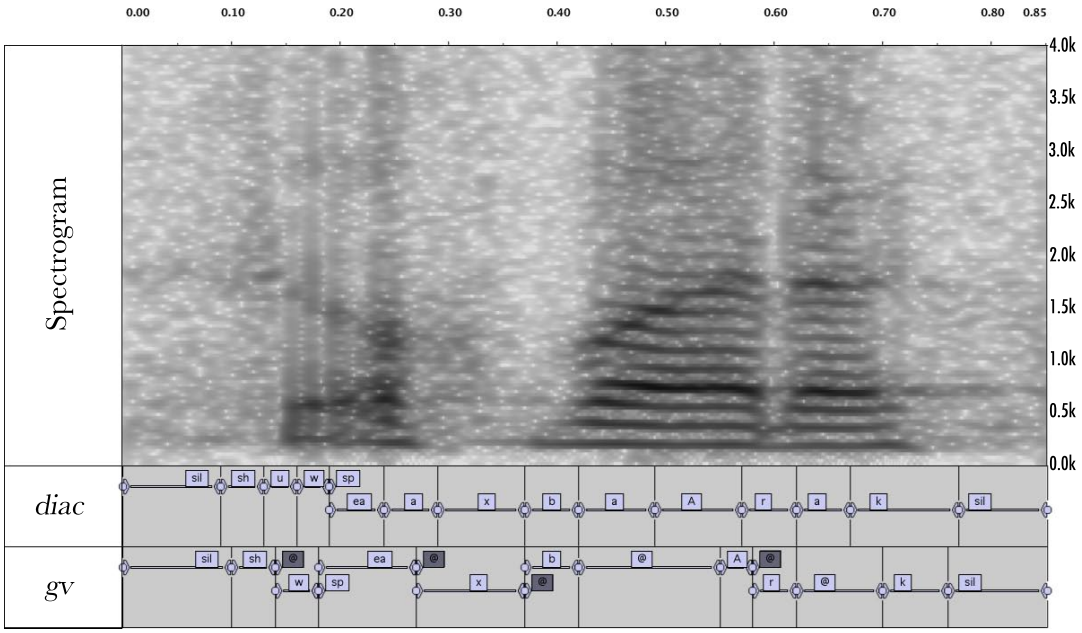


Figure 6.2: Spectrogram and phoneme labelling for the utternace "$w >xbArk" using two acoustic model sets: *diac*, containing graphemes and true vowels, and *gv*, containing graphemes and a generic vowel model (denoted with @). Some occurances of @ have a zero duration (colored in grey) and some are longer than the expected length for a short vowel. The *sp* model is a word boundary indicator that does not hold any acoustic value i.e. has a zero duration.

second set, *gv*, used only a GV model to represent all short vowels. The pronunciation dictionary for *gv* is generated in a straightforward manner where a GV model, denoted as @, is inserted between every two graphemes within a given word. The pronunciation dictionary used for *diac* includes only the seen diacritisation variants in the training data which were manually generated by the corpus collectors. As a result, only one pronunciation variant is generated per word for *gv* opposed to 1.26 pronunciations per word when true vowels are used. Table 6.4 lists pronunciations for the undiacritised utterance "$w >xbArk" (How are you?) when GV model or true vowels are used:

Here, the pronunciation dictionaries are used for both training and decoding. Using forced alignment on the word-level with these pronunciation dictionaries, a comparison between the phoneme segmentation using *gv* and *diac* is closely examined. Figure 6.2 visualises the spectrogram of the utterance "$w >xbArk" from the AppenLCA test set along with two tracks of labelling. The first labelling track is based on the phoneme segmentation using *diac*, containing all graphemes and true short vowels, and the second is based on *gv*, containing all grapheme and a GV model (denoted as @ symbol) to represent all short vowels. Some of the @ models have been skipped, marked with grey shade in the figure, indicating that no short vowel is required at this specific position. By comparing the two tracks in the figure, the following is model-level alignment (all models with a duration of zero are excluded):

diac	sil	sh	u	w	ea	a	x	b	a	A	r	a	k	sil
gv	sil	sh	w		ea		x	b	@	A	r	@	k	sil

It can be seen that the sequences [u w] and [ea a] in the *diac* labelling track are replaced by [w] and [ea][4] in *gv* labelling track. Because GV model represents three different vowelic values ([a],[i] and [u]), it has more general probability distribution than that of another model which is always assigned to one of the three vowels. An example is the model "w" which has two phonemic values ([u:] and [w]) so if a frame is actually a [u] phoneme, the model "w" is more representative than a GV model. This explains the reason for skipping a GV model at these locations. Another observation that *seems* to contradict the previous remark is that in the sequence [@ A] more frames are assigned to the GV model than to the long vowel A (see Figure 6.2). This is due to the fact that the proportion of short vowel [a] is more than 50% of all diacritics and around 17% among all graphemes in the AppenLCA training corpus. This distribution under the GV model can be more biased toward the vowel [a], as opposed to [i] and [u]. However, when the GV model appears between two consonantal graphemes and there is an actual hidden vowel in between, the

[4]Although in the actual alignment it is represented by [@ w] and [ea @] but the GV models, @, have a duration of zero which means they have not been used.

GV model is not skipped and always captures the vowelic frames, as in the sequence [r a k] and [r @ k].

A unit count of the aligned test set using *gv* shows that 22.2% of the presented GV models are skipped during the alignment, representing either the absence of vowels in these locations or that they were either preceded or followed by a much stronger and specialised vowel model. This percentage is very close to the actual percentage of the consonants sequence (CC) shown in Table 6.2. That indicates that the GV model has captured the vowelic essence by which when it is not available, the GV model is skipped and it is reflecting the actual behaviour statistically.

A fundamental trade-off between the simplicity of using GV in pronunciation generation and the richness of the local acoustic context was observed. Such that the systematic insertion of a GV model after every grapheme normalises the immediate context (around the model) and makes it less informative (all grapheme are surrounded by GV models) in comparison to using true vowels instead. Therefore, longer context should be included, such as quadphones and quintphones, to obtain informative context from a given phoneme sequence. Unfortunately, modelling such context requires more data and longer utterances which are not available for CA training data.

The resulting ambiguity can be solved by predicting the identity of each vowel, i.e. diacritisation or vowelisation, rather than using a generic model which is used merely as a vowelic place holder. In the following section automatic diacritisation is explored in order to design a better phonetic representation for acoustic and pronunciation modelling.

6.3 Grapheme-to-phoneme (G2P) based diacritisation

Automatic diacritisation had been explored mostly as a preprocessing step for a written sentence in the context of text-to-speech (TTS), such as that of Gal (2002), Nelken and Shieber (2005) and Habash and Rambow (2007). However, as discussed above, there have been some attempts in using diacritisation as a pronunciation generation method using pronunciation rules (Afify et al., 2006). Statistical joint-sequence modelling was successfully employed in grapheme-to-phoneme (G2P) conversion but limited research using statistical joint-sequence modelling for diacritisation has been reported.

In this section, G2P-based diacritisation is introduced using joint-sequence modelling, where the concept is introduced first as it is used in G2P conversion tasks (Section 6.3.1). This is followed by a description on how it can be adopted to solve the diacritisation problem (Section 6.3.2). Finally, a discussion based on diacritisation results is presented in Section 6.3.3.

6.3.1 G2P conversion

Grapheme-to-phoneme is the task of finding the most likely pronunciation, a sequence of phonemes $\mathbf{d} = \{d_1 d_2 ... d_n : d_i \in D\}$, for a word, given its written form, a sequence of graphemes $\mathbf{g} = \{g_1 g_2 ... g_m : g_j \in G\}$, where D and G are phoneme and grapheme sets respectively. This problem can expressed formally as:

$$\mathbf{d} = \arg\max_{\mathbf{d}' \in D} \quad P(\mathbf{g}, \mathbf{d}') \tag{6.1}$$

Many machine learning techniques have been exploited to design G2P converters, such as neural networks (Sejnowski and Rosenberg, 1988), decision trees (Dietterich and Bakiri, 1995), classification and regression tree (CART) (Breiman et al., 1984) and generalized decision trees (Vazirnezhad et al., 2005). All these techniques predict a phoneme by considering a grapheme along with some contextual information, which can be the surrounding graphemes and/or phonological structures. Other techniques, such as hidden Markov models (HMMs) (Taylor, 2005), considered the previously predicted phonemes as an additional input stream in order to predict the current phoneme. Another approach, known as pronunciation by analogy (Dedina and Nusbaum, 1991; Yvon, 1996), generates pronunciation for an unseen word by computing similarities or distances, such as Levenshtein distance, between the given word and words, or parts thereof, in an existing dictionary. Pronunciation fragments are then concatenated together into a new pronunciation. Most of the state-of-the-art G2P converters use a joint-sequence modelling approach (Deligne et al., 1995; Bisani and Ney, 2002; Chen et al., 2003) which pairs sequences of graphemes with sequences of phonemes and aligns them together.

In joint-sequence modelling, the G2P problem is solved in two main stages. First, training samples (pairs of grapheme sequences and phoneme sequences) are aligned or co-segemented where both grapheme and phoneme sequences are segmented into equal number of segments. For example, the alignment for the English pair (speech,/spi:tʃ/) is as follows:

speech		s	p	ee	ch
spi:tʃ	=	s	p	i:	tʃ

,

where the pairs (s,/s/), (p,/p/), (ee,/i:/) and (ch,/tʃ/) are known variously as joint-multigrams (Deligne et al., 1995), grapheme-to-phoneme correspondences (Galescu and Allen, 2001), graphonemes (Vozila et al., 2003) or graphones (Bisani and Ney, 2002). In each graphone, a sequence of one or more graphemes, is joined with a sequence of one or more phonemes and sometimes one of the sequences would be empty. There might be more than one co-segmentation/alignment for a given pair of grapheme and phoneme

sequences. Therefore, the joint probability $P(\mathbf{g}, \mathbf{d})$ is computed by summing all different co-segmentations:

$$P(\mathbf{g}, \mathbf{d}) = \sum_{\mathbf{q} \in S(\mathbf{g}, \mathbf{d})} P(\mathbf{q}) \tag{6.2}$$

where $\mathbf{q}$ is a graphone sequence using a certain co-segmentation and $S(\mathbf{g}, \mathbf{d})$ is the set of all possible co-segmentations of $\mathbf{g}$ and $\mathbf{d}$. Then graphone sequences form the new domain of symbols which are learnt using n-gram methods, such as maximum likelihood (ML) expectation-maximisation (EM) training (Deligne et al., 1995). So for a sequence of graphones, $\mathbf{q}$:

$$\begin{aligned} P(\mathbf{q}) &= \prod_{i=1}^{N} P(q_i | q_{i-1}, q_{i-2}, ..., q_1) \\ &= \prod_{i=1}^{N} P(q_i | q_{i-n+1}, q_{i+2}, ..., q_i - 1) \end{aligned} \tag{6.3}$$

where n is the depth of the history to be considered in the prediction process and N is the length of $\mathbf{q}$.

6.3.2 Diacritisation using G2P converter

By analogy, the diacritisation problem can be formulated as a G2P conversion problem where the targeted diacritised word is the phoneme sequence, $\mathbf{d}$, and the given undiacritised word is considered as grapheme sequence, $\mathbf{g}$. The set of Arabic diacritised graphemes, D, contains all graphemes and diacritic symbols whereas diacritics are excluded in the Arabic undiacritised grapheme set, G. For example, the Arabic pair (HDrtk,HaDratak) would have the following graphone sequence:

$$\begin{array}{|c|} \hline \text{HDrtk} \\ \text{HaDratak} \\ \hline \end{array} = \begin{array}{|c|} \hline \text{H} \\ \text{Ha} \\ \hline \end{array} \begin{array}{|c|} \hline \text{D} \\ \text{D} \\ \hline \end{array} \begin{array}{|c|} \hline \text{r} \\ \text{ra} \\ \hline \end{array} \begin{array}{|c|} \hline \text{t} \\ \text{ta} \\ \hline \end{array} \begin{array}{|c|} \hline \text{k} \\ \text{k} \\ \hline \end{array},$$

where each grapheme in the undiacritised sequence is aligned with either a sequence of grapheme and diacritic or a grapheme only to indicate the absence of diacritics in the diacritised grapheme sequence.

6.3.3 Results and discussion

In order to evaluate the prediction performance of G2P-based diacritiser, the AppenLCA test set was used. It consists of 8491 tokens with 2302 unique words. Since the G2P-based diacritiser considers a word list and not a set of sentences, diacritisation errors are weighted

Table 6.5: G2P model training resource statistics, showing the total number of words and diacritised variants; and the percentage of diacritics observed among the total number of graphemes in each source.

Source	Vocabulary	Diacritised variants	%Diacritics					
			a	i	u	F	K	N
AppenLCA	30347	40480	17.3	11.1	4.2	0.0	0.0	0.0
BAMA	48707	68920	15.9	9.3	4.6	0.0	0.0	0.0

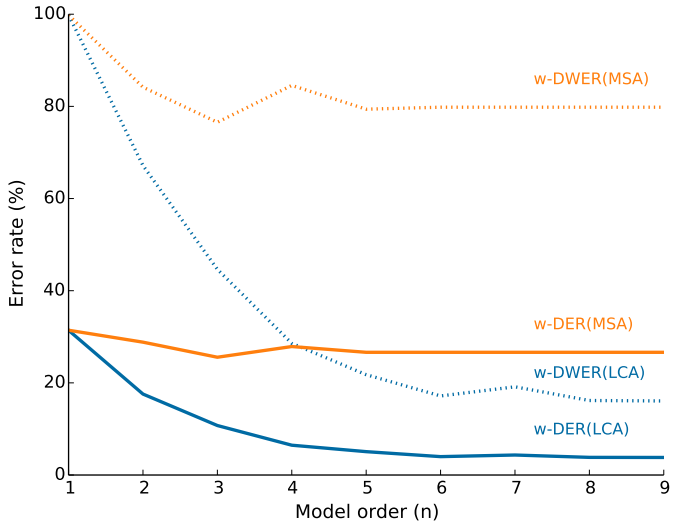


Figure 6.3: Diacritisation error rates (y-axis) using G2P convertion among different n-gram lengths (x-axis). w-DER and w-DWER are weighted diacritic errors (dotted line) and weighted diacritised word errors (solid line), respectively. *g2pLCA* is trained on LCA samples (blue) and *g2pMSA* is trained on MSA samples (orange).

by the word frequency to reflect how much of the test set would be incorrectly diacritised. These are weighted DWER (w-DWER) and weighted DER (w-DER). w-DWER and w-DER are variants of DWER and DER respectively where the errors are weighted by the word frequency within the test set.

Two different G2P converters were trained, using the Sequintur G2P toolkit (Bisani and Ney, 2008), which is based on the joint-sequence modelling discussed above. The first G2P converter was trained on training samples derived from AppenLCA with an average of 1.33 diacritised variants per word. Since some diacritised resources for MSA data exist, another G2P converter was trained on samples derived from BAMA, which provided all possible diacritised variants for more than $48{\times}10^3$ MSA stem words, giving an average of 1.41 diacritised variants per word. Table 6.5 shows the distribution of diacritics within each resource along with the exact number of diacritised forms and their undiacritised words.

Figure 6.3 summarises the results of diacritising the word list of the AppenLCA test set, weighted by word frequency. In the figure, the n-gram depth is shown on the x-axis,

with the error rate on the y-axis. Generally, adding more context increased prediction accuracy, especially when using lower n-gram G2P models. Although both diacritisers have similar performance when using unigram G2P, the gain obtained from adding more context differs enormously. While moving from unigram G2P to 5-gram G2P improved the performance of *g2pLCA* by a 84% relative w-DER and a 78% w-DWER, it only improved the accuracy of *g2pMSA* by a 15% relative w-DER and 20% w-DWER. This confirms that MSA diacritisation patterns differ from their CA counterparts and this difference becomes more evident in wider context.

While *g2pMSA* performance stabilised and did not improved beyond a context of length four, *g2pLCA* continuously improved with increased context. The lack of improvement in higher order *g2pMSA* was most likely caused by the nature of training samples which are derived from the BAMA diacritised stems list, where words tend to be shorter (without being concatenated to any affixes). Hence the model backs off to lower order due to data sparsity. Therefore, one would expect to observe a change in the performance when longer words are included in the training data.

To measure the impact of the training set size, the AppenLCA training set is divided randomly into 10 disjoint subsets where all diacritisation variants of the same word are within the same subset, and where each subset resembles the diacritic distribution of the whole set. Ten G2P models are trained based on accumulated training subsets, i.e. the first G2P model is trained on one set, the second is trained on two sets and so on. To reduce the effect of irregularities in the training subsets, the experiment was repeated with different arrangements of subsets inclusion, and the two sets of results were averaged. As illustrated in Figure 6.4, increasing the training samples from $8{\times}10^3$ to $80{\times}10^3$ does not improve the overall performance on either weighted metric, w-DWER or w-DER, an indication of consistency in the G2P performance (Schlippe et al., 2012).

The promising performance of which method proved that the local mapping between an undiacritised word and its diacritisation variants can be learnt.

6.4 Extralinguistic information and diacritisation

Context does not only imply the graphemeic sequence but also other high-level information. A *Local* mapping is learnt using context within a word, but not across neighbouring words; therefore, it does not guarantee accurate diacritisation when moving from one word to the next one. The *global* context and characteristics are derived from a complete sentence and are usually used by native Arabic speakers to disambiguate different diacritised forms.

Sociophonetic studies showed that extralinguistic factors might have a significant impact on production and perception of the vowels; e.g. Al-Wer (2002) and Abudalbuh

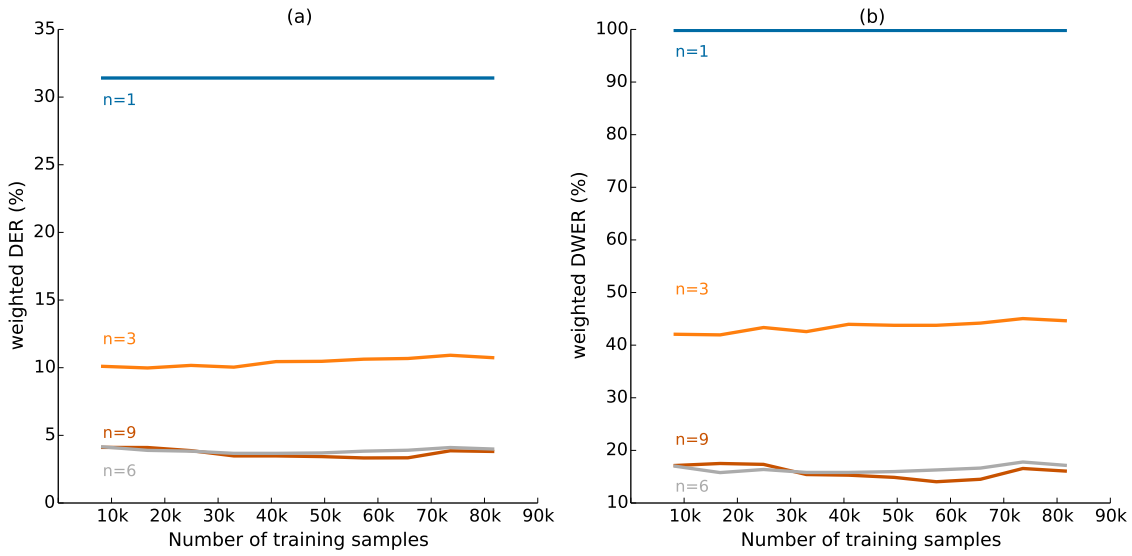


Figure 6.4: G2P-based diacritisation performance trained on LCA words, with different orders (n=1, 3, 6 and 9), and different training set sizes (x-axis). Performance is measured in (a) w-DER and (b) w-DWER.

(2010). Extralinguistic factors are the remaining speech qualities when communicative (verbal and emotional) information is removed from the signal. Extralinguistic information includes speaker gender, dialect and age. For example, when the speaker dialect is known, it would be possible to predict the pronunciation of a word, which is reflected by the chosen diacritisation, thus reducing the number of possible pronunciation variants and the overall ambiguity during the process. For instance the Arabic word "Enb" (grapes) is pronounced by an Iraqi speaker as /ʕanab/, which is reflected in the diacritisation variant "Eanab". A Levantine speaker would pronounce it as /ʕinab/ which is written as "Einab".

Most of the high-quality corpora provide such demographic information about the speaker participating in their corpus; however, the overall objective of autodiacritisation is to predict diacritics with limited resources or with features that can be generated automatically. Many studies showed successful gender and dialect detection (with average accuracy more than 85%) (Parris and Carey, 1996; Biadsy et al., 2009b) but with less success in age detection (Schuller et al., 2013). It is possible to generate such information automatically as well.

6.5 Conditional random field (CRF) based diacritisation

High-level information such as extralinguistic information about the speaker's gender and dialect, or even some articulation information about how certain phonemes are pronounced, could be useful in predicting the current vowel (i.e. appropriate diacritic). Unfortunately

such wide context and multiple dependencies cannot be captured efficiently using HMMs; therefore, other models are explored, such as conditional random fields (discussed in the next section), to incorporate wide context and additional features.

Conditional random fields are probabilistic graphical models introduced by Lafferty et al. (2001) which model the conditional distributions, $P(\mathcal{Y}|\mathcal{X})$, between two sets of random variables where $\mathcal{X}$ is a set of input random variables and $\mathcal{Y}$ is a set output random variables. CRFs are a special case of Markov random fields (MRF). This section gives a brief introduction to graphical models which are designed for sequence labelling, with a focus on three well-known examples: hidden Markov models (HMMs), maximum entropy Markov models (MEMMs) and conditional random fields (CRFs).

6.5.1 Directed and undirected graphical models

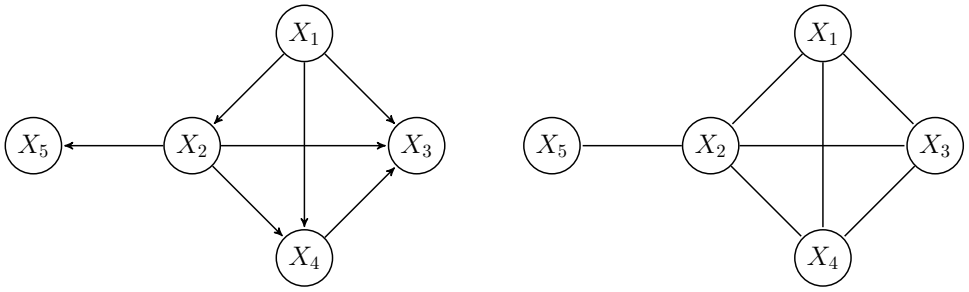
Graphical models (Pearl, 1988) are probabilistic models that can efficiently model the joint probability distributions, or multiple dependencies, between random variables. As their name indicates, graphical models are represented by means of a graph. A graph is a visual representation that is composed of nodes and edges, where each node represents a random variable and an edge denotes a statistical dependency between the two nodes at its end. Graphical models decompose the joint probability distribution between given variables into a product of smaller probability distributions which are independent of each other. This process of decomposing a joint probability distribution into independent subgraphs is known as factorisation. Graphical models can be either directed or undirected which differ in their graphical structure and inference methods.

Directed graphical models, also known as Bayesian networks, are described using a directed graph $\mathcal{G} = (\mathcal{X}, \mathcal{E})$, where $\mathcal{X} = \{X_1, ..., X_N\}$ are the graph vertices or nodes, N is the total number of nodes in the graph and $\mathcal{E} = \{(X_i, X_j) : X_i, X_j \in \mathcal{X}; i \neq j\}$. Each node, $X_i \in \mathcal{X}$, represents a random variable and each edge, $(X_i, X_j) \in \mathcal{E}; i \neq j$, denotes a dependency between the two random variables X_i and X_j. The joint probability distribution of a directed graphical model is described as:

$$P(x_1, ..., x_n) = \prod_{i=1}^{N} P(x_i|\mathcal{P}_i) \tag{6.4}$$

where x_i is the value taken by the random variable X_i and $\mathcal{P}_i$ denotes all values taken by the parent nodes[5] of the node representing the variable X_i. An example of a Bayesian network of five variables is shown in Figure 6.5a. Using Equation 6.4, the joint probability of the graph in Figure 6.5a is computed as:

[5]A parent node is the source of a directed edge.



(a) Directed graph, aka. Bayesian networks.

(b) Undirected graph, aka. Markov networks.

Figure 6.5: Examples of directed and undirected graphical models

$$\begin{aligned} P(x_1, x_2, x_3, x_4, x_5) &= P(x_1|\mathcal{P}_1)P(x_2|\mathcal{P}_2)P(x_3|\mathcal{P}_3)P(x_4|\mathcal{P}_4)P(x_5|\mathcal{P}_5) \\ &= P(x_1)P(x_2|x_1)P(x_3|x_1, x_2, x_4)P(x_4|x_1, x_2)P(x_5|x_2). \end{aligned} \quad (6.5)$$

In other words, Equation 6.5 simplifies, or factorises, finding the joint probability distribution $P(x_1, x_2, x_3, x_4, x_5)$ into a product of five smaller conditional probability distributions: $P(x_1)$, $P(x_2|x_1)$, $P(x_3|x_1, x_2, x_4)$, $P(x_4|x_1, x_2)$ and $P(x_5|x_2)$.

In a similar notion, undirected graphical models, also known as Markov random fields (MRFs) or simply Markov networks, are described using an undirected graph, $\mathcal{G} = (\mathcal{X}, \mathcal{E})$, where $\mathcal{X} = \{X_1, ..., X_N\}$ are the graph vertices or nodes and $\mathcal{E} = \{(X_i, X_j) : X_i, X_j \in \mathcal{X}; i \neq j\}$. Each node, $X_i \in \mathcal{X}$, represents a random variable and each edge, $(X_i, X_j) \in \mathcal{E}; i \neq j$, denotes a relation between the two random variables X_i and X_j. All nodes connected directly to a given random variable, X_i, are its neighbours, denoted as $\mathcal{N}_i$. A fully connected set of variables, $X_c \subseteq \mathcal{X}$, are known as a clique and $\mathcal{C}$ represents a set of all cliques in a graph $\mathcal{G}$. For example, the graph in Figure 6.5b has five cliques of size 1, $\{X_1\}$, $\{X_2\}$, $\{X_3\}$, $\{X_4\}$ and $\{X_5\}$, seven cliques of size 2, $\{X_1, X_2\}$, $\{X_1, X_3\}$, $\{X_1, X_4\}$, $\{X_2, X_3\}$, $\{X_2, X_4\}$, $\{X_2, X_5\}$ and $\{X_3, X_4\}$, four cliques of size 3, $\{X_1, X_2, X_3\}$, $\{X_2, X_3, X_4\}$, $\{X_1, X_2, X_4\}$ and $\{X_1, X_3, X_4\}$, and only one clique of size 4, $\{X_1, X_2, X_3, X_4\}$. This last clique is the maximum clique in the graph, representing the largest subset of nodes that are fully connected with every other node in the subset, whereas a maximal clique is a clique that cannot be extended to a larger clique such as $\{X_2, X_5\}$ and $\{X_1, X_2, X_3, X_4\}$. The joint probability distribution described by MRFs is decomposed into factorised local conditional distributions as follows:

$$P(x_1, ..., x_n) = \frac{1}{Z} \prod_{c \in \mathcal{C}} \psi_{X_c}(\mathbf{x}_c), \quad (6.6)$$

where $\mathbf{x}_c$ are values of a clique c, ψ_{X_c} is a positive real-value of potential function which is concerned with only a subset of the random variables, X_c, and Z is a normalising partition factor to ensure the validity of the probability distribution, i.e. $\sum_{X_i} P(X_i) = 1$, then:

$$Z = \sum_{X_i \in \mathcal{X}} \prod_{c \in \mathcal{C}} \psi_{X_c}(\mathbf{x}_c). \tag{6.7}$$

The potential function, ψ , in Equation 6.6 can be expressed as an exponential instead:

$$\psi_{X_c} = \exp \phi_{X_c}(\mathbf{x_c}). \tag{6.8}$$

Then, Equation 6.6 can be rewritten by substituting the potential function by its definition in Equation 6.8 and rearranging the equation to have the exponential of sums instead of product of exponentials as follows:

$$\begin{aligned} P(x_1, ..., x_n) &= \frac{1}{Z} \prod_{c \in \mathcal{C}} \exp \phi_{X_c}(\mathbf{x}_c) \\ &= \frac{1}{Z} \exp \sum_{c \in \mathcal{C}} \phi_{X_c}(\mathbf{x}_c) \end{aligned} \tag{6.9}$$

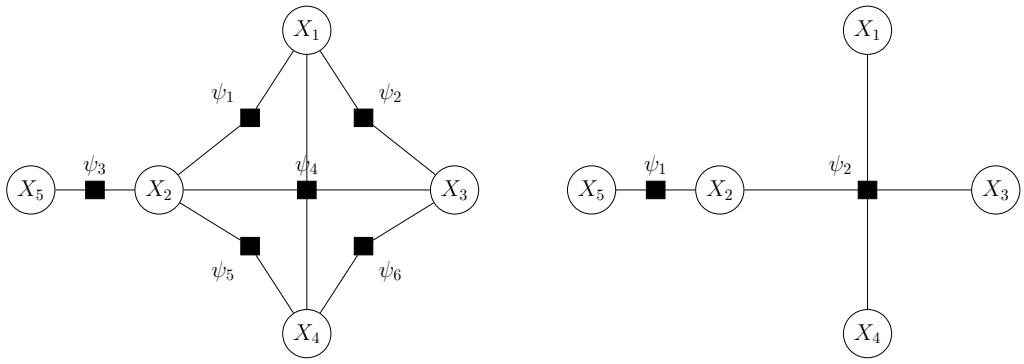
In other words, it is possible to define the probability encoded within MRFs by local functions which are concerned with only smaller subsets of the random variables by which the probability is factorised over the graph $\mathcal{G}$.

MRFs satisfy the Markovean property by which a random variable, X_i, is conditionally independent from all other variables in the graph given its neighbours, $\mathcal{N}_i$, thus:

$$P(x_i | x_j, i \neq j) = P(x_i | x_{\mathcal{N}_i}), \tag{6.10}$$

where x_j are values of the random variables $X_j \in \mathcal{X}$ and $j \neq i$. In this sense, a random variable's neighbours are referred to as its Markov blanket. It is sufficient to define potential functions for a set of all maximal cliques in the graph to represent the probability distribution described by a given graph. For example, $\mathcal{C} = \{\{X_2, X_5\}, \{X_1, X_2, X_3, X_4\}\}$ for the graph in Figure 6.5b.

The factorised probability distribution, described above, can be represented graphically by using a factor graph (Kschischang et al., 2001), $\mathcal{G} = (\mathcal{V}, \mathcal{F}, \mathcal{E})$, as illustrated in Figure 6.6a and Figure 6.6b, where $\mathcal{V}$ and $\mathcal{E}$ are, as previously defined, the random variables and their dependencies respectively whereas the new set $\mathcal{F}$ represents factor functions. Unlike an undirected graph of MRFs where factorisation is implicitly modelled via cliques, a factor graph introduces factor function nodes to denote factorisation explicitly. As shown



(a) Factor functions represents all possible cliques.

(b) Factor functions represents only maximal cliques.

Figure 6.6: Two possible factorisations for the graph shown in Figure 6.5b

in Figure 6.6a , the factor functions encode all possible cliques while the factor functions of Figure 6.6b only encode maximal cliques.

Using a graphical modelling framework as discussed, it is possible to model the probability of complex dependencies within a set of random variables. In sequential classification tasks, the random variables set, $\mathcal{X}$, is divided into two disjoint subsets: input variables, $\mathcal{O}$, and output variables, $\mathcal{S}$ where each output variable is associated with a label $l \in L = l_1, l_2, ..., l_Q$. It is necessary to predict the labelling values assigned to an output sequence, $\mathbf{s} = s_1, s_2, ..., s_T$, given the observations of input variables, $\mathbf{o} = o_1, o_2, ..., o_T$ where T is the sequence length of both input and output sequences.

Graphical models can be either generative or discriminative depending on the type of probability distribution to be described. In a generative model, modelling the joint distribution, $P(\mathbf{o}, \mathbf{s})$, is required whereas discriminative models only model the conditional distribution $P(\mathbf{s}|\mathbf{o})$ which is similar to the decoding task. Considering the input and output variables disjoint sets, Equation 6.10 can be written as follows:

$$P(\mathbf{s}_i|\mathbf{o}, \mathbf{s}_j, i \neq j) = P(\mathbf{s}_i|\mathbf{o}, \mathbf{s}_{\mathcal{N}_i}), \tag{6.11}$$

Conditional random fields (CRFs) are discriminative undirected graphical models and were proposed by Lafferty et al. (2001) to solve the sequence labelling problem. Like any undirected graphical model, its structure can be arbitrary; however, only one specific structure is considered here where sequences of input and output variables are in parallel, known as linear chain CRFs (illustrated in Figure 6.7c). In this case, only dependencies between all the input sequence, $\mathbf{o}$, and at most two adjacent output labels, s_{t-1} and s_t[6], are

[6]where t indicates current variable

considered. It is known as first-order Markov linear-chain CRFs[7]. Log-potential functions of CRFs, ϕ_t, are defined in terms of edges and vertices:

$$\phi_t = \sum_{i,e\in c} \lambda_i f_i(e, \mathbf{s_e}, \mathbf{o}, t) + \sum_{j,v\in c} \mu_j g_j(v, \mathbf{s_v}, \mathbf{o}, t)), \tag{6.12}$$

where f is a transition feature function which is applied on the edges within a given clique, c, whereas g is a state feature function that is applied on the nodes within the same clique c, while λ and μ are the weights of f and g, respectively. The possibility of using a whole input sequence in the prediction process allows the capture of a long span of variations and dynamics within that sequence. With linear chain CRFs (illustrated in Figure 6.7c), two cliques are defined for each current output variable: two successive output variables, $\{s_{t-1},s_t\}$, and the current state with all the input sequence, $\{s_t, \mathbf{o}\}$. Thus, using linear chain CRFs, Equation 6.12 is written as follows:

$$\phi_t = \sum_{i} \lambda_i f_i(s_{t-1}, s_t) + \sum_{j} \mu_j g_j(\mathbf{o}, s_t)). \tag{6.13}$$

Decoding finds the best sequence of output variables, $\hat{\mathbf{s}}$, given the observations, $\mathbf{o}$:

$$\hat{\mathbf{s}} = \arg\max_{\mathbf{s}} P(\mathbf{s}|\mathbf{o}) = \arg\max_{\mathbf{s}} \exp \sum_{t=1}^{T} \phi_t, \tag{6.14}$$

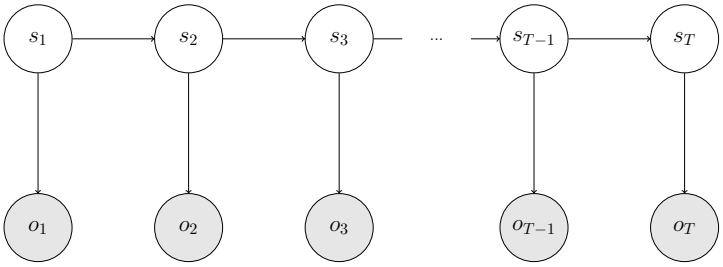
which can be performed using a dynamic programming algorithm, i.e. Viterbi algorithm. Estimation of the parameters λ_i and μ_j in Equation 6.13 is performed by searching for the best set of weights $\{\boldsymbol{\lambda}, \boldsymbol{\mu}\}$ to maximise the regularised log-likelihood function, $\mathcal{L}$, of a given set of N observation and output sequences:

$$\mathcal{L} = \sum_{k=1}^{N} \log\left(P(\mathbf{s}_k|\mathbf{o}_k)\right) - \sum_{i} \frac{\lambda_i^2}{2\sigma^2} - \sum_{j} \frac{\mu_j^2}{2\sigma^2} \tag{6.15}$$

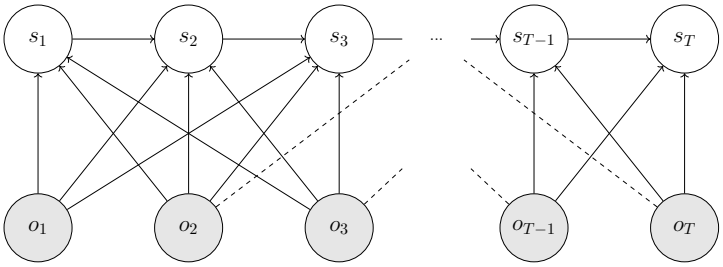
where σ are regularisation parameters. Equation 6.15 is a convex function, thus it can be maximised using iterative numerical optimisation methods such as conjugate gradients (Hestenes and Stiefel, 1952) or limited memory quasi-Newton procedures (L-BFGS) (Nocedal, 1980).

CRFs differ from Hidden Markov models in graphical structure and estimation procedure significantly. HMMs are generative directed graphical models which describe the joint probability over state sequences and output sequence. Their graphical structure is depicted in Figure 6.7a where each state, s_i, can be connected with two adjacent states,

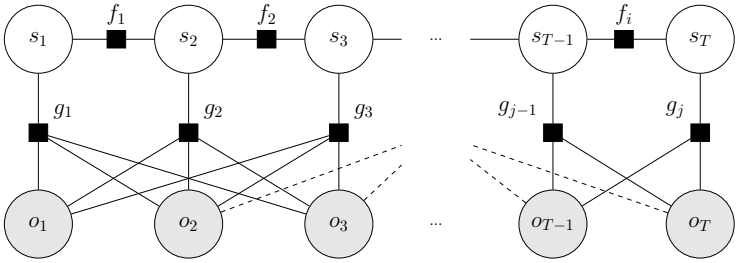
[7]Higher k^{th}-order CRFs can be modelled where the dependencies between the input sequence $\mathbf{x}$ and $k+1$ output labels are considered.



(a) HMMs as a directed graph.



(b) MEMMs as a directed graph.



(c) Linear-chain CRFs as an undirected factor graph.

Figure 6.7: Graphical structure of three examples of first-order Markov graphical models for sequence labelling tasks: hidden Markov models (HMMs), maximum entropy Markov models (MEMMs) and linear-chain conditional random fields (CRFs). Each node represents a random variable while edges denote a probablistic dependency between the connected nodes. In each model, the top layer of nodes represents the output labels while the bottom layer (coloured in gray) represents the observed input data.

s_{i-1} and s_{i+1}, and an output, o_i, by which the current state is conditionally independent of the graph by its previous state while the current output is conditionally independent of the graph by the current state. HMM parameters, θ, are estimated using maximum likelihood estimation (ML) where the estimated parameters, $\hat{\theta}$, are found by:

$$\hat{\theta} = \arg\max_{\theta} p_{\theta}(\mathbf{x}, \mathbf{y}), \tag{6.16}$$

in other words, it is necessary to enumerate over all observation sequences to find the maximum likelihood. Therefore, using a long span of dependencies or multiple features is intractable, while in CRFs it is possible to model such dependencies since it does not attempt to generate the input sequence but only the conditional probability of the output labels given the input sequence.

In contrast to HMMs, and similar to CRFs, maximum entropy Markov models (MEMMs) describe the conditional probability of the output sequence given an input sequence. MEMMs are discriminative directed graphical models where each state is an exponential model that takes the whole input sequence and outputs a distribution over the next state. While CRF is a single exponential model for the joint distribution for a whole output sequence given an input sequence, MEMMs use an exponential model, at each state, for the conditional probabilities of the next state given the current state. Therefore, a system prefers paths with fewer competing transition, so it gives that path a higher probability than a path with more competing transitions regardless of the observed input. This problem is known as a labelling bias problem. Owing to the global normalisation (by the partition function Z in Equation 6.6), there are no constraints over the sum of the outgoing mass from a certain state to the next.

6.5.2 Diacritisation using CRFs

On the one hand, using HMMs can be used to find the best state sequence, given the input sequence, but modelling rich features and long dependencies into the inference process might be intractable. On the other hand, MEMMs can incorporate rich and long term features but, due to the labelling bias problem discussed above, they might not find the best sequence, given the input sequence. CRFs addressed the labelling bias problem by global normalisation. They model the conditional probabilities of the output given the input sequence rather than modelling the joint probability distribution between the two sequences. Hence, it is possible to design different level of features that can interact with each other without the Markovian assumption of their independence.

Using linear chain CRFs, the diacritisation problem can be formulated as follows: let S be a finite set of output variables or states where each state is associated with a diacritic

$d \in \{d_1, d_2, ...d_D\}$ and a given sequence of graphemes $\mathbf{c} = c_1, c_2, c_3, ..., c_n$. Using Equation 6.14, the objective is to find the best sequence of states, $\hat{\mathbf{s}}$, that maximises the conditional likelihood:

$$\hat{\mathbf{s}} = \arg\max_{\mathbf{s}} P(\mathbf{s}|\mathbf{c}), \tag{6.17}$$

where the state sequence probability estimate is given by:

$$P(\mathbf{s}|\mathbf{c}) = \frac{1}{Z} \exp \sum_{t=1}^{T} \sum_{i} \lambda_i g_i(s_{t-1}, s_t) + \sum_{j} \mu_j f_j(s_t, \mathbf{c}), \tag{6.18}$$

where f_i and g_j are binary state and transition feature functions respectively, λ_i and μ_j are their corresponding weights. f_i and g_j are defined as follows:

$$f_i(s_t, \mathbf{c}) = \begin{cases} 1 & if\ \delta(\mathbf{c}, t) = 1\ and\ s_t = d \\ 0 & otherwise \end{cases} \tag{6.19}$$

$$g_j(s_{t-1}, s_t) = \begin{cases} 1 & if\ s_t = d\ and\ s_{t-1} = d' \\ 0 & otherwise \end{cases} \tag{6.20}$$

where $\delta(\mathbf{c}, t)$ in Equation 6.19 is a logical function that indicates whether or not the observation in location t holds a certain property, and $s_t = d$ in Equation 6.19 and Equation 6.20 means that diacritic d is associated with the state s_t. For example, $\delta(\mathbf{c}, t) : [c_t == "x"]$ asks whether the observation at index t, c_t, equals the value "x". The weights are the model parameters as described above; which are estimated by maximising the likelihood function with respect to the training data. The decoding process is performed using the Vitirbi algorithm as aforementioned.

Diacritics do not depend only on the sequence of graphemes (Vergyri and Kirchhoff, 2004), other characteristics can contribute to prediction (discussed in Section 6.4). The majority of these properties are to be found in the wider context, and some are even speaker dependent. In this thesis, properties employed are categorised into two groups: lexical-level and speaker-level properties.

Lexical-level properties include the grapheme sequence ($\mathbf{c}$), including a symbol for word boundaries, as well as a combination of articulation place, manner and voicing (PVM) along with some graphemeic grouping ($\mathbf{p}$) such as whether the grapheme is assimilated if it is preceded by the definitive "Al", i.e. solar, or not, i.e. lunar. Although PVM properties define phonological characteristics, they are considered as lexical properties because most of the graphemes, especially consonants[8], have a one-to-one mapping with their correspond-

[8]With some exceptions which are to be further discussed in Chapter 7.

ing phoneme. In case a grapheme is associated with more than one phoneme, such as the grapheme "p" (which can pronounced as /h/, /t/ and /a/), the PVM of the phoneme with higher frequency is assigned to the grapheme. A final attribute, denoted as *group*, is added to indicate if a grapheme belongs to a certain homographemic group. A homographemic group is a set of graphemes that are either share the same phoneme with or are commonly misspelled with each other. For instance, the grapheme "Y" (pronounced as /a:/) is commonly misspelled as the grapheme "y" (pronounced as /i:/ or /j/) in printed media. Table 6.6 lists the values of **p** property assigned to each Arabic grapheme.

Speaker-level properties are the speaker's dialect (d) and gender (g) which are derived either from the corpus demographic information or by means of automatic dialect and gender identification.

In the sequence labelling process, the diacritiser outputs one of six diacritics (all possible diacritics except gemination since it does not expose a hidden acoustic value as aforementioned) in addition to a no-diacritic label for each grapheme, i.e. $d \in \{a, u, i, F, K, N, o\}$. A full list of possible values for each property when considering diacritising an LCA text is listed in Table 6.7 while Table 6.8 demonstrates the use of these properties to describe the phrase "masaA Alxayr" (good evening). It should be noted that within a segment, properties **c** and **p** are more dynamic and change in the short term, while properties **d** and **g** are mostly static for the whole segment.

6.5.3 Results and discussion

It has been shown that speaker dialect, regardless of gender, has an impact on pronunciation. So it would be expected that incorporating the dialect attribute in addition to the textual sequence would have a significant impact on the diacritisation performance. To evaluate this, a set of experiments is conducted with different combinations of properties in the diacritisation process. Here, a CRF-based diacritiser with second-order Markov dependencies is designed as described above, using a modified version of the FlexCRF toolkit (Phan, 2005). The AppenLCA development set is used as training materials for the diacritiser whereas the AppenLCA test set is used for evaluation. For both sets, the corpus collectors have provided demographic information about the speakers such as gender and dialect. Figure 6.8a illustrates the distributions of these demographic characteristics for each set. This allows extracting the speaker-level properties described above for each utterance to be used as training samples in addition to the lexical-level properties described in Table 6.6. A feature is counted if there is at least one occurrence in the training samples, i.e. the cut-off threshold for features was set to 1. Feature functions weights, λ_i and μ_i, are estimated using the limited memory quasi-Newton optimisation algorithm (L-BFGS).

Following the evaluation in the literature, performance is obtained on the AppenLCA

Table 6.6: Lexical-level properties for each Arabic grapheme. *Place-voicing-manner* attributes are assigned to a grapheme based on its most frequent associated phoneme. *Lam* indicates whether the lam in the definitive "Al" is assimilated (solar) or not (lunar) if it is followed by a grapheme. *Group* indicates whether a grapheme shares its pronunciation with another or whether they are substituted by mistake. The final grapheme "sp" or "-" indicates word boundaries with no acoustic value.

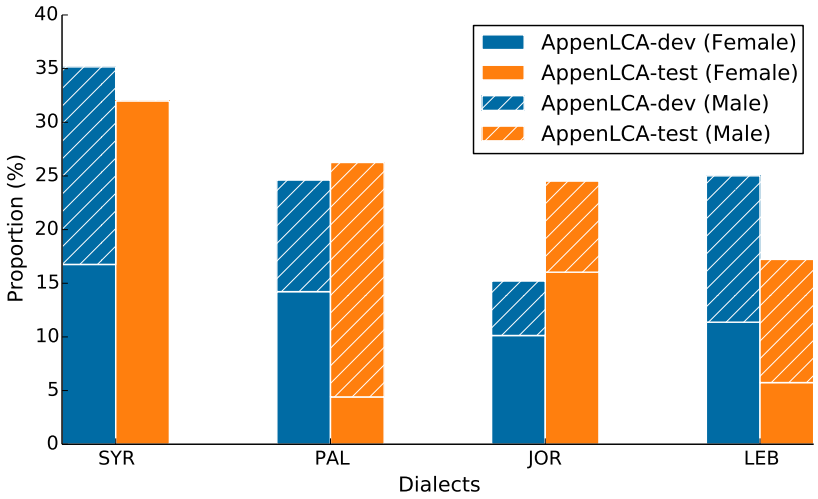
Grapheme	Manner	Place	Voicing	Lam	Group
>	stop	glottal	voiced	lunar	alif
<	stop	glottal	voiced	lunar	alif
|	vowel	lowback	voiced	lunar	alif
e	stop	glottal	voiced	lunar	none
&	stop	glottal	voiced	lunar	waw
}	stop	glottal	voiced	lunar	yaa
A	vowel	lowback	voiced	lunar	alif
Y	vowel	lowback	voiced	lunar	yaa
b	stop	bilabial	voiced	lunar	none
t	stop	alveolar	unvoiced	solar	none
v	fricative	dental	unvoiced	solar	none
j	fricative	postalveolar	voiced	lunar	none
H	fricative	pharyngeal	unvoiced	lunar	none
x	fricative	velar	unvoiced	lunar	none
d	stop	alveolar	voiced	solar	none
*	fricative	dental	voiced	solar	none
r	liquid	alveolar	voiced	solar	none
z	fricative	alveolar	voiced	solar	none
s	fricative	alveolar	unvoiced	solar	none
$	fricative	postalveolar	unvoiced	solar	none
S	fricative	alveolar	unvoiced	solar	none
D	stop	alveolar	voiced	solar	none
T	stop	alveolar	unvoiced	solar	none
Z	fricative	dental	voiced	solar	none
E	fricative	pharyngeal	voiced	lunar	none
g	stop	velar	voiced	lunar	none
f	fricative	labiodental	unvoiced	lunar	none
q	stop	uvular	unvoiced	lunar	none
k	stop	velar	unvoiced	lunar	none
l	liquid	alveolar	voiced	solar	none
m	nasal	bilabial	voiced	solar	none
n	nasal	alveolar	voiced	solar	none
h	fricative	glottal	unvoiced	lunar	taa
w	liquid	labialvelar	voiced	lunar	waw
y	liquid	palatal	voiced	lunar	yaa
p	fricative	glottal	unvoiced	lunar	taa
G	fricative	velar	voiced	lunar	none
sp / -	none	none	none	none	boundary

Table 6.7: Values for each property extracted from the training text for CRF diacritiser training. *lam* describes whether the /*l*/ in article "Al" is pronounced, i.e. lunar lam, when followed by the grapheme or not, i.e. solar lam. *group* describes extra properties by certain graphemes such as different shapes of alif.

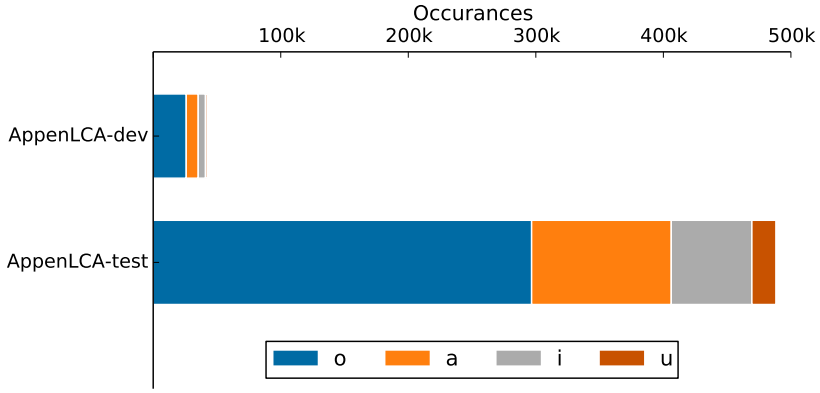
property		values
grapheme		39 grapheme + sp (word delim.)
phonetic classes	place	alveolar, bilabial, dental, glottal, labial velar, labiodental, lowback, palatal, pharyngeal, post-alveolar, uvular, velar, none
	manner	fricative, liquid, nasal, stop, vowel , none
	voicing	voiced, unvoiced, none
	lam	solar, lunar, none
	group	alif, yaa, taa, boundary, none
speaker dialect		syrian, lebanese, jordanian, palestinian
speaker gender		male, female

Table 6.8: An example of properties corresponding to the sentence "masaA Alxayr" used in diacritisation.

grapheme (c)	phonetic classes (p)					speaker dialect (d)	speaker gender (g)	diacritic
	place	manner	voicing	lam	group			
m	bilabial	nasal	voiced	lunar	none	syrian	female	a
s	alveolar	fricative	unvoiced	solar	none	syrian	female	a
A	lowback	vowel	voiced	lunar	alif	syrian	female	o
sp	none	none	none	none	boundary	syrian	female	o
a	lowback	vowel	voiced	lunar	alif	syrian	female	o
l	alveolar	liquid	voiced	solar	none	syrian	female	o
x	velar	fricative	unvoiced	lunar	none	syrian	female	a
y	palatal	liquid	voiced	lunar	yaa	syrian	female	o
r	alveolar	liquid	voiced	solar	none	syrian	female	o



(a) Distribution of speakers' sub-dialects and genders within the AppenLCA development set (containing 112.7k words) and test set (containing 9.8k words). LCA sub-dialects are Syrian (*SYR*), Palestinian (*PAL*), Jordanian (*JOR*) and Lebanese (*LEB*).



(b) Distribution of diacritics within the AppenLCA development and test sets.

Figure 6.8: Statistical details about the AppenLCA development and test sets.

test set using two metrics: the diacritics error rate (DER) and the diacritisation word error rate (DWER). Table 6.9 shows the results for these experiments. For each combination, the number of features, $\delta(\mathbf{c}, t)$, is shown. This indicates the number of parameters (weights) to be estimated. Among each property, the set of PVM features has the highest number of features because it is a multi-stream attribute with high number of possible assignments[9], whereas the other attributes are single-stream. When the number of different features increases, the number of weights to be estimated increases, thus, leading to insufficient training samples for the estimation. This explains the degradation in the prediction performance when using all the attributes (**cpdg**) in the prediction process, yielding the highest DER of 11.3%. Both chosen extralinguistic attributes improved the local prediction (DER) but behave totally differently on the global level (DWER). On one hand, the speaker's gender (**g**) seems the least informative attribute to be combined with the grapheme sequence (**c**) among all the three attributes. This is to be expected since it only has two values (female and male) and it is a static attribute that does not change during the whole segment, i.e. it only divides the training sets into two subsets, which does not contribute to long-span prediction to get the highest DWER of 76.9%. On the other hand, speaker's dialect (**d**) achieved the lowest DER and DWER of 8.8% and 23.3% respectively. This means that using speakers' dialects along with the grapheme identity factorises the overall graphemes and diacritics distribution into more homogeneous smaller distributions which is translated into more accurate prediction. The DER per diacritic correlates with the proportion of the diacritic within the training set (shown in Figure 6.8b) by which the lowest DER obtained is for the vowel-less mark while the highest DER is for the nunation diacritic ("F") which constitutes 0.003% of the overall diacritics in the training set.

The amount of training data needed for training of CRFs is of crucial importance as fully diacritised transcriptions are expensive to generate. The above experiments give a clear indication that the dialect itself is a strong indicator and thus ideally transcripts for each dialect are required. Thus a second experiment was conducted to look at the impact of the amount of training data on diacritisation performance. Figure 6.9 illustrates the DWER as a function of training set size when combining grapheme (**c**), dialect (**d**) and PVM and other phonological attributes (**p**) in the diacritisation. The baseline performance for a training set of 110k words (this corresponds to approximately 8 hours of speech), that generates 23.4k of logical functions, is a DWER of 28.5% and a DER of 11.2%. The gain in performance with approximately 4 times the amount of data is significant, but the slow non-uniform decrease suggests that either modest or very large quantities of data are

[9]PVM attribute has five input streams (place, manner, voicing, lam and group) the total number of permutations is 3510 ($= 5 \times 3 \times 3 \times 6 \times 13$).

Table 6.9: Diacritisation performance and features counts, $\delta(\mathbf{c}, t)$, when using different combination of properities in training CRF-based diacritisers. Properities included are the grapheme sequence (**c**), PVM and other phonolgical attributes (**p**), speaker's dialect (**d**) and gender (**g**).

Proprieties				Count of	DWER	DER					
c	**p**	**d**	**g**	$\delta(\mathbf{c}, t)$		Average	o	a	i	u	F
x				19811	27.2	10.4	3.6	14.1	20.1	28.8	56.3
x			x	20065	76.9	9.0	3.1	13.7	21.4	31.8	59.4
x		x		20287	23.3	8.8	3.2	12.1	17.5	19.4	28.1
x	x			28876	27.4	10.5	4.1	13.5	20.7	27.5	53.1
x	x	x		28986	28.2	10.8	4.2	14.4	20.2	29.0	65.6
x	x	x	x	29240	29.1	11.3	3.9	12.7	20.0	26.2	43.8

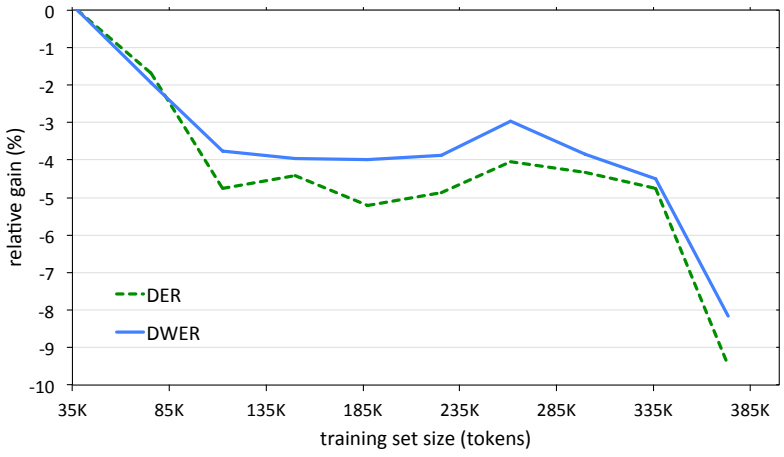


Figure 6.9: DWER and DER gain over training set size.

Table 6.10: ASR recognition performance and number of generated pronunciations per word using undiacritised graphemes (*graph*), manually diacritised graphemes (*mandiac*), generic vowels (*gv*) and CRF-based autodiacritised graphemes (*crfdiac*).

AM	Pronunciations/word	WER	Substitutions	Deletions	Insertions
graph	1	71.4	48.5	20.2	2.7
mandiac	1.3	70.8	48.9	19.8	2.1
gv	1	74.1	51.1	20.1	2.9
crfdiac	1.1	70.6	48.0	20.6	1.9

Table 6.11: Percentage of data available for each sub-dialect from AppenLCA training set and the corresponding DER, DWER and WER when automatically and manually diacritising is employed. LCA sub-dialects are Syrian (*SYR*), Palestinian (*PAL*), Jordanian (*JOR*) and Lebanese (*LEB*).

Sub-dialect	% of training	*crfdiac*			*mandiac*
		DER	DWER	WER	WER
SYR	35.1	10.3	27.6	73.6	73.1
PAL	24.6	10.2	27.1	66.2	66.0
JOR	15.2	10.8	29.3	65.2	64.5
LEB	25.0	11.7	29.6	79.2	82.2
Average		10.7	29.8	70.6	70.8

needed for training.

These results are similar to those yielded by the 4-gram character-based of Vergyri et al. (2005) on an LCA data and achieved a DWER of 30% and a DER of 7% on a 6K-word test set. The authors chose their training words, consisting of 40K words, carefully to reach the best vocabulary coverage. Apart from the system configured to use only the grapheme sequence (**c**) and the speaker's gender (**g**), the proposed system achieved a better performance in terms of DWER using a training set with half the size, 20.5K words, of the one employed in Vergyri et al. (2005)'s work on a test set that is larger by 30%, 8K words. Moreover, by increasing the size of the training data, further improvement can be achieved.

6.6 CTS ASR Experiments

The impact of modelling diacritics explicitly after restoration using each of the methods described in Section 6.2 to 6.5 is measured in this section. The task used for this evaluation is LCA conversational telephone speech recognition. Appendix B describes the system used for the evaluation. The AppenLCA training and test sets were used which are both described in Appendix B.

As a baseline, an undiacriticised recognition dictionary was generated by splitting the

undiacritised word into its graphemes, and then each grapheme is replaced by its corresponding model name. This dictionary was used for training the undiacritised graphemic acoustic model set (*graph*). Using the manual diacritisation provided by the corpus collectors, a dictionary is generated where only seen diacritisation variants are provided to be used for training manually diacritised graphemic acoustic model set (*mandiac*). Using each of the investigated diacritisation methods (see Section 6.2 to 6.5), two additional acoustic model sets are trained: generic vowel (*gv*), and CRF-based diacritisation (*crfdiac*).

Table 6.10 compares the WER results by each acoustic model set. A modest gain, 0.8% WER relative, from using manual diacritics can be observed over the grapheme system baseline. Based on matched pair sentence segment word error (MAPSSWE) test, this improvement proved to be statistically significant with $p < 0.001$, and it agrees with the previous studies which compared between diacritised and undiacritised graphemic acoustic units. This gain can be even improved by using the CRF-based diacritisation system, reaching 1.1% WER relative compared to the system using *graph*. This can be due to the fewer number of pronunciations generated by the automatic CRF-based diacritisation system where the average is shown in the second column of Table 6.10. This observation is consistent with Hain (2002)'s findings in reducing the number of pronunciation variants in a dictionary improves ASR performance.

Although no pronunciation variants were used, employing a generic vowel does not contribute to the overall recognition performance due to the model's limitations discussed previously in Section 6.2.

crfdiac is trained using automatically diacritised transcriptions from the AppenLCA training set using a CRF-based diacritiser which was trained using the AppenLCA development set. By comparing the automatically diacritised transcriptions of the AppenLCA training set against its manually diacritised transcriptions, 29.8% of DWER with 10.7% DER is found. Table 6.11 gives a breakdown of DWER and DER among dialects and the corresponding WER. The source of improvement in the *crfdiac* over *mandiac* lies in the Lebanese subset of the data (LEB). Nevertheless, diacritisation performance is the poorest at this subset, ASR outperformed *mandiac* by 3.6% WER relative. A possible explanation is that the transcribers of this subsets diacritised some part of it incorrectly but these errors did not occur in the automatically diacritised data, hence did not match the incorrectly labelled references. As a consequence, higher diacritisation errors were observed. However, the automatically diacritised data matched the acoustic better than incorrectly-labelled transcription, and consequently, ASR performance improved. Apart from this subset, this difference does not show any statistical significance based on MAPSSWE. This is a strong indication of the consistency of the proposed CRF-based diacritisation system over manual diacritisation.

6.7 Summary and conclusion

Arabic transcriptions lack short vowels and gemination information while a formal Arabic dictionary does not provide pronunciations but only diacritisation variants and a variant is chosen according to the given context. Since hidden diacritics holds one third of the acoustic information, it is crucial for acoustic modelling purposes to retrieve those diacritics. After discussing previously introduced diacritisation methods along with their limitations, three diacritics restoration methods have been introduced for automatically diacritisation of CA using transcriptions only and evaluated in the context of diacritisation performance and speech recognition performance.

Inserting generic vowel with a skip after every grapheme weakens the context and triphones become less informative. As a consequence, the recognition performance degraded in a CTS ASR task.

Two original CA automatic diacritisation systems were proposed. First diacritisation system was implemented based on a grapheme-to-phoneme (G2P) framework. It requires a small amount of diacritised seeding data (5000 diacritised CA words) to achieve a consistent and highly accurate performance. G2P framework has not been employed for diacritisation purposes of either MSA and CA in previous studies. For the second diacritisation system, CRF models were employed to incorporate long-span features in the prediction process.

It was found that contextual and extralinguistic information can improve diacritisation by a significant margin; However, in practice data sparsity remains and issue and limits the number of features that can be included. In accordance it was shown that an increase in the amount of training data for diacritisation could lead to the better diacritisation performance. Speaker-dependent information has not been used in previous studies for vowelisation or diacritisation of both MSA and CA.

ASR results suggest that training of acoustic models on automatically diacritised improves the recognition performance, reaching 1.1% WER relative which shows to be statistically significant with $p < 0.001$ using MAPSSWE. Although improvements were observed, the issue of high confusability due to large numbers of pronunciations per word remains. Generally, there was no indication of any statistical significance between acoustic models trained on automatic and manual diacritised transcriptions. However, a case was shown in which automatic diacritisation recover some of the manual diacritisation errors which improved the recognition performance by 3.2% WER relative.

Chapter 7

Redefining the Arabic Acoustic Inventory

Contents

In an ASR system, pronunciation is modelled by mapping words into sequences of acoustic units by the means of a dictionary. Recognition performance for a particular language depends largely on the choice of acoustic units and the accuracy of the dictionary.

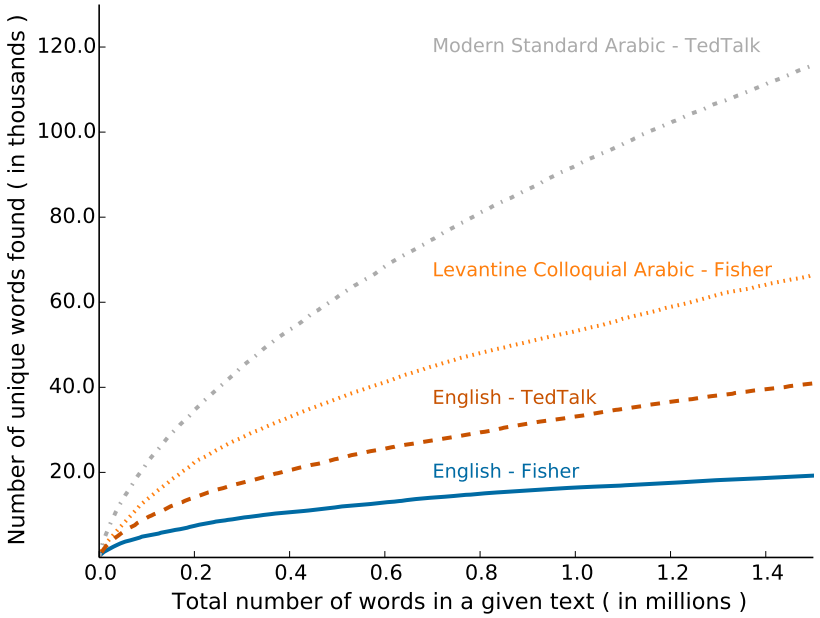


Figure 7.1: Vocabulary growth in different variants of Arabic in comparsion to English.

For instance, Killer et al. (2003) found that using phoneme-based units outperformed grapheme-based units for English and Spanish by 33% and 8% relative in terms of WER, respectively. Each unit in the set of acoustic units in an ASR describes the acoustically minimum distinguishable unit required for classification. These units are used to describe the pronunciation of a word in a language and are interchangeably referred to as subword units, acoustic inventory, acoustic model set or acoustic unit set or simply acoustic units and acoustic models in the literature and through the course of this chapter.

In English language ASR systems, dictionaries are typically manually designed and created by experts, with phones taken as acoustic units. However, such phonetic dictionaries are not available for CA for several reasons, and therefore many studies on Arabic speech recognition have used undiacritised grapheme-based units, such as that of Afify et al. (2006), Choueiter et al. (2006) and Billa et al. (2002). First, Arabic is a morphologically rich language where a word is generated by applying a combination of morphological processes where new words can be easily created either by applying a template to a root (derivation), or by concatenating articles and prepositions to a word without changing the original word (agglutination), or by concatenating pronouns and applying changes to the original word (inflection). As a result, the number of unique words in a given amount of text (vocabulary size) for Arabic generally becomes significantly greater than it in other languages which lack such rich morphology, such as English. For all languages, as the amount of the given text increases, the vocabulary size increases with a certain rate (known as vocabulary growth rate). This rate is considerably larger for languages with

rich morphology, such as Arabic, than for other languages with a simpler morphological system, such as English. Figure 7.1 shows the average growth of the vocabulary size for two Arabic variants in comparison to English. The vocabulary growth rate has a strong positive relationship with the amount of text in MSA. The rate is considerably smaller for LCA, which is expected as LCA is an exclusively conversational variant, unlike MSA, which is a high variant as discussed previously in Chapter 2. Second, the lack of diacritics in the Arabic text generally makes predicting the pronunciation for a given word ambiguous and these diacritics can only be disambiguated through the context. Moreover, CA introduces additional phonemes without adding new graphemes. Because CA has no writing conventions, writers improvise the spelling by assigning these CA phonemes to existing graphemes inconsistently, hence, more pronunciation variants are included for a CA word. As an example of such variety of pronunciations for one word is the undiacritised word "b>qwl" (I will say or I am saying), which has been transcribed in CA with 26 diacritised variants[1].

Extending the acoustic model set to include diacritics improves the recognition performance significantly, for both MSA and CA ASR (Kirchhoff et al., 2002a; Gales et al., 2007; Diehl et al., 2008; Soltau et al., 2009). In Chapter 6, several methods were investigated for restoring missing diacritics from a given CA undiacritised text, and these were successfully employed for improving CA ASR performance. Although the representation of a written utterance is brought closer to its actual acoustic realisation by retrieving the unwritten diacritics, there are still other discrepancies between the written form of Arabic and its spoken form, that have not been addressed previously, such as homographemes and silent graphemes (Section 2.4). While these issues exist in MSA generally, it becomes more prominent in CA due to the introduction of new phonemes without introducing new graphemes. Consequently, existing graphemes are associated with the new phonemes which resulted in an increasing number of multi-phoneme graphemes. This chapter fulfils Objective 4 of this thesis by investigating the derivation of the most suitable acoustic model set for CA ASR that overcomes these issues in the absence of a conventional pronunciation dictionary. It also seeks a mapping between the newly introduced inventory and the graphemic representation to be used for generating pronunciations for unseen vocabulary.

This chapter is organized as follows: First, the conventional types of acoustic unit inventories for Arabic are introduced in Section 7.1, along with the previous work in the existing literature on choosing acoustic units for ASR. The unresolved issues, homographemes and silent graphemes, in using diacritised graphemes for acoustic modelling are addressed in Section 7.2 where context-dependent solutions are introduced, and in Section 7.3 where the

[1] 26 diacritised variants were collected across three dialects in the Appen corpora (Gulf, Iraqi and Levantine CA).

acoustic model space is restructured. The impact of using the new inventory is evaluated in phone and speech recognition tasks, as described in Section 7.4. Finally, conclusions are drawn in Section 7.5.

7.1 Related research

Acoustic units are the smallest representation of sound in an ASR system. Phones are the most commonly used units in ASR. If a phone definition is not available for a given language due to lack of a pronunciation dictionary, graphemes are used. Graphemes are another representation derived from the textual reference. Arabic speech recognition has been extensively researched in the last decade. The chosen acoustic units for most of previous studies were graphemic-based due to the lack of phonetic dictionaries for Arabic.

Generally, diacritised graphemes have been considered to be equivalent to the phonetic transcription of a given utterance. Thus, these symbols were used as acoustic units. In most of the previous studies, silent and ambiguous graphemes were treated as pronunciation modelling problems. For instance, the assimilation of the definitive lam, if it is followed by one of the solar letters, was addressed by generating alternative pronunciations. Another example is in handling the ambiguous pronunciation of the glottal stop when it appears at the beginning of a word. Glottal stop, known as hamza, is represented by four graphemes. The chosen grapheme is disambiguated by using knowledge-based tools such as BAMA (Buckwalter, 2002; 2004a) and MADA (Habash et al., 2007), or simply by normalising all its possible occurrences into one grapheme instead.

Taking a contrary approach, two studies have mapped the graphemic form to conventional phonemes using phonological rules. Ali et al. (2008) adopted a purely phonological approach by using phonetic rules to extend the acoustic inventory and to use these rules to generate a pronunciation dictionary. Their new acoustic inventory extended to 46 phones with 12 vowels (three short, three long, three emphatic and three pharyngeal), two diphthongs and 31 consonants. A similar approach was employed by Biadsy et al. (2009a) but they limited their inventory extension to the diphthongs only; however, they used morphological information in order to disambiguate silent and ambiguous graphemes. In contrast, Ali et al. did not address the latter grapheme category in their work.

7.1.1 Motivation

It was empirically shown in Chapter 6, which is also aligned with other previous studies such as Kirchhoff et al. (2002a); Gales et al. (2007); Diehl et al. (2008); Soltau et al. (2009), that increasing the detail of the acoustic inventory to include diacritics improves recognition performance significantly for both MSA and CA ASR. Figure 7.2 lists three

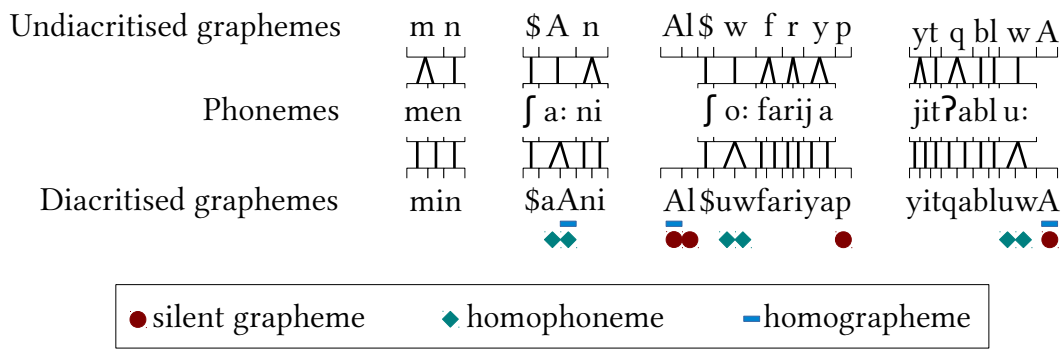


Figure 7.2: Graphemic representation (undiacritised and diacritised) and their mappings to phonemes of an LCA sentence "mn $An Al$wfryp ytqblwA" (for the drivers to be accepted). If no mappings between grapheme and phoneme, it is a silent grapheme. Issues in diacritised graphemes are shown by markers under graphemes with these issues.

representations for an example of an LCA sentence "mn $An Al$wfryp ytqblwA" (for the drivers to be accepted). The top and bottom representations are graphemic with the middle sequence showing the phonetic transcription written in IPA. A mapping between the units in the graphemic representations and their corresponding phonemic units are shown. For instance, "m" in undiacritised grapheme sequence is mapped to two phonemes /me/ while "m" in the diacritised grapheme sequence is only mapped to one phoneme /m/. Mostly, each grapheme can be mapped to more than one phoneme in undiacritised form but not in the diacritised form. More one-to-one mappings between the chosen acoustic units and phonemes can guarantee a better separation, hence better classification and recognition. That explains the reason for the recognition improvement when using diacritised graphemes for acoustic modelling over undiacritised graphemes, for example, the number of one-to-one mappings between phonemes and graphemes in Figure 7.2 67%[2] of the diacritised graphemes are having a one-to-one mapping to a phoneme but only 45% of the undiacritised graphemes are with one-to-one mappings. For a given language, an optimal acoustic unit set should have a one-to-one mapping between the units and phonemes in that language for every unit in the set. Adding diacritics narrows the gap between the written form and its phonetic realisation. However, there are still some acoustic phenomena which are not represented by the fully diacritised written form, namely silent graphemes, homophonemes and homographemes. These issues are presented by markers underneath diacritised graphemes in Figure 7.2, for instance, the phoneme /o:/ is mapped to a sequence of more than one diacritised graphemes "uw" and "A" grapheme at the last word is silent. Moreover, considering the mapping from letter to sound in Arabic and the

[2]There are 20 out of 30 diacritised graphemes with one-to-one mappings to a phoneme but only 9 out of 20 in undiacritised grapheme sequence.

context-dependent conditions (discussed in Section 2.4), even diacritised graphemes cannot be considered an optimal acoustic inventory for ASR tasks due to the existence of the following issues:

A. Homophonemes: several graphemes which are sharing the same phoneme, such as "Y" and "A" both pronounced as /a:/, while the hamza (glottal stop /ʔ/) can be represented by six different graphemes (<,>,|,},&,',A).

B. Silent grapheme: a grapheme that can be mapped to either a phoneme or silence.

C. Context-dependent homographeme: a grapheme represents more than one phoneme and can be disambiguated by the context, such as "w" as /u:/ or /w/, and "y" as /i:/ or /j/.

D. Context-independent homographeme: a grapheme represents more than one phoneme and cannot be disambiguated by the context, such as "q" which can be pronounced as /q/, /k/, /ʔ/ and /g/.

One of the issues of diacritised graphemic-based acoustic models mentioned above (issue A) can be easily resolved by nominating only one grapheme to be associated with a certain phoneme. However, the other issues are more complicated and need to be addressed either by introducing new models to map the most frequent graphemes onto the newly introduced models depending on the context, or by introducing a rule-based approach to decide the chosen phoneme for the homographemes.

7.2 Development of the new Arabic acoustic inventory

When examining diacritised graphemes, the closest representation available to the phonetic transcription in written form, one can observe subtle differences between written and spoken forms. Some graphemes are mapped to the same phoneme (*homophoneme*) or vice versa where one grapheme can be pronounced as several phonemes (*homographeme*) and some can be even silent. These differences are problematic for both acoustic and pronunciation modelling.

On one hand, one of the objectives of acoustic modelling is to design the acoustic units within a language where they are separable for classification. Then, these units are used to describe word pronunciations in the system. Homophonemes in the diacritised graphemes model the same phoneme using several graphemic units instead of only one unit. On the other hand, in terms of predicting pronunciation from the graphemeic representation, homophonemes are not problematic if there exists only *one* mapping from a grapheme to the shared phoneme.

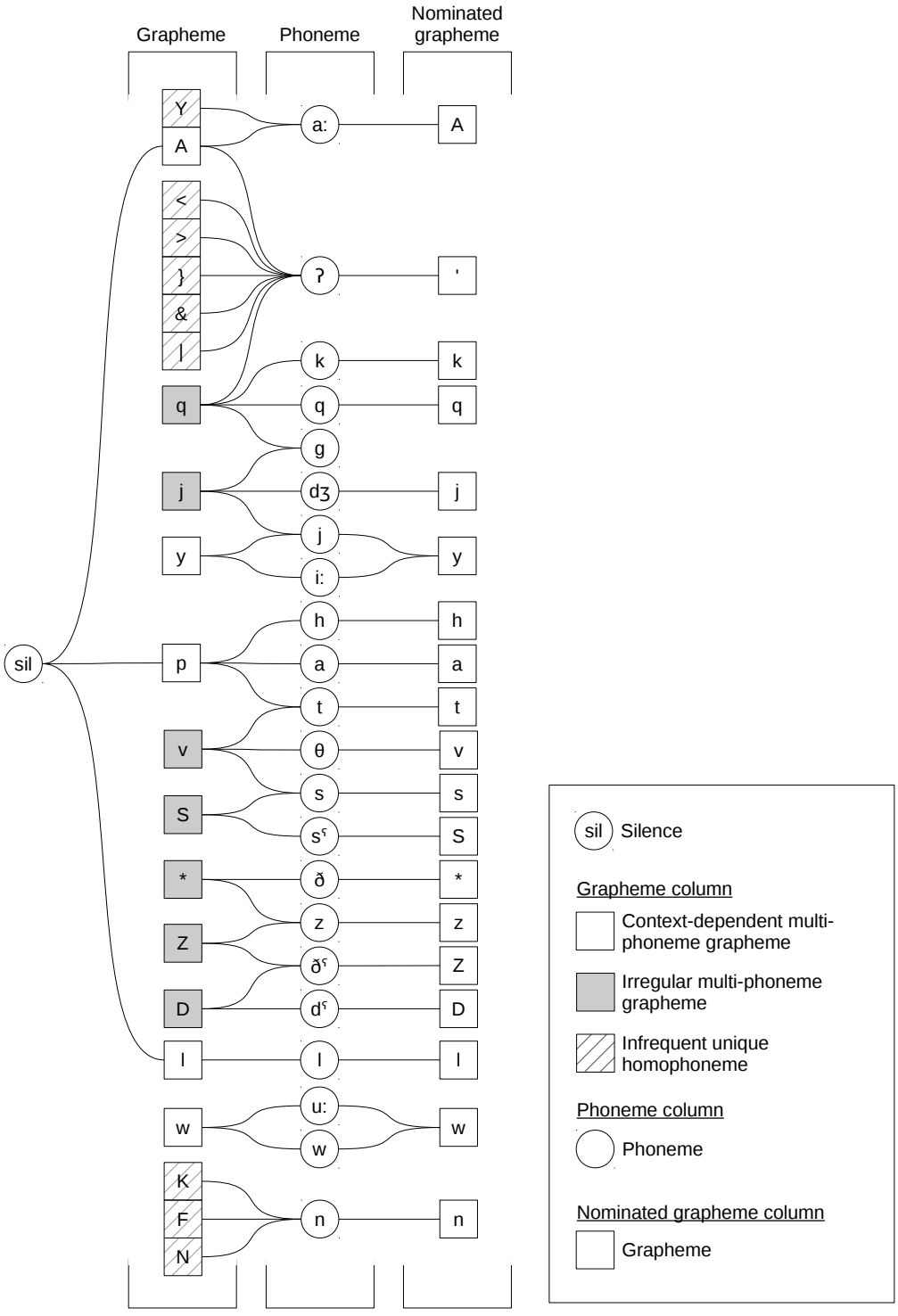


Figure 7.3: Mappings between graphemes and phonemes in CA that cause issues in acoustic modelling. Phonemes are written in IPA symbols. Nominated grapheme is the grapheme usually chosen in the language to represent the associated phoneme.

Unlike homophonemes, homographemes are ambiguous because they might be assigned to several phonemes which must be disambiguated in order to reach to the closest representation of the actual acoustic value in the speech signal. Fortunately, some of these ambiguous graphemes can be disambiguated based on the context but some are not. Figure 7.3 visualises the mappings between these problematic graphemes (homophonemes and homographemes) and their assigned phonemes. Three columns of nodes are shown, with phonemes in the middle linked with their nominated graphemes in the right column. The problematic graphemes are listed in the left column. A nominated grapheme is a grapheme used to represent a phoneme when it is out of context. For example, the phoneme /z/ on its own in English is associated with the grapheme "z" even when the phoneme can be associated with other graphemes in certain context such as "s" and "x".

Figure 7.3 shows those graphemes which are mapped to the same phoneme (e.g. "Y" and "A" both pronounced as /a:/) and vice versa (e.g. "q" as /ʔ/, /k/, /q/ and /g/) and those which can even be silent (e.g. "l"). Moreover, there exist phonemes which are shared by several homographeme sets. For example, /ʔ/ is mapped to three different sets {A}, {<, >, }, ', &, |} and {q}. Some of the graphemes associated with multiple phonemes are context-dependent. This means the assigned phoneme can be decided based on the surrounding context and therefore is predictable, for instance "l" is assimilated if it is followed by a solar letter, and for other graphemes it cannot be predicted.

There are three categories of graphemes:

- *Unique homophoneme*: a grapheme that is assigned to only one phoneme. This phoneme is mapped to one or more other graphemes.
- *Non-silent multi-phoneme grapheme*: a grapheme that is assigned to several phonemes but never becomes silent. Some of these mappings are *context-dependent* by which the phoneme can be chosen based on the context (text or acoustic) while other mappings are *context-independent*.
- *Silent multi-phoneme grapheme*: a grapheme which is assigned to one or more phonemes and can become silent.

Given the issues discussed above, using diacritised graphemic representation as acoustic models resulted in one third[3] of acoustic units with ambiguous phoneme assignments. Figure 7.4 shows the average frequency distribution of each unit in the diacritised graphemes in four Arabic dialects. It also illustrates that the 13 ambiguous graphemes represent 32% of the data, while more than one fifth of the units represents less than 3%. Moreover, the frequency distribution of the grapheme set is skewed towards one or two vowels, for

[3] 13 is one third of the 39 total units in the diacritised grapheme set.

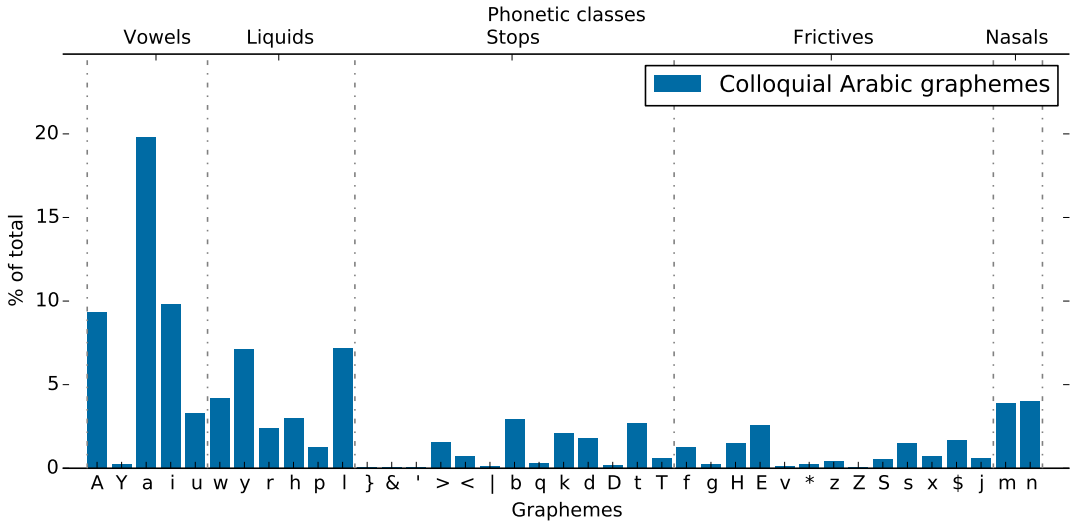


Figure 7.4: Average frequency of diacritised graphemes in the transcriptions of a 36-hour set from four CA: MSA, LCA, ICA and GCA. Phonetic classes are labelled on the top x-axis while models belong to each class is shown on the bottom x-axis between enclosed between the dashed vertical lines.

instance, an average of nearly 20% of the graphemes seen across different dialects is "a". Theoretically, resolving these issues increases the separability between the units, consequently this should lead to a better recognition performance in general as it has been previously discussed when using diacritised graphemes over undiacritised graphemes in acoustic modelling in Chapter 6.

Each of these issues will be addressed in this section, except for the context-independent multi-phoneme graphemes, which will be discussed in Section 7.3.

7.2.1 Infrequent unique homophonemes

A homophoneme is a grapheme that shares a phoneme with another grapheme, i.e. there are multiple mappings associating a given phoneme with more than one grapheme. For example, "Y" and "A" are homophonemes (shown in Figure 7.3) because both are associated with the same phoneme /a:/. If one of these homophonemes has no other mappings to any other phoneme, then this grapheme will be referred to as a unique homophoneme because it has a unique mapping to a phoneme. For example, for the homophonemes "Y" and "A", "Y" is only mapped to the phoneme /a:/ but no other phonemes, hence, it is a unique homophoneme while "A" is not a unique not a unique homophoneme because it is also has two additional mappings to /ʔ/ and silence. Some of these homophonemes are less frequent than others which share the same phoneme. For instance, the glottal stop /ʔ/ is represented by six unique homophonemes, which are {"A", "<", ">", "}", "&", "‖"},

with some of these homophonemes are less frequent that the others (average frequency distribution is illustrated in Figure 7.4). In such cases, mapping all these homophonemes to one model improves the probabilistic distribution representing that particular phoneme.

Taking the mappings in Figure 7.3, the frequency distribution of each grapheme in Figure 7.4 and the phonological knowledge driven from Ali et al. (2008) and Biadsy et al. (2009a), the following rules were devised to map each unique homophoneme set onto one model. Each rule is written in the format: $L \rightarrow R$, where L is left side graphemic sequence which will be replaced by the right side phonemic sequence, R. The symbol + indicates whether L and R are composed of more than one unit, for instance $a+b \rightarrow c$ corresponds to replacing the sequence of graphemes a and b with the phoneme c. Finally, possible alternatives are grouped between square brackets, for example, $[a, b] \rightarrow c$ represents replacing any occurrences of a or b with the phoneme c. Each of the following rules maps several graphemes to one dedicated unit that represents the associated phoneme to these graphemes.

Hamza rule

$$[>, <, \}, \&] \rightarrow e,$$

Madda rule

$$| \rightarrow e + aa,$$

Nunation rule

$$F \rightarrow a + n,$$
$$K \rightarrow i + n,$$
$$N \rightarrow u + n,$$

where e represents the glottal stop /ʔ/, aa is the long vowel /a:/, n is the nasal phoneme /n/ and a, u, i are the short vowels.

7.2.2 Context-dependent non-silent multi-phoneme graphemes

If a grapheme is assigned to several alternative phonemes and its acoustic realisation can be disambiguated by the context but never becomes silent, it is called a context-dependent non-silent multi-phoneme grapheme. There are two such graphemes: “w”, which represents the glide /aw/ or long vowel /u:/ and “y” which represents the glide /ay/ or long vowel /i:/, where their pronunciation can be chosen based on the surrounding diacritics[4].

Arabic vowels are presented using either graphemes or diacritics, indicating long or short duration respectively. Having a sequence of a grapheme and a diacritic where both

[4]This derivation requires a diacritised grapheme sequence as an input. If no diacritics are available, automatic diacritisation is employed, as discussed in Chapter 6.

represents the same vowel is an indication that the pronounced sequence is a long vowel. For example, if the diacritic "u" (/u/) followed or preceded "w" (/u:/) as in "$uw" (what), the sequence is merged into a long vowel to be pronounced as (/u:/). However, if the vowel represented by the diacritic differs from the vowel represented by the grapheme, the sequence is pronounced as a diphthong or a glide. For instance, if the grapheme "w" is preceded "a" (/a/), then it is pronounced as the glide /aw/. Otherwise, if there are no diacritics around, this grapheme cannot be disambiguated. These conditions can be translated into the following rules:

w and y as diphthongs

$[a, i] + w \rightarrow aw + w$,
$[a, u] + y \rightarrow ay + y$,
$w + [a, i] \rightarrow aw + [a, i]$,
$y + [a, u] \rightarrow ay + [a, u]$,

w and y as long vowels

$u + w \rightarrow uw$,
$i + y \rightarrow iy$,

If there are no diacritics, w and y are ambiguous

$w \rightarrow w_a$,
$y \rightarrow y_a$,

where aw and ay are the diphthongs /aw/ and /ay/ respectively while uw and iy are the long vowels /u:/ and /i:/ and w_a and y_a mark the cases where the graphemes cannot be acoustically disambiguated from the written form.

In order to disambiguate w_a and y_a, acoustic information can be employed. This can be achieved by force-aligning these graphemes to a model representing either a long vowel or a glide. Such process requires an initial training for the long vowel models (uw and iy) and the glide models (aw and ay). These models are then used as pronunciation variants for any given word including one of these ambiguous units.

7.2.3 Context-dependent silent multi-phoneme graphemes

As aforementioned, some Arabic graphemes can be silent in certain conditions. Falling in this category are the definitive "Al" graphemes, "A" and "l", where both can be silent and assimilated in certain conditions. In addition, if the grapheme "A" is followed by a diacritic, it indicates that it is a hamza wasl (i.e. skippable or silent hamza); otherwise it is pronounced as a long vowel /a:/. These conditions can be formulated into the following rules:

Solar and lunar letters

$$A+l+[t,v,d,*,r,z,s,\$,S,D,T,Z,l,n] \rightarrow A+l_0+[t,v,d,*,r,z,s,\$,S,D,T,Z,l,n],$$

A in Al

$$A+l \rightarrow a_0+l,$$

Hamza wasl

$$A+[a,i,u,l] \rightarrow a_0+[a,i,u,l]$$

Waw aljamaa

$$w+A \rightarrow w+a_0$$

Alif as a vowel

$$A+![a,u,i] \rightarrow aa+![a,u,i]$$

l_0 and a_0 are skippable variants of the models l and a while aa is the long vowel /a:/ and the symbol $![x]$ means any grapheme other than x. A skippable model is a model by which there is an option not be used in the pronunciation, i.e. to be skipped. For example, the word "Aqr>" (read) can be pronounced as /ʔqraʔ/ and /qraʔ/ where the first phone can be skipped, hence, the corresponding pronunciation after applying these rules would be "a_0 q r a e".

Another grapheme which can be included in this category is the grapheme "p", known as taa marbouta, which can be pronounced as /h/ when it is followed by a pause, as /t/ when it is either connected to the next word or followed by a nunation or it can be silent. Thus the rules are:

Taa marbouta and nunation

$$p+[K,N,F] \rightarrow t+[K,N,F],$$

Otherwise

$$p \rightarrow p_{ao},$$

where p_{a0} is an ambiguous model, representing the sounds as /t/ or /h/, or no sound.

The models a_0, l_0 and p_{a0} can be disambiguated using acoustic contextual information via forced alignment. However, in order to initialise these models properly, a one-state path is added in their topology (as illustrated in Figure 7.5) to represent an almost absence of the model in the acoustic signal. The reason for using a one-state path instead of a skip is completely for practical reasons. Because it is possible to have a sequence of two or more skippable graphemes, this will create an unnecessary expansion during the alignment in Viterbi decoding. Using this topology reduces the ambiguity from generating pronunciation variants for all the possible permutations, especially where a word can contain more than

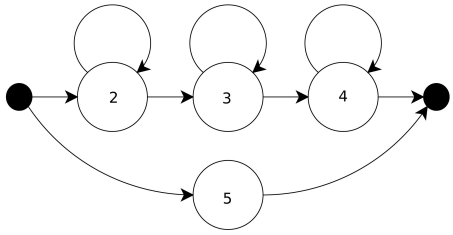


Figure 7.5: HMM topology for skippable acoustic units.

one potentially silent model. By using these initial models in the forced alignment, the final phonetic sequence can be obtained.

7.2.4 Acoustic inventory derivation procedure summary

The rules in Sections 7.2.1 to 7.2.3 were formulated to either map several graphemes to one phoneme or disambiguate a chosen phoneme based on the context which can be derived from the grapheme sequence or from acoustic values via forced alignment. There is an interdependency between these rules, for instance, the nunation rule ($F \rightarrow a{+}n$, $K \rightarrow i{+}n$ and $N \rightarrow u+n$) cannot be applied before using the nunation symbols to disambiguate the "p" in ($p + [K, N, F] \rightarrow t + [K, N, F]$), so these heuristics must be applied in a certain sequence.

The overall derivation procedure of the new acoustic inventory is extracted from transforming transcriptions in diacritised grapheme by applying a sequence of phonological rules. This procedure can be summarised in the following steps:

1. Diacritise the given transcription if it is in undiacritised form using one of the methods described in Chapter 6.

2. Generate initial pronunciations by applying the heuristic rules discussed in the previous sections in the following sequence where skippable and ambiguous units are employed.

   2.1. Hamza rule: $[>, <, \}, \&] \rightarrow e$

   2.2. Madda rule: $| \rightarrow e + aa$

   2.3. Al rule: $A + l \rightarrow a_0 + l$

   2.4. Solar and lunar letters: $l + \text{Solar} \rightarrow l_0 + \text{Solar}$

   2.5. Hamza wasl: $A + [a, i, u] \rightarrow a_0 + [a, i, u]$

   2.6. Waw aljamaa: $w + A \rightarrow w + a_0$

   2.7. Alif as long vowel: $A{+}![a, u, i, l] \rightarrow aa{+}![a, u, i, l]$

2.8. w and y rules:

2.8.1. As diphthongs by precedent: $[a, i] + w \rightarrow aw + w$, $[a, u] + y \rightarrow ay + y$

2.8.2. As diphthongs by subsequent: $w + [a, i] \rightarrow aw + [a, i]$, $y + [a, u] \rightarrow ay + [a, u]$

2.8.3. As long vowels: $u + w \rightarrow uw$, $i + y \rightarrow iy$

2.8.4. Otherwise: $w \rightarrow w_a$, $y \rightarrow y_a$

2.9. Taa marbouta rules:

2.9.1. With nunation: $p + [K, N, F] \rightarrow t + [K, N, F]$

2.9.2. Otherwise: $p \rightarrow p_{ao}$

2.10. Nunation rules: $F \rightarrow a + n$, $K \rightarrow i + n$, $N \rightarrow u + n$

2.11. Sequence of repeated units must be substituted by only one unit.

3. Train initial models using the initial pronunciations where only skippable units use the topology with a skip illustrated in Figure 7.5.

4. Generate a disambiguation dictionary where each word with an ambiguous model will have pronunciation variants as follows:

$$\begin{array}{lcl} a_0 & \in & \{a_0, \quad \text{aa}, \quad \text{e}\} \\ l_0 & \in & \{l_0, \quad \text{l}\} \\ p_{ao} & \in & \{p_{ao}, \quad \text{t}, \quad \text{h}, \quad \text{a}\} \\ w_a & \in & \{\text{w}, \quad \text{uw}\} \\ y_a & \in & \{\text{y}, \quad \text{iy}\} \end{array}$$

5. Use forced alignment with the initial models on the disambiguation dictionary.

6. Remove all skippable units from the pronunciation when the path including the skip state is chosen.

7. Replace all ambiguous units with the chosen units from the resulted alignment.

8. Final acoustic inventory is extracted from units used in the aligned transcription.

In addition to the new acoustic inventory, two other training resources can be extracted. These are a pronunciation dictionary which can be derived from the aligned transcription and aligned training transcriptions with the new acoustic inventory.

7.2.5 Results and discussion

Several experiments to assess the proposed acoustic inventory were conducted. The proposed acoustic inventory can be used with different CA dialects. At first, the derivation

Table 7.1: The percentage of words affected by applying the phonological rules to derive a new acoustic inventory in LCA training set in the vocabulary list (%vocab) and weighted by word frequency in the training data (%data). e is a glottal stop. X_0 are skippable models (disambiguated by skip topology) while X_a are ambiguous models (disambiguated by pronunciation variants). Solar $\in \{v, *, Z, t, T, d, D, z, s, S\}$, Nunation $\in \{K, N, F\}$ and $V_{\text{short}} \in \{a, u, i\}$.

	Rule	**description**	**%vocab.**	**%data**
1.	hamza rule	$[\{,\&,<,>] \rightarrow e$	12.8	13.8
2.	madda rule	$\mid \rightarrow e{+}aa$	0.8	0.7
3.	solar lam rule	$l{+}\text{Solar} \rightarrow l_0{+}\text{Solar}$	3.7	1.8
4.	Al rule	$A{+}l \rightarrow a_0{+}l$	17.0	12.6
5.	hamza wasl	$A{+}V_{\text{short}} \rightarrow a_0 + V_{\text{short}}$	7.7	5.9
6.	waw aljamaa	$w + A \rightarrow w + a_0$	3.9	4.1
7.	alif as long vowel	$A{+}![V_{\text{short}}, l] \rightarrow a_0 + ![V_{\text{short}}, l]$	37.1	13.5
8.	waw and yaa rules:			
	1. glides by precedent	$[a + i]{+}w \rightarrow aw{+}w$	6.2	4.4
		$[a + u]{+}y \rightarrow ay{+}y$	7.2	9.0
	2. glides by subsequent	$w{+}V_{\text{short}} \rightarrow aw{+}V_{\text{short}}$	14.3	8.9
		$y{+}V_{\text{short}} \rightarrow ey{+}V_{\text{short}}$	10.9	9.1
	3. long vowels	$u{+}w \rightarrow uw$	11.7	8.7
		$i{+}y \rightarrow iy$	26.4	22.5
	4. otherwise	$w \rightarrow w_a$	3.5	1.3
		$y \rightarrow y_a$	1.8	1.3
9.	taa marbouta rules:			
	1. with nunation	$p{+}\text{Nunation} \rightarrow t{+}\text{Nunation}$	0.0	0.0
	2. otherwise	$p \rightarrow p_{a0}$	12.3	6.8
10.	nunation rule	$K \rightarrow i{+}n$	0.0	0.0
		$N \rightarrow u{+}n$	0.0	0.0
		$F \rightarrow a{+}n$	0.3	0.5
11.	repeated units	replaced by one unit		

of this inventory was analysed through the chosen pronunciation in the forced alignment and the change in the final acoustic set frequency distribution. Then, the captured phonotactics from the sequence of the proposed acoustic units were compared across four Arabic dialects to assess the sensitivity of the proposed set to the dialect's phonotactics. Finally, acoustic confusability was assessed across the dialects using forced alignment and phone recognition tasks by analysing the impact of incorporating the context of the dialect from the phonotactic language model in these tasks. The phonotactic analysis and acoustic confusability experiments were replicated in English for five North American dialects to obtain control results, to further aid the analysis of the observed behaviour in the Arabic experiments.

Table 7.2: Skippable and ambiguous units and their proportions within the initial training set.

Models	a_0	l_0	p_{a0}	w_a	y_a
Pronunciations	e aa a_0	l l_0	t h a p_{a0}	w uw	y iy
% of data	2.0	0.5	1.2	1.1	1.6

Analysis of the derivation procedure

In order to assess the derived heuristics, a small-scale phone-level experiment was conducted using a subset of the AppenLCA training set. A dialect and gender balanced data set containing 10 hours of speech was constructed. As an evaluation set, 10% of the selected subset was randomly selected on a single-side level instead of two-side level for better speaker coverage, while ensuring speaker-set separation between training and test sets.

Rules were applied in sequence as discussed in Section 7.2.4. For each rule, the words for which a rule applied were counted and weighted by the word frequency in the training set to provide an indication of its significance. The higher the percentage, the more significant a rule can be considered to be. Table 7.1 illustrates the sequence of applying the derivation rules along with the percentage of the affected words in the selected training set. Long vowel derivation rules (rule number 7 and 8.3 in Table 7.1) were the most frequent rules to be applied, followed by rules that resolved different shapes of alif and hamza (rule number 1, 2, 4 and 5) while the nunations (rule number 10) are the most infrequent due to their rare usage in CA.

As a result, a new pronunciation dictionary was generated, with set of 43 units, where three of these units are skippable and two are ambiguous units. An initial set of models was trained based on these resulting pronunciation transcriptions where the skippable models used the topology illustrated in Figure 7.5, while the rest of the models used a standard left-to-right 3-state HMM topology. The main purpose of this set is to validate the resulting pronunciations in the forced alignment process (Step 5 in the derivation procedure) where skips are identified and ambiguous models are resolved. For this, an alignment dictionary was generated where for each word containing any ambiguous model, pronunciation variants were generated using the pronunciations allowed for that model (listed in Table 7.2). For example, the resulting pronunciation for the diacritised word “diktawr” (a doctor) is {d i k t aw w_a r} which will have the following pronunciation variants given the available alternatives for the model w_a in Table 7.2:

Table 7.3: Forced-alignment results for the distributions of skippable and ambiguous models. X_0 represents skippable models while X_a represents ambigious models.

unit	% of data	aligned model	% of unit
a_0	2.02	a_0 (skip path)	64.69
		a_0 (non-skip path)	22.71
		e	8.13
		aa (long vowel)	4.48
p_0	1.19	p_0 (skip path)	55.47
		p_0 (non-skip path)	15.34
		t	14.94
		h	11.47
		a (short vowel)	2.77
l_0	0.46	l_0 (skip path)	50.38
		l_0 (non-skip path)	32.72
		l	16.90
y_a	1.60	iy (long vowel)	71.50
		y (glide)	28.50
w_a	1.05	uw (long vowel)	63.82
		w (glide)	36.18

diktawr : d i k t aw w r
diktawr : d i k t aw uw r

All permutations are listed as pronunciation variants if a word has more than one ambiguous or skippable model. Hence, the number of pronunciation variants increases exponentially with the number of ambiguous models within the initial pronunciation. In the current analysis, this resulted in a pronunciation dictionary with an average of 2.6 pronunciations per word. The initial models are then used to disambiguate the pronunciation of the ambiguous and skippable models using forced alignment. Table 7.3 shows the distribution of chosen pronunciations for each ambiguous and skippable model based on the alignment results. More than two thirds of the ambiguous w_a and y_a are aligned as long vowels. At least 50% of the skippable models tend to be aligned using the skip path within the model. This skip was modelled with a one-state path, i.e. at the expense of one frame. This waste of frames can be an issue when there is a high frequency of skips and should be avoided by removing the occurrences of these models when this state is chosen.

As a result, new pronunciations were generated where all skippable models were removed when they were aligned using the path containing the skip state. Ambiguous models were replaced by the chosen unit during the forced alignment. This resulted in 41 units which were retrained using a standard left-to-right HMM with 3 states, i.e. no special topology for skippable models was used.

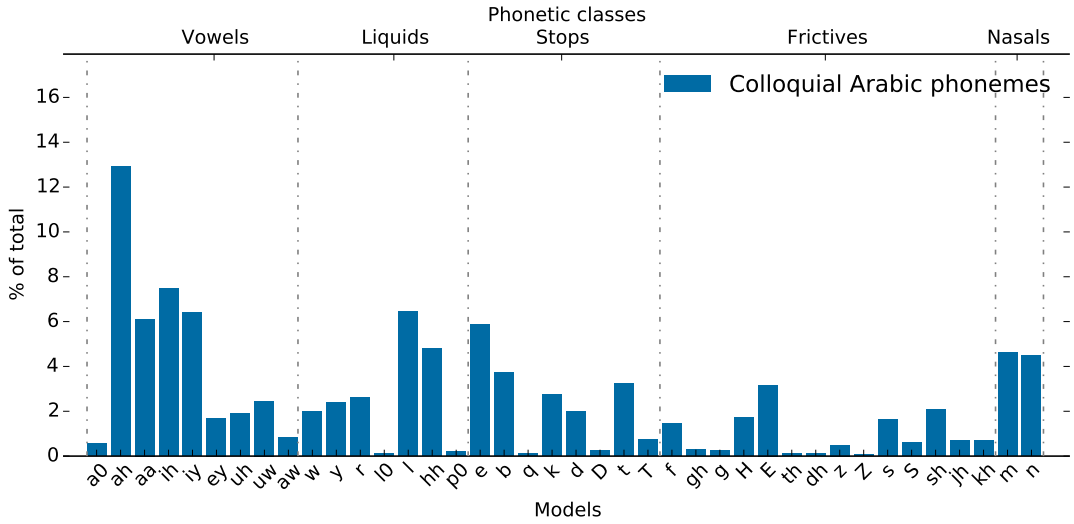


Figure 7.6: Frequency of the new acoustic units. Phonetic classes are labelled on the top x-axis while models belong to each class is shown on the bottom x-axis between enclosed between the dashed vertical lines.

The final acoustic model set is composed of nine vowel models as opposed to seven in the diacritised grapheme set, including two ambiguous models; seven liquid phoneme models as opposed to six models, including three ambiguous; and two potentially silent models in diacritised grapheme set, eight stop phoneme models as opposed to 13 models in diacritised grapheme set. The number of fricative and nasal phoneme models remains the same. Figure 7.6 illustrates the frequency distribution of the new acoustic set on the training data from AppenLCA. In comparison to the diacritised grapheme distribution (shown in Figure 7.4), the new acoustic set introduces new models mostly for the vowel and liquid phonetic classes towards which the distribution of diacritised graphemes was skewed. Also, it merges the models in the stop phonetic classes where most of these models are infrequent. This reduces the skewness of the original grapheme distribution and makes it more balanced.

Sensitivity to the dialect's phonotactics

As aforementioned dialects differ in their phonology and lexicon. Thus, it is interesting how phonotactics change across different CA dialects. In a contrastive experiment, the behaviour of the proposed acoustic set and transcriptions are compared with an experiment in English conversational speech. Similar training and testing sets were chosen from three additional Arabic dialects, Gulf (GCA), Iraqi (ICA) and MSA, and from five Northern American English dialects, Canadian (CAN), midland (MID), northern (NTH), southern (STH) and western (WST) (details can be found in Appendix B).

Table 7.4: Perplexity across Arabic dialects for phonotactic language models (PLM) of orders (n value) 2, 3, 4 and 5 grams in (a), (b), (c) and (d) respectively.

(a) n=2

	test sets			
PLM	GCA	ICA	LCA	MSA
GCA	14.7	15.3	16.7	16.7
ICA	15.7	14.7	17.6	17.6
LCA	21.8	21.1	14.0	21.4
MSA	25.6	27.2	26.7	12.4

(b) n=3

	test sets			
PLM	GCA	ICA	LCA	MSA
GCA	10.6	12.2	13.1	16.5
ICA	12.6	10.5	15.1	18.1
LCA	16.4	17.1	9.6	20.6
MSA	68.0	71.6	67.6	6.3

(c) n=4

	test sets			
PLM	GCA	ICA	LCA	MSA
GCA	8.6	10.5	11.4	16.1
ICA	11.0	8.4	14.0	18.7
LCA	14.4	15.4	7.1	20.5
MSA	149.4	162.0	146.3	2.7

(d) n=5

	test sets			
PLM	GCA	ICA	LCA	MSA
GCA	7.9	10.0	11.0	16.1
ICA	10.5	7.8	13.8	18.8
LCA	14.2	15.6	6.1	20.3
MSA	181.3	194.1	173.0	1.8

For each Arabic dialect, an acoustic set was derived and trained using the proposed process as described for LCA dialect, while a standard training recipe was used for the English sets, where a conventional pronunciation dictionary was employed. Model-level transcriptions were obtained by force-aligning the word-level transcription with its corresponding acoustic set. These model-level transcriptions were used as training materials for estimating a phonotatctic language model (PLM). Estimating an PLM uses a similar approach to building an n-gram statistical language model. Instead of using word sequences as training samples, acoustic unit sequence for each utterance were employed. Using the SRILM toolkit (Stolcke, 2002), standard 5-gram PLMs were estimated, with Witten-Bell smoothing applied. For language models, only perplexity was used to describe the uncertainty within the models. Table 7.4 compares the perplexities of language models which were estimated using the four Arabic dialects and computed on the model-level transcription of test sets with orders of 2, 3, 4 and 5. Evidently, the phonotactics of CA differ from those of MSA, which is evident in the perplexities when applying on any CA PLM on the MSA test set. For instance, when using a trigram PLM (Table 7.4b) trained on ICA, the estimated perplexity on any of the CA test sets ranged between 10.5-15.1 as opposed to 18.1 when the perplexity was estimated on the MSA test set. This difference increased considerably when estimating the perplexity using an MSA PLM on a CA test set, which ranged between 67.6-71.6 as opposed to 6.3 on the MSA test set. Generally, the perplexity decreases as the PLM order increases, for example, By increasing the order from 2- (Table 7.4a) to 5-gram (Table 7.4d), the perplexity computed using a GCA PLM on a LCA test

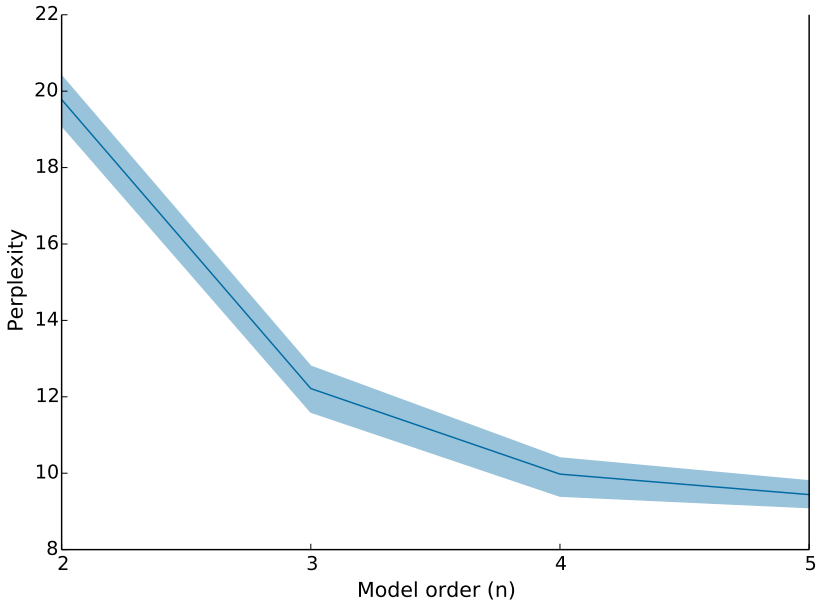


Figure 7.7: Perplexity (y-axis) across English dialects for phonotactic language models of order 2 to 5 grams (x-axis). Average perplexities is shown in solid line while the enclosed area above and below indicating the range of the computed perplexities.

set decreased from 16.7 to 11.0. This lowest perplexity was obtained by applying a PLM on a test set with a matching dialect. Results are shown as the diagonal in the Tables 7.4. From the trend of these perplexities, it can be concluded that Gulf and Iraqi dialects are closer to each other than to the remaining Arabic dialects since they yielded the lowest perplexity in cross-dialect tests, in contrast to other dialects. Conversely, this diversity in perplexity cross-test seen in Arabic is not observed on the North American English dialects. Figure 7.7 shows the average perplexity computed across dialects with the minimum and maximum values found in these tests.

If two language models predict similar sequences or generate similar perplexities, it can be considered that they were estimated from related languages. Similarly, if two phonotactic language models generate similar perplexities, the languages used for their estimation should be considered related as well. Therefore, in an attempt to quantify the relationship between the phonotactics of different dialects, the correlation between perplexities on a test set was compared by unit. For a given test set, the perplexity was computed for each unit in the given transcription using a phonotactic language model. In an acoustic set of size N, the perplexity is computed for each unit, a_i and $i \in |\{1, ..., N\}$, on a test set using an $\mathrm{PLM_A}$ of order n, formally:

$$UnitPPL_{\mathrm{PPL_A}}(a_i) = -\frac{1}{M}\sum_{m=1}^{M} \log\ P_{\mathrm{PPL_A}}(a_i|h_{n-1}) \tag{7.1}$$

where M is the number of occurrences of the unit a_i in the test set and h_{n-1} is the a sequence of $n-1$ units preceding u_i. Computing this perplexity for each unit a_i in the acoustic set, using Equation 7.1, a vector is constructed as follows:

$$uppl_A = [UnitPPL_{\mathrm{PPL_A}}(a_i)] \qquad 1 \leq i \leq N, \tag{7.2}$$

Two vectors of perplexities for the same acoustic set are constructed: $uppl_A$ and $uppl_B$. $uppl_A$ is computed with $\mathrm{PLM_A}$ and $uppl_B$ is computed using $\mathrm{PLM_B}$. Both PLMs are of the same order n. Then correlation between the perplexities of these units is computed as follows:

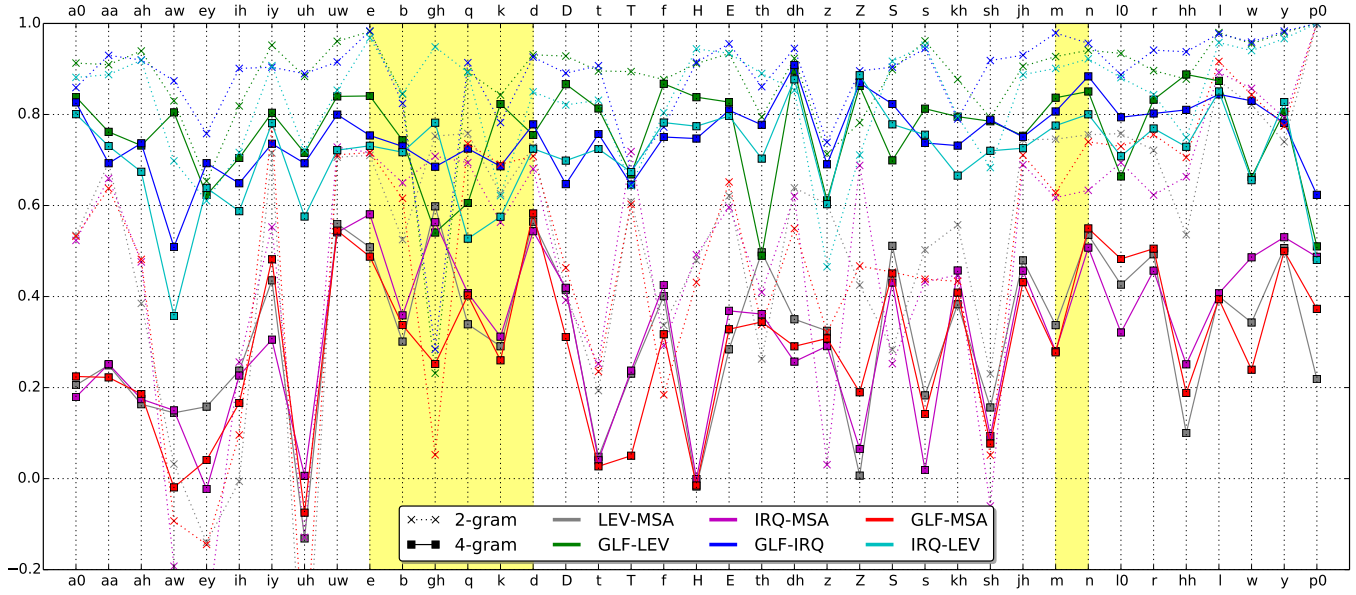
$$\mathrm{Correlation}(\mathrm{PLM_A}, \mathrm{PLM_B}) = \frac{\mathrm{Cov}(uppl_A, uppl_B)}{\sqrt{\mathrm{Cov}(uppl_A, uppl_A) * \mathrm{Cov}(uppl_B, uppl_B)}} \tag{7.3}$$

where $\mathrm{Cov}(uppl_A, uppl_B)$ is the covariance between the two vectors: $uppl_A$ and $uppl_B$. It is defined as:

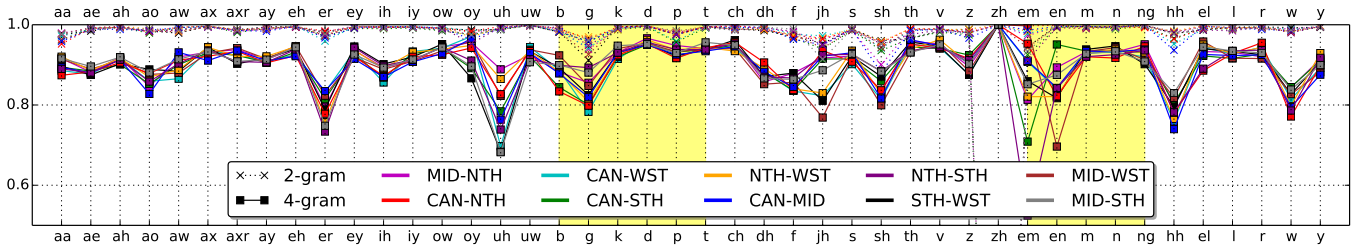
$$\mathrm{Covariance}(uppl_A, uppl_B) = \frac{1}{N-1}\sum_{i=1}^{N}(uppl_A^{(i)} - \mu_A) * (uppl_B^{(i)} - \mu_B) \tag{7.4}$$

where N is the size of the acoustic unit set. μ_A and μ_B are the mean of the the unit perplexity vectors $uppl_A$ and $uppl_B$, respectively. $uppl_A^{(i)}$ and $uppl_B^{(i)}$ are the perplexity of the unit u_i using $\mathrm{PPL_A}$ and $\mathrm{PPL_B}$, respectively.

Figure 7.8 depicts these correlations for Arabic and English dialects using GCA and CAN test sets respectively. For brevity, only perplexity produced from bigram and quadgram PLMs are shown. Generally in both languages, a stronger correlation is observed between bigram sequences than between quadgram sequences, suggesting that the dialects tend to be more distinguishable on wider context rather than a local context. For Arabic phonotactic correlation, shown in Figure 7.8a, it was obvious that there were two bands of correlations: one between CA dialects and MSA and the other among the CA dialects themselves. It can be seen that the correlation between dialects and MSA was weak, as low as -0.5 for bigram PLMs and it can reach -0.1 for quadgram PLMs, especially for the vowels, H, $ and h which occur more often in CA (as exclusive affixes which do not exist in MSA). On the other band, dialects showed stronger correlation between each other which reached 0.2 for bigram PLM and 0.4 for quadgram PLM. This drastic difference in correlation between all four Arabic dialects generally was not observed in English at all. All English dialects behaved in the same manner toward the test set and generated similar perplexities regardless of the dialect used in the acoustic set.



(a) Correlation of Arabic perplexities by unit computed on a Gulf test set. Arabic dialects: Gulf (GLF), Iraqi (IRQ), Levantine (LEV) and MSA.



(b) Correlation of English perplexities by unit computed on a Canadian test set. Northern American dialects: Canadian (CAN), Midlands (MID), Northern (NTH), Southern (STH) and Western (WST).

Figure 7.8: Correlation of unit perplexities in different dialects using PLMs. Each line shows the correlation (y-axis) for each unit (x-axis) between two PLMs (line colors) using different orders (line markers). The scale used for y-axis is the same for both plots (a) and (b). Alternating background in the x-axis shows the phonetic classes: vowels, stops, fricatives, nasal and liquids, respectively.

Acoustic confusablility across dialects

To assess the impact of using these models and the acoustic confusablility observed across different dialects in the model set, a test set was force-aligned on the model-level where a given unit can be assigned to any unit within the acoustic set without any restrictions, i.e. regardless of the context. In this experiment, referred to as unconstrained forced alignment, the number of units to be recognised is known for each test utterance along with the existence of silence units [5]. Two metrics were employed: unit error rate and frame error rate. Unit error rate (UER) is the edit distance between the reference and the aligned sequence; however, only substitution errors are considered in the unconstrained force-alignment test because the number of the units per test utterance does not change with any insertions or additions. Frame error rate (FER) is the percentage of misclassified frames in the test set.

As Table 7.5 shows, the cross-test results for Arabic dialects, the best performance achieved is when the dialect of an acoustic set matches that of the test set. Such a preference for matching the acoustic set and the test set was not observed in the English cross-tests overall. The performance of aligning the MSA acoustic set to the MSA test set significantly outperformed the other tests by an average of 25% UER and 30% FER absolute, possibly due to the nature of the data, as it was recorded in a controlled and quiet environment. This differs from CA sets which were drawn from CTS conversation without any control over the recording environment or channels. Moreover, the mapping between phonemes and graphemes in MSA is more stable and simpler than CA, as discussed earlier in Section 2.4. The similarity between GCA and ICA based on their PLM perplexities observed above is confirmed by the UER and FER scores obtained from aligning the GCA test set using the ICA acoustic set and vice versa.

One of the important distinctions between the unconstrained force-alignment test and the context-free phone recognition test is that the number of units to be recognised in each utterance is known in the former, along with the existence of silence units. Without any constraints on the number of units in the utterance, insertion and deletion errors occur. The unknown location of silence models causes the recognition performance to be dependent on the quality of the silence model within the acoustic set. The lower quality of silence model, the more deletion errors occur (because most of the frames will be *incorrectly* assigned to the silence model) while more insertions occur with better silence models. Such a case can be clearly observed when the MSA acoustic set was used, due to the higher quality of the audio recordings, where UER exceeded 100% due to insertions. However, FER

[5]Such information is not known in a context-free phone recognition task where no context information was used.

Table 7.5: Unit error rate (UER) and frame error rate (FER) for various test settings across Arabic dialects using the proposed phonetic acoustic inventory.

(a) Unconstrained forced-alignment

AM	Test sets							
	GCA		ICA		LCA		MSA	
	UER	FER	UER	FER	UER	FER	UER	FER
GCA	**78.2**	**54.2**	81.8	58.1	81.4	61.0	86.4	66.8
ICA	80.6	59.0	**79.1**	**49.7**	82.3	61.5	83.9	59.3
LCA	82.0	62.5	83.4	61.2	**79.3**	**55.0**	84.6	63.0
MSA	91.0	77.3	90.3	73.5	90.0	75.7	**53.7**	**24.2**

(b) Context-free phone recognition

AM	Test sets							
	GCA		ICA		LCA		MSA	
	UER	FER	UER	FER	UER	FER	UER	FER
GCA	**81.9**	**57.8**	**63.2**	60.2	**63.5**	63.2	74.0	69.6
ICA	84.3	62.2	75.7	**52.5**	80.1	63.8	94.8	63.4
LCA	91.5	65.6	86.5	63.5	78.1	**57.8**	105.1	66.8
MSA	96.0	77.5	89.3	73.8	90.3	75.8	**38.6**	**26.5**

(c) Phone recognition with bigram PLM

AM & PLM	GCA		ICA		LCA		MSA	
	UER	FER	UER	FER	UER	FER	UER	FER
	Test sets							
GCA	61.7	**54.6**	62.8	57.9	63.3	61.8	74.2	65.1
ICA	**61.6**	57.2	**56.7**	**47.4**	61.4	60.2	67.4	56.8
LCA	64.0	61.3	63.2	60.3	**57.3**	**52.7**	68.0	59.5
MSA	73.4	76.3	70.5	72.5	70.1	73.9	**26.0**	**21.0**
AM	*PLM & Test set*							
GCA	**61.7**	**54.6**	60.2	55.6	60.2	58.3	68.5	59.6
ICA	62.9	58.8	**56.7**	**47.4**	60.4	59.1	63.4	53.2
LCA	64.8	62.2	61.8	59.2	**57.3**	**52.7**	63.8	55.0
MSA	74.4	78.6	67.1	70.9	67.3	72.5	**26.0**	**21.0**
PLM	*AM & Test set*							
GCA	61.7	**54.6**	59.0	49.2	59.3	54.7	28.8	24.1
ICA	**59.9**	52.9	**56.7**	**47.4**	57.6	53.1	27.5	22.5
LCA	59.9	53.0	57.3	48.0	**57.3**	**52.7**	27.4	22.4
MSA	60.9	54.5	58.5	49.3	58.6	54.2	**26.0**	**21.0**

Table 7.6: Unit error rate (UER) and frame error rate (FER) for various test settings across English dialects using a phonetic acoustic inventory.

(a) Unconstrained forced-alignment

AM	Test sets									
	CAN		MID		NTH		STH		WST	
	UER	FER	UER	FER	UER	FER	UER	FER	UER	FER
CAN	88.0	66.1	84.2	61.3	86.5	61.4	90.6	71.4	85.6	61.9
MID	86.9	65.3	83.6	60.4	86.1	61.0	89.2	70.7	83.9	59.3
NTH	87.0	64.8	83.5	59.6	89.1	70.0	89.8	71.3	84.9	60.1
STH	87.4	66.0	83.8	60.5	86.0	61.9	85.2	59.6	84.9	59.8
WST	87.5	65.5	84.6	61.3	86.1	61.2	89.7	71.3	84.6	59.4

(b) Context-free decoding

AM	Test sets									
	CAN		MID		NTH		STH		WST	
	UER	FER	UER	FER	UER	FER	UER	FER	UER	FER
CAN	96.1	65.1	107.3	66.0	102.6	65.6	101.3	70.4	102.8	63.5
MID	96.1	65.0	100.4	65.1	97.6	65.3	94.6	68.4	95.3	60.3
NTH	97.6	65.5	100.3	64.4	94.7	63.6	96.8	70.7	91.4	61.5
STH	95.5	66.0	98.7	64.8	97.0	66.8	98.2	68.6	91.8	59.7
WST	102.0	65.7	107.0	66.6	97.7	65.9	99.6	70.7	91.8	60.2

(c) Phone recognition with bigram PLM

AM & PLM	CAN		MID		NTH		STH		WST	
	UER	FER	UER	FER	UER	FER	UER	FER	UER	FER
	Test sets									
CAN	71.9	55.8	66.3	58.0	70.7	56.5	74.2	62.7	65.5	54.0
MID	72.4	54.5	67.3	60.3	70.4	58.5	73.3	60.5	65.2	51.2
NTH	72.0	55.0	66.4	55.8	69.4	56.4	73.5	62.8	64.7	52.8
STH	72.9	56.7	67.8	57.2	71.5	59.8	73.1	58.7	65.2	51.3
WST	71.8	55.4	67.3	57.5	69.8	58.7	72.7	62.2	63.9	51.5
AM	*PLM & Test set*									
CAN	71.9	55.8	66.3	58.1	70.7	56.5	74.1	62.2	65.4	54.0
MID	72.4	54.4	67.3	60.3	70.3	58.3	73.3	60.3	65.3	51.5
NTH	71.9	54.7	66.2	55.9	69.4	56.4	73.4	62.9	64.8	52.9
STH	72.9	57.1	67.8	57.1	71.5	59.9	73.1	58.7	65.3	51.3
WST	71.7	55.3	67.3	57.5	69.7	58.9	72.8	62.4	63.9	51.5
PLM	*AM & Test set*									
CAN	71.9	55.8	67.2	59.9	69.2	56.3	73.1	58.7	63.9	51.6
MID	71.9	55.5	67.3	60.3	69.4	56.3	73.1	58.7	63.9	51.4
NTH	71.9	55.3	67.2	60.3	69.4	56.4	73.1	58.8	63.8	51.5
STH	71.9	55.3	67.3	60.2	69.4	56.7	73.1	58.7	63.9	51.6
WST	71.9	55.8	67.3	60.2	69.4	56.9	73.1	58.9	63.9	51.5

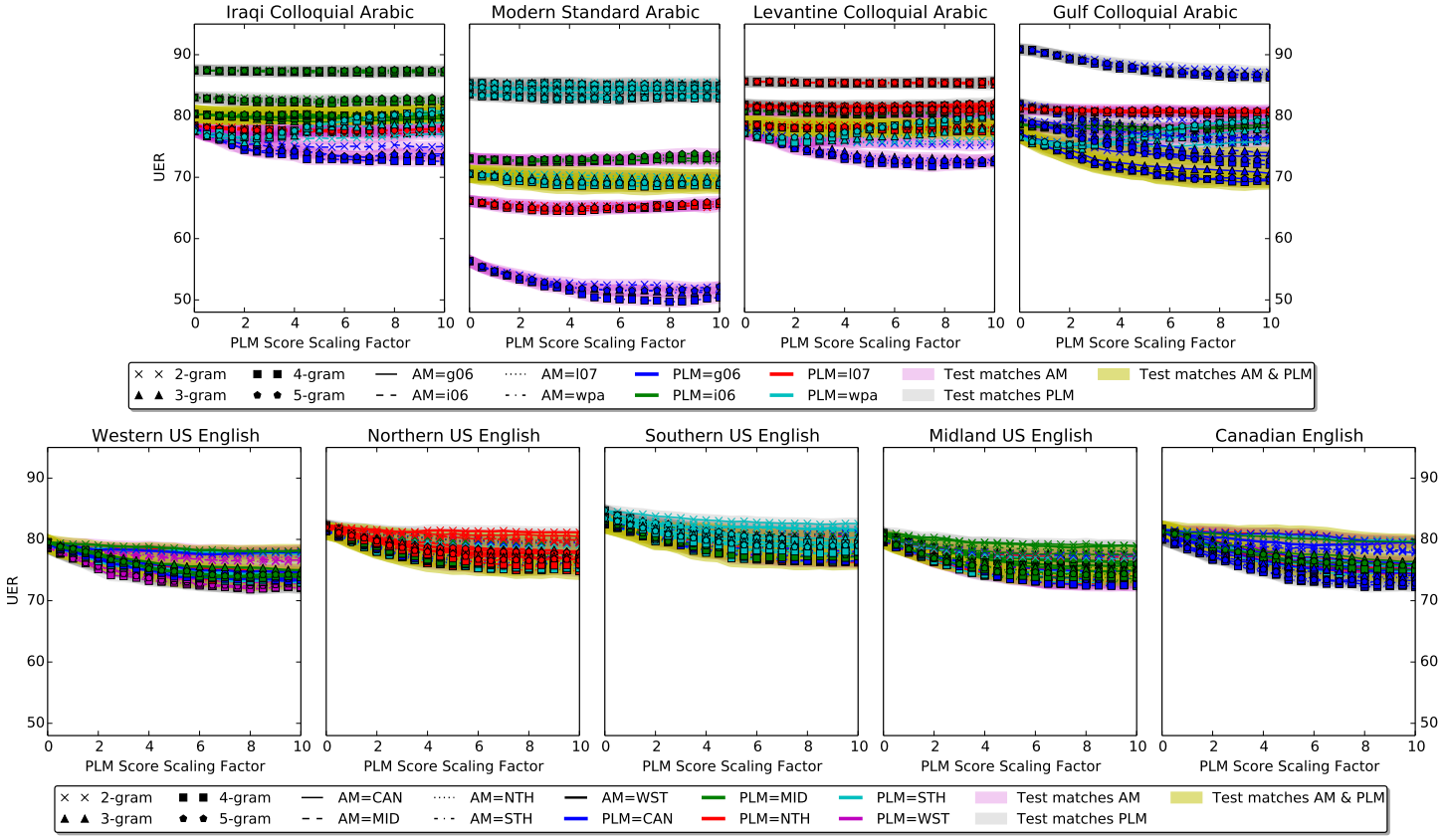


Figure 7.9: UER of Lattice re-scoring experiments. Each plot represents a test set of a certain dialect, while lines show the PER (y-axis) over increasing the grammar scaling factor (x-axis) when using different combination of acoustic model (line styles) and PLM (line colors) with different orders (line markers). Arabic dialects: Gulf (GLF), Iraqi (IRQ), Levantine (LEV) and MSA. Northern American English dialects: Canadian (CAN), Midlands (MID), Northern (NTH), Southern (STH) and Western (WST).

always degrades by 0.1-5.1% absolute in context-free phone recognition in comparison to unconstrained force-alignment for both languages. In this test, the best performance was obtained in terms of FER when the dialect of the acoustic set matched the dialect of the test set as in the unconstrained force-alignment test. This was not the case if the UER metric was used where using GCA acoustic set seems to be preferred by all other CA dialects.

Considering a local context as small as two units (as in bigram PLM) improves the performance for phone recognition significantly in terms of UER and FER, as expected . UER improved by at least 12.6% absolute and FER by at least 3.2% absolute for the Arabic and English dialects. For any combinations of resources (Acoustic models, PLM, test set), four different cases were tested:

- The dialect of acoustic models and the dialect of PLM must be matched.
- The dialect of acoustic models and the dialect of test set must be matched.
- The dialect of PLM and the dialect of test set must be matched.
- The dialect of all resources must be matched.

Overall, using a matching acoustic set and PLM to a test set had the best performance for Arabic dialects (the last case). Using this case as a baseline, using PLM matched to the test set was not as effective as using a matching acoustic set in CA, where UER degrades by 1.2-7.7% absolute in the former case and only by 0.3-2.8% absolute in the latter. Again, such a preference toward matching dialects between resources was not observed in English dialects.

In order to incorporate context information in the unconstrained force-alignment test, a lattice was generated for each test set using an acoustic set, where only bigram context were considered in order to restrict the lattice size. Afterwards, the lattice was re-scored using PLMs with different orders. As in the phone recognition task with bigram PLM, the same four combinations of resources were tested. In addition, the scores obtained from PLM are scaled using grammar scaling factor to evaluate the impact of increasing the reliance on the phonotactic of a certain dialect on the recognition performance.

Figure 7.9 visualises UER plots for each setting on each test set for both languages. Each plot represents a test set where each line shows the UER for a test setting combination of acoustic set and PLM order that progresses over an increasing grammar scaling factor for the PLM scores. Generally, higher order PLMs have a positive impact on recognition performance as long as it did not suffer from data sparsity, as in 5-g PLMs. This observation is consistent in all English test sets, but not always true for Arabic dialects

where the improvement, if any, was marginal. This confirms the conclusion from the phone recognition task that changing the PLM has less effect on the recognition performance than changing the acoustic set.

7.3 Context-independent multi-phoneme graphemes and restructuring the acoustic inventory

In the previous section, only issues that can be resolved via contextual information derived from either graphemic representation or from acoustic information were investigated. However, another issue that arises for CA are homophone words, where several words differ in their written form but share the same pronunciation, which stems from the new *phonemic* mapping between consonants. For example, the phoneme /g/ maps to both graphemes "q" and "j", leading to homophone words such as "qryt" (I read) and "jryt" (I ran) which are both pronounced as /gri:t/. This is only a problem in CA as these examples are not homophone words in MSA. This ambiguity in the mapping is not context-dependent, so the chosen phoneme cannot be identified from the surrounding context. Moreover, these mappings are inconsistent within a dialect and sometimes even for the speakers themselves. Therefore, these graphemes will be referred to as context-independent multi-phoneme graphemes.

To visualise how such cases can be problematic for acoustic modelling, three context-dependent models were chosen from the trained acoustic set used in the Section 7.2: *s*, *th* (pronounced as /θ/) and *iy*. These three models do not share any of their states with each other but they have similar context in terms of phonetic classes, i.e. on the left they have a glide and on the right silence. Figures 7.10 and 7.11 compare the PLP features and their first derivatives (in the first 26 dimensions) represented by these three models. Due to the difficulty in visualising a multivariate Gaussian[6], each dimension was plotted individually with a separate Gaussian for each state. As shown in the figure, distributions that belong to models *th* and *s* are closer to each other than those belonging to model *iy*, and almost identical in several dimensions. As it has been shown in Figure 7.3, *th* and *s*, representing the nominated graphemes "v" and "s", share the phoneme /s/ in an inconsistent manner. It is not possible to predict which phoneme should be chosen from the context. The similarity between the models *s* and *th* shows that these two models can actually represent the same phoneme /s/.

To resolve this ambiguity and reduce the number of models competing over the same acoustic value, these two models should be represented by one model. Hypothetically, all triphones representing the three phonemes /iy/, /s/ and /th/ can be plotted in some

[6]which in this case would have 39 dimensions: 13 PLP features and their first and second derivatives.

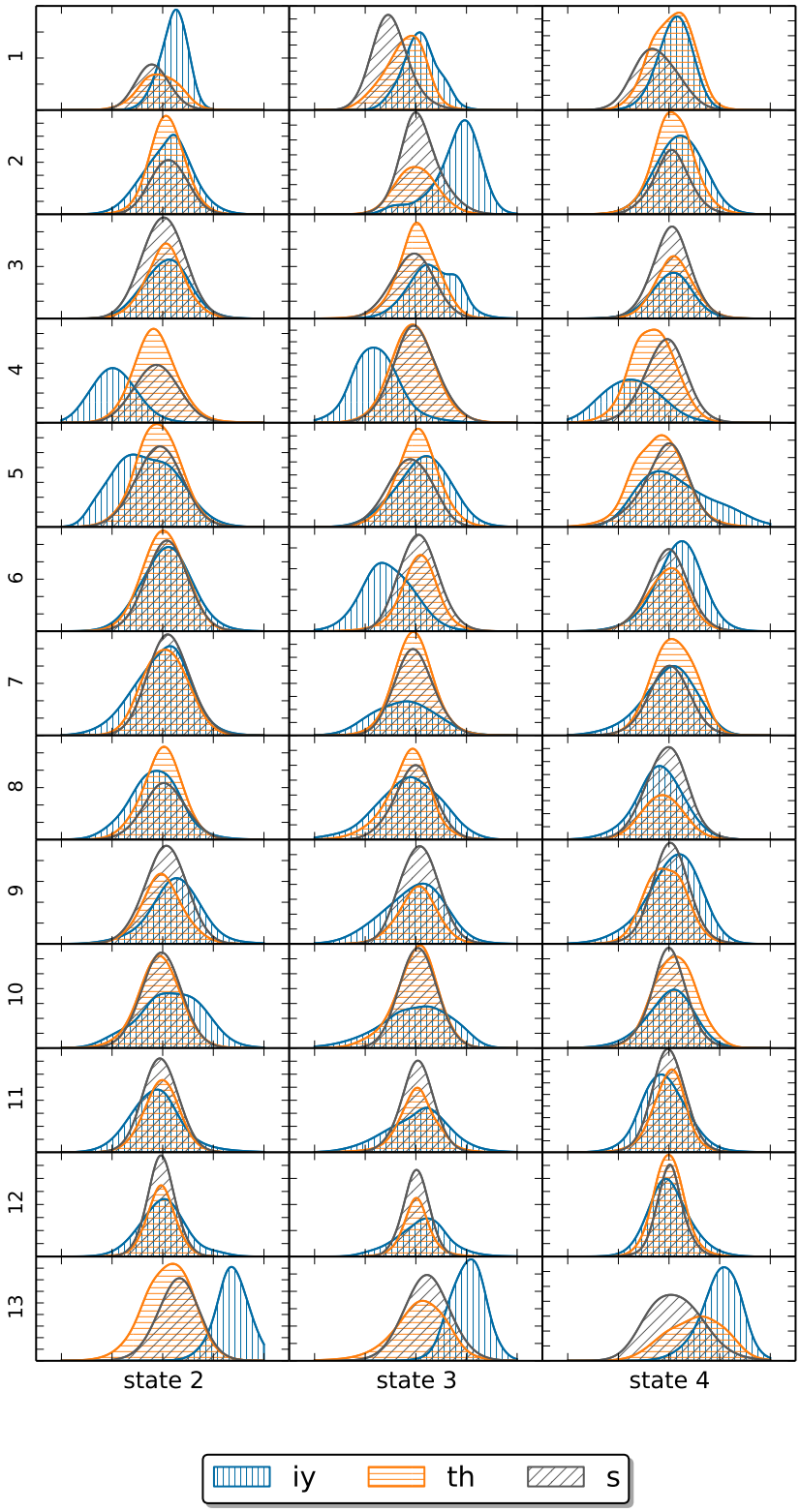


Figure 7.10: Comparison of the PLP features (first 13 dimensions) represented by three acoustic models: *iy* (vertical blue), *th* (horizontal orange) and *s* (diagonal gray). Each model has 32 Gaussian mixtures. The models *s* and *th* are closer to each other than to *iy*.

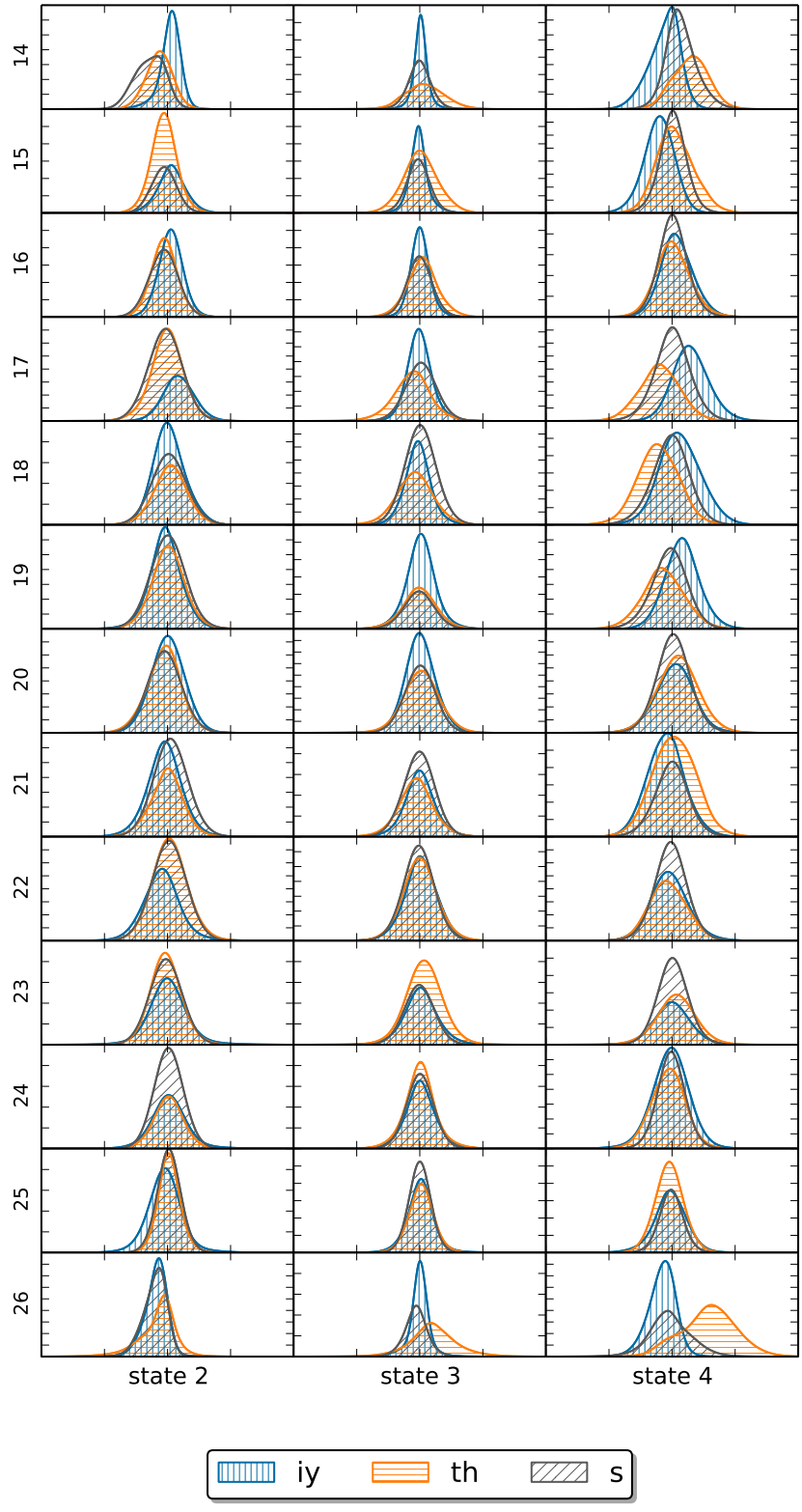


Figure 7.11: Comparison of the first derivatives of the PLP features represented by three acoustic models: *iy* (vertical blue), *th* (horizontal orange) and *s* (diagonal gray). Each model has 32 Gaussian mixtures. The models *s* and *th* are closer to each other than to *iy*. The scale of the x-axis is different from that of Figure 7.10 .

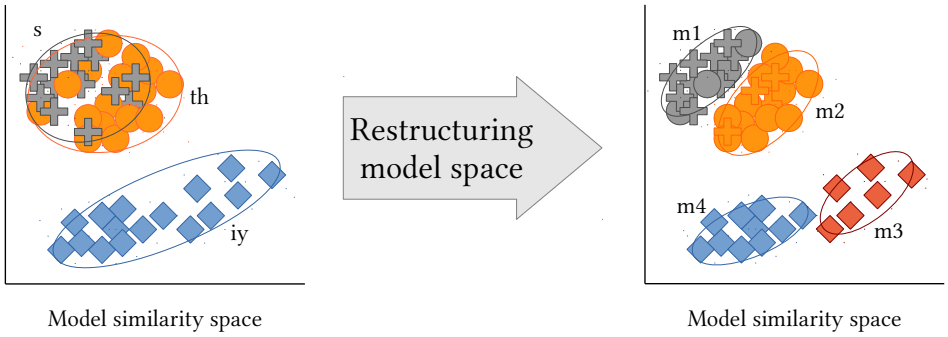


Figure 7.12: A hypothetical illustration of all triphones of the three models /s/, /th/ and /iy/ in Figure 7.10 and Figure 7.11 in some model similarity space. (left) current situation of the three models; (right) optimum model separation.

similarity space where the smaller a distance between two data points (or triphones), the similar these units are. Figure 7.12 illustrates this hypothetical plot and shows the optimum model space where similar units are grouped together. In order to reveal which models are representing the same phoneme, the model space can be organised based on some distance metric. Based on the (dis)similarity metric, similar models can be clustered together and substituted by a single model. Hartmann et al. (2013) proposed a strategy to discover acoustic units and create a pronunciation dictionary from an initial grapheme-based ASR system automatically. In their method, a similarity matrix was constructed from context-dependent grapheme units. Then spectral-based clustering was used to group similar units into clusters. These clusters represent the discovered units. Then, translation rules from a grapheme sequence to a discovered unit sequence were obtained using an SMT-based approach. Not all translation rules were kept. Each rule was evaluated based on average log-likelihood during force-alignment after applying that rule on the pronunciation dictionary. Only rules that improved the average log-likelihood were kept and used for pronunciation generation.

In the following sections, a modified version of Hartmann et al. (2013)'s strategy is proposed for restructuring the model space into less acoustically confused units whilst retaining phonemic separation. Starting from the context-dependent acoustic models, derived in Section 7.2, a dissimilarity matrix is computed (Section 7.3.1). After that, a hierarchical agglomerative (bottom-up) clustering is applied (Section 7.3.2) to generate hierarchical decomposition for the original context-dependent acoustic models where the extraction of the new units (represented by clusters) takes place. Each unit is substituted by its assigned cluster name in the training transcription by which all different pronunciation variants can be obtained for each word (Section 7.3.3), then finally a G2P model is trained based on these new variants (Section 7.3.6).

7.3.1 HMM similarity

An HMM, h, is defined by three parameters:

- the initial state distribution, λ;
- the state transition probability matrix, $\mathbf{A} = [a_{i,j}]$ where $a_{i,j}$ is the transition probability from state i to state j, s.t. $\sum_j a_{i,j} = 1$;
- the output emission probability distribution for each state, $B = [b_i(o_t)]$ where $b_i(o_t)$ is the probability of an observation o_t being generated by the state i.

To compute the similarity between two HMMs, h and h' with I and J states respectively, two components must be considered: the similarity between the output distributions within the states and the similarity in the transition between these states. Hartmann et al. (2013) formulated this as follows:

$$\text{HMM}_{\text{sim}}(h, h') = \sum_{i=2}^{I-1} \sum_{j=2}^{J-1} \alpha_{ij} \text{ Similarity}(h_i, h'_j) \tag{7.5}$$

where h_i and h'_j are the i^{th} and j^{th} states within h and h' respectively, starting from the first to the last emitting states. α_{ij} approximates the probability of simultaneously being in states i and j of the two HMMs h and h' respectively. Similarity(h_i, h'_j) computes the similarity between output probability distributions within the states h_i and h'_j. This can be expressed for example in terms of their divergence, $D(h_i||h'_j)$, as one of the following expressions:

$$\text{Similarity}(h_i, h'_j) = (D(h_i||h'_j) + 1)^{-1}, \tag{7.6}$$

$$\text{Similarity}(h_i, h'_j) = \exp(-k\ D(h_i||h'_j)), \tag{7.7}$$

$$\text{Similarity}(h_i, h'_j) = \exp(\frac{-D(h_i||h'_j)^2}{k^2}), \tag{7.8}$$

where k in Equations 7.7 and 7.8 is a constant scaling parameter. $D(h_i||h'_j)$ can be one of the Kullback-Leibler divergence (KLD) variants, such as Sahraeian and Yoon (2011), or any other divergence which can be computed (or approximated) between the Gaussian mixtures models.

Divergence between output probability distributions

Following Hartmann et al. (2013), the Cauchy-Schwarz divergence (CSD) was chosen since it has a closed-form solution and has been shown to perform comparably to the KLD (Kampa et al., 2011), while the latter cannot be computed analytically for GMMs. The CSD between two GMMs is defined as follows:

$$\begin{aligned} \text{CSD}(p||q) &= -\log \frac{\int p(x)q(x)dx}{\sqrt{\int p(x)^2dx \int q(x)^2dx}}, \\ &= -\log \int p(x)q(x)dx + 0.5 \log \int p(x)^2dx + 0.5 \log \int q(x)^2dx, \end{aligned} \quad (7.9)$$

where $p(x)$ and $q(x)$ are GMM distributions with different parameters for each:

$$p(x) = \sum_{m=1}^{M} \pi_m \ \mathcal{N}(x; \mu_m, \Lambda_m^{-1}) \quad (7.10)$$

and

$$q(x) = \sum_{k=1}^{K} \tau_k \ \mathcal{N}(x; \nu_k, \Omega_k^{-1}) \quad (7.11)$$

and $\mathcal{N}$ is a multivariate Gaussian distribution of dimension D which is given by:

$$\mathcal{N}(x; \mu_m, \Lambda_m^{-1}) = \frac{|\Lambda_m|^{0.5}}{(2\pi)^{0.5D}} \exp\left(-0.5(x-\mu_m)^\top \Lambda_m (x-\mu_i)\right) \qquad : x \in \Re^D \quad (7.12)$$

The product of two Gaussian distributions is also Gaussian (derivation is provided in Appendix D):

$$\begin{aligned} p(x)q(x) &= \mathcal{N}(x; \mu, \Lambda)\mathcal{N}(x; \nu, \Omega) \\ &= \mathcal{N}(\mu; \nu, \Lambda + \Omega)\mathcal{N}(x; \xi, \Phi) \end{aligned} \quad (7.13)$$

where $\Phi = (\Lambda^{-1} + \Omega^{-1})$ and $\xi = \Phi(\Lambda^{-1}\mu + \Omega^{-1}\nu)$. Hence, the integral of two Gaussian distributions in Equation 7.9 can be computed as:

$$\int p(x)q(x)dx = \int \mathcal{N}(x;\mu,\Lambda)\mathcal{N}(x;\nu,\Omega)dx \tag{7.14}$$
$$= \int \mathcal{N}(\mu;\nu,\Lambda+\Omega)\mathcal{N}(x;\xi,\Phi)dx \tag{7.15}$$
$$= \mathcal{N}(\mu;\nu,\Lambda+\Omega)\int \mathcal{N}(x;\xi,\Phi)dx \tag{7.16}$$
$$= \mathcal{N}(\mu;\nu,\Lambda+\Omega) \tag{7.17}$$

Equation 7.15 replaces the product of two Gaussian distributions in Equation 7.14 using the definition in Equation 7.13. Since the term $\mathcal{N}(\mu;\nu,\Lambda+\Omega)$ is independent of x, Equation 7.16 takes it outside the integral operation. The original distributions represent probability distributions, so $\int p(x)dx = 1$; hence Equation 7.17. Using this definition, in Equation 7.17, the closed-form expression of CSD$(p||q)$ (Equation 7.9) independently of x is as follows (Kampa et al., 2011) :

$$\begin{aligned}\text{CSD}(p||q) &= -\log\left(\sum_{m=1}^{M}\sum_{k=1}^{K}\pi_m\tau_k z_{mk}\right)\\ &+ 0.5\log(\sum_{m=1}^{M}\frac{\pi_m^2|\Lambda_m|^{0.5}}{(2\pi)^{0.5D}}) + 2\sum_{m=1}^{M}\sum_{m'<m}\pi_m\pi_{m'}z_{mm'}\\ &+ 0.5\log(\sum_{k=1}^{K}\frac{\tau_k^2|\Omega_k|^{0.5}}{(2\pi)^{0.5D}}) + 2\sum_{k=1}^{K}\sum_{k'<k}\tau_k\tau_{k'}z_{kk'}\end{aligned} \tag{7.18}$$

where:

$$z_{mk} = \mathcal{N}(\mu_m;\nu_k,(\Lambda_m^{-1}+\Omega_k^{-1})) \tag{7.19}$$
$$z_{mm'} = \mathcal{N}(\mu_m;\mu_{m'},(\Lambda_m^{-1}+\Lambda_{m'}^{-1})) \tag{7.20}$$
$$z_{kk'} = \mathcal{N}(\nu_k;\nu_{k'},(\Omega_k^{-1}+\Omega_{k'}^{-1})) \tag{7.21}$$

State occupancy distribution

One of the properties of Markov chains is that they are ergodic (Norris, 1998) by which the knowledge of the initial distribution, λ, fades over the time until it disappears at the end to reach a distribution known as stationary distribution. This distribution defines the state occupancy within the system over the time. However, ASR employs left-to-right HMMs, which each has an entry and an exit non-emitting states, so it is not infinite; therefore, it does not have a stationary distribution.

To approximate this distribution, Hartmann et al. (2013) suggested computing an *oc-*

cupancy matrix (denoted as α_{ij} in Equation 7.5) over a sequence of N steps. α_{ij} approximates the probability of simultaneously being in a pair of states i and j from the two HMMs h and h' respectively. In order to compute α_{ij}, the probability of being at state s at time t must be computed for each HMM. As time passes, i.e. t gets larger, the probability becomes smaller until it reaches zero because it is going toward exiting the state and the HMM eventually. Figure 7.13 illustrates this behaviour in a left-to-right HMM where the probability of the first emitting state to be occupied is high at the early steps in the sequence while it is zero for the other states, as the time passes, the probability of the first state to be occupied decreases as the probability to occupy the next state increases until the probability of occupying any of the states tends to be zero for greater values of t. Formally, the probability of being at state i in HMM h at time t,$P_h(s_t = i)$, is computed as follows:

$$P_h(s_t = i) = P_h(s_{t-1} = i) * a_{i,i} + P_h(s_{t-1} = i-1) * a_{i-1,i} \tag{7.22}$$

where the first term, $P_h(s_{t-1} = i) * a_{i,i}$, accumulates the probability of staying in the state (self-loop) while the second term, $P_h(s_{t-1} = i-1) * a_{i-1,i}$, accumulates the probability of arriving at the state i from all previous states. This computation always starts from the first state, i.e. $P_h(s_0 = 0) = 1$.

Since the event of being at state i of h at time t is independent of the event of being at state j of h' at the same time t, the probability of these two events occurring together is the product of the probabilities of each event individually. So α_{ij} can be expressed as:

$$\alpha_{ij} \quad = \quad \frac{\sum_t P_h(s_t = i) P_{h'}(s_t = j)}{\max\left(\sum_m \sum_t P_h(s_t = m), \sum_n \sum_t P_{h'}(s_t = n)\right)} \tag{7.23}$$

where $\sum_m \sum_t P_h(s_t = m)$ is the accumulated probability of being at any state m in the HMM h over the time t. The denominator is the maximum of this accumulation between the two HMMs h and h' and it acts as a normalisation factor to approximate the computed quantity into the desired probability where $0 \leq \alpha_{ij} \leq 1$.

Similarity matrix

Using Equation 7.5, a similarity matrix can be composed for a given set of HMMs where each element in the matrix represents the similarity between two HMMs. As a similarity metric, it satisfies the following conditions (proofs are shown in Appendix D):

- Symmetry: $\text{HMM}_{\text{sim}}(x, y) = \text{HMM}_{\text{sim}}(y, x)$
- Non-negativity: $\text{HMM}_{\text{sim}}(x, y) \geq 0$
- Triangle inequality: $\text{HMM}_{\text{sim}}(x, z) \leq \text{HMM}_{\text{sim}}(x, y) + \text{HMM}_{\text{sim}}(y, z)$

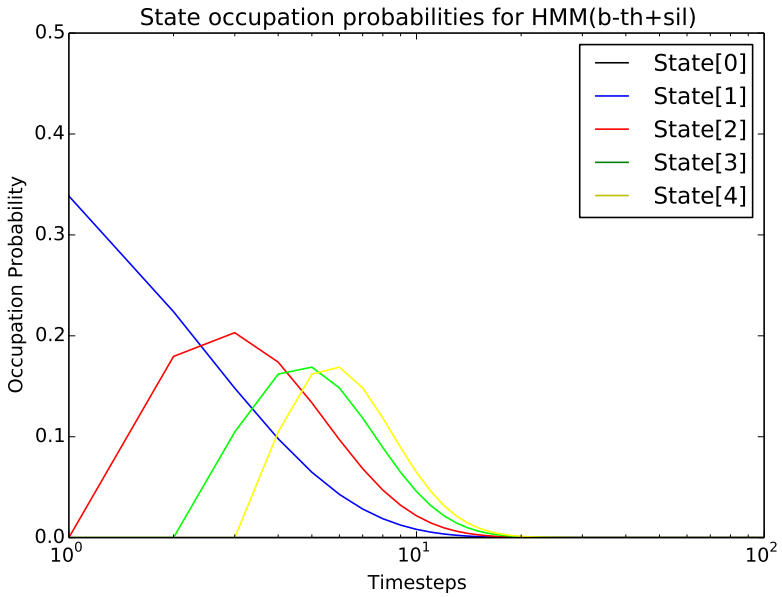


Figure 7.13: An example of state occupancy probabilities for each state in a 3-state left-to-right HMM. State[0] is the entry non-emitting state, which it is occupied only when $t = 0$. State[4] is the exit non-emitting state.

But not:

- Identity: $\text{HMM}_{\text{sim}}(x, x) = 1$

In order to satisfy the identity condition, each row in the similarity matrix is normalised by the value of $\text{HMM}_{\text{sim}}(x, x)$ of that row. To ensure the symmetry condition is satisfied, the resulting matrix should be symmetrised by the average of $\text{HMM}_{\text{sim}}(x, y)$ and $\text{HMM}_{\text{sim}}(y, x)$ as follows:

$$\text{HMM}_{\text{sim}}(x, y) = \text{HMM}_{\text{sim}}(y, x) = \frac{\text{HMM}_{\text{sim}}(x, y) + \text{HMM}_{\text{sim}}(y, x)}{2}$$

7.3.2 Agglomerative hierarchical clustering

A hierarchical clustering splits a given set of objects into clusters recursively by either a top-down (*divisive*) or bottom-up (*agglomerative*) fashion. In divisive hierarchical clustering, all objects are grouped into one cluster initially, then the cluster is divided based on the (dis)similarity metric into sub-clusters recursively until there is only one object per cluster. In contrast, agglomerative hierarchical clustering is initialised by assigning each object to its own cluster, then these clusters are merged based on the (dis)similarity metric recursively until there is only one cluster containing all objects. Both methods generate a hierarchical structure, known as a *dendrogram*, which is a tree-like representation where all objects are

leaves at the bottom and the inner nodes represent clusters while the similarity between two objects (or clusters) is proportional to the height of the first inner node joining them where all leaves are setting at the zero-level.

There are several hierarchical clustering algorithms, based on how the similarity between formed clusters is computed. Typical methods are *single*, *complete* and *average* linkage. If there are two clusters $A = \{a_1, a_2, ..., a_n\}$ and $B = \{b_1, b_2, ...b_m\}$ where $a_1, ..., a_n$ and $b_1, ..., b_m$ are member objects of these clusters respectively, single linkage considers the distance between A and B to be equal to the shortest distance between any pair (a_i, b_j) of their members (Sneath et al., 1973), whereas complete linkage considers the longest distance between any pair (a_i, b_j) to be equal to the distance between the clusters A and B (King, 1967). Average linkage considers the distance between any two clusters to be equal to the average distance from any member of one cluster to any member of the other (Ward Jr, 1963; Murtagh, 1983). Clusters based on single linkage suffer from chaining by which two clusters are merged if at least two points from each clusters are close to each other. Therefore, single linkage is not robust to noisy data. In contrast, clusters based on complete linkage are compact but they suffer from crowding where they are not far enough apart. Average linkage balances between single and complete linkage where its clusters are relatively far apart and compact. However, all these linkage methods are sensitive to noise, outliers and subsets of different size and densities. The distribution of context-dependent HMMs in a given set of transcriptions is not uniform and depends on the frequency of occurrences of these models in the data. Consequently, the overall frequency distributions of a given data has several densities with different sizes, according to their frequency in the data. Therefore, these linkage methods should be avoided.

The *graph degree linkage* method (Zhang et al., 2012) overcomes these problems. It is based on graph theory by which a weighted directed adjacency graph is built based on the (dis)similarity matrix. Using this graph, the similarity is computed between clusters based on the product of so called *indegree* and *outdegree* in the graph. Indegree of a node from a cluster measures the density near the node, while outdegree of a node to a cluster measures the similarity between a node and a cluster with respect to its K-neighbours. This algorithm outperforms other linkage methods in image clustering and object matching tasks, even handling manifold structure in high-dimensional space (Zhang et al., 2012).

To obtain the desired number of clusters, the dendrogram is cut at a fixed height where the number of internal nodes on the level directly below the cut height is equal to the number of desired clusters, with each internal node constituting a cluster. Figure 7.14 illustrates an example for a dendrogram from which different numbers of clusters can be obtained. In the example, the hierarchical structure for a set of eight objects $\{A, B, C, D, E, F, G\}$ is generated based on a (dis)similarity metric where the most similar

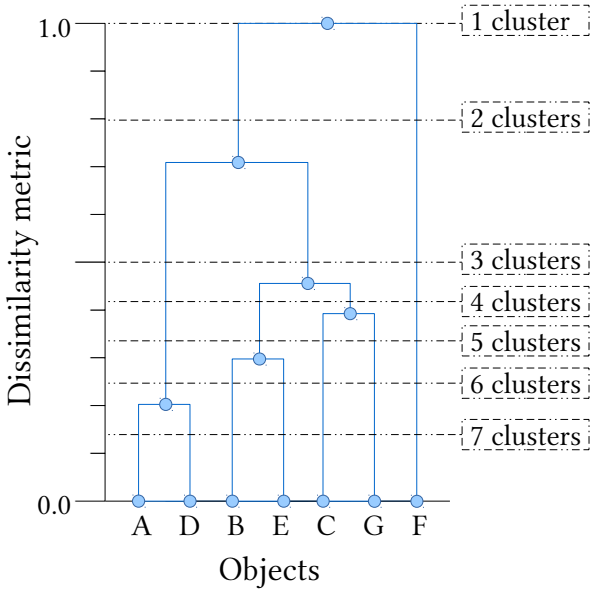


Figure 7.14: An example of a dendrogram and choice of clusters using the fixed height cut method. Different cut-off points define the number of obtained clusters which are the number of first internal nodes encountered below the cutt-off point.

nodes are connected together by an internal node. For example, (A,D) are clustered together as well as (B,E) and (C,G), then the latter two clusters are grouped together and so on. The closer the internal nodes are to the baseline, the more similar the clusters, i.e. (A,D) are more similar to each other than (B,E).

By defining a cut-off point, the number of clusters obtained is the number of first internal nodes below the cut-off point. Figure 7.14 shows several cut-off points along with the obtained number of clusters, e.g. three clusters for a cut-off point of 0.5. This method for extracting clusters is known as *fixed height cut* method because the cut-off is made uniformly at a certain level.

For the purpose of clustering context-dependent HMMs, the main objective is to decrease the ambiguity and the possibility of creating homophone words. This means that phonemic sounds should not be merged even if they are similar to each other acoustically. If the clustering method generates clusters that violate such constraint, an alternative strategy to fixed height cut should be employed for defining the cut-off points in the dendeogram while retaining the desired number of clusters.

7.3.3 Restructuring the CA acoustic inventory

Using the proposed method to cluster a context-dependent acoustic model set, the model space hierarchical structure can be revealed and the models can be grouped based on the

similarity between the models into X clusters, while phonemic segregation is retained. Each cluster then represents a new model. These X models are used to substitute all the context-dependent models in the training transcription to be used to train the new clustered model set, cX, where X is the number of clusters obtained. After that, the newly transformed transcriptions are used to train context-independent cX models and state-tied context-dependent cX models by applying Expectation-Maximisation (EM) algorithm.

7.3.4 State tying for the restructured models

Generally, context-dependent acoustic models encode the impact of neighbouring phonemes on the acoustic realisation of a given phoneme, known as coarticulation. Each context-dependent HMM models the centre phoneme in a sequence of three (triphone) or five phonemes (quentphone). Modelling all possible contexts results in a large number of models and thus parameters which are not all observed in the training data. To counter this, clustering is used to tie these parameters together. All states within the same cluster are tied, thus reducing the total number of parameters to allow more robust estimation. This clustering is customarily based on the context's phonetic classes using some form of parameter tying scheme such as phonetic decision trees (Odell and Woodland, 1994). A phonetic decision tree (Breiman et al., 1984) is a binary tree which splits the HMM states by recursively answering a logical question about the immediate phonemic context. An example of such questions is "Is a nasal on the left of the current phoneme?". These questions are generated either manually, based on linguistic knowledge of the mapping between the acoustic units and their phonetic class (Young and Woodland, 1994), or automatically, based on a bottom-up clustering of context-independent HMM model states (Beulen and Ney, 1998).

In order to generate such questions for tying the parameters of cX model set, either the mapping between each unit and a phonetic class should be established or states should be clustered in an agglomerative fashion. Before restructuring the model space into the cX model set based on the HMM similarity, each unit in the *phn* acoustic set is mapped to a phonetic class. However, after extracting the new models from the hierarchical structure of the original model space, this mapping is no longer clear which dismisses the first strategy in generating the state-tying questions. Therefore, the latter strategy is taken and an automatic process is introduced based on the CSD (discussed in Section 7.3.1) between the states of the cX model set.

Similar to the method of Beulen and Ney (1998), an agglomerative clustering can be applied on the states of context-independent cX model set. Beside the dissimilarity matrix between the states, which can be computed based on CSD, the position of the state can also be used. This means that, for example, only the first states in HMMs are considered

Algorithm 7.1 Generate K questions based on dissimilarity matrix D and state position information

```
1:  procedure GenerateQuestions(D, H, K)
2:      N ← K      ▷ N number of clusters extracted by graph degree linkage hierarchical clustering
3:      PreviousCount ← 0
4:      Update ← N/2
5:      Q ← {}                                                    ▷ Q set of question clusters
6:      while |Q| ≠ K  &  Update/2 ≠ 0 do    ▷ |Q| number of questions extracted so far
7:          A ← GDLHC(D, N)              ▷ Use graph degree linkage hierarchical clustering
8:          Q ← ClusterByStatePosition(A, H)
9:                                  ▷ Further split found clusters based on state position H
10:         if |Q| > K then
11:             if PreviousCount ≤ 0 then
12:                 Update ← Update/2
13:             N ← N − Update
14:             PreviousCount ← 1
15:         else
16:             if PreviousCount ≥ 0 then
17:                 Update ← Update/2
18:             N ← N + Update
19:             PreviousCount ← −1
20:     return Q
```

for clustering together and only second states are clustered together and so on. In order to consider both features, states dissimilarity and position, in generating the questions set, the clustering process can be performed in two steps. At first, a hierarchical structure is generated for these states based on a dissimilarity matrix, based on CSD. This is followed by further splitting of each cluster into sub-clusters based on the position of the states in a given HMM. The number of resulting clusters is equivalent to the number of questions to be generated. All units belonging to a cluster represent affirmative answers to the questions. For example, if the models {m1, m32, m40} belong to the cluster clst_1, then the answer to the questions "Is a clst_1 model in to the right?" will be YES if and only if the model on the right is either m1, m32 or m40; otherwise, the answer will be NO. These steps are summarised in iterative algorithm shown in Algorithm 7.1 to extract K questions using state position information H and state dissimilarity matrix D.

In a phonetic decision tree (Breiman et al., 1984), the question at each node is chosen to improve the likelihood of the clustered states while there is sufficient[7] amount of training data for robust estimation. Therefore, having a large pool of questions to choose the most effective questions to build the decision tree is preferable. By using the method described

[7]By defining threshold to indicate a minimum accepted amount of training data.

above for generating the questions, the maximum number of questions is the total number of units because each unit belongs to only one cluster. To extend the number of generated questions, soft clustering can be employed, where a unit can be assigned to one or more clusters. Alternatively, clustering the states of the context-dependent *cX* model set is used instead of the states of the context-independent *cX* model set. As a result, a unit is associated with every cluster (or question) if it is the centre phone for a model in that cluster. For example, for a given three context-independent models, {q, e, w}, the following clusters were extracted:

$$\begin{aligned} \text{clst}_1 &= \{ \text{ q, e } \} \\ \text{clst}_2 &= \{ \text{ w } \} \end{aligned}$$

This means the questions for each context-independent model is:

$$\begin{aligned} \text{q} &: \text{clst}_1 \\ \text{e} &: \text{clst}_1 \\ \text{w} &: \text{clst}_2 \end{aligned}$$

Here, each model is having only one question assigned to it. However, when considering the subset of context-dependent models {w-e+q, w-e+w, w-q+e, w-q+w, q-e+w, q-w+e}, the following clusters were extracted:

$$\begin{aligned} \text{clst}_1 &= \{ \text{ w-e+w, w-q+e, w-q+w } \} \\ \text{clst}_2 &= \{ \text{ w-e+q, q-e+w, q-w+e } \} \end{aligned}$$

By considering only the centre phone, the following clustering will be assigned instead after removing all repeated units:

$$\begin{aligned} \text{clst}_1 &= \{ \text{ e, q } \} \\ \text{clst}_2 &= \{ \text{ e, w } \} \end{aligned}$$

As a results, more questions can be assigned to the same model which can be considered as soft clustering, such that:

$$\begin{aligned} \text{q} &: \text{clst}_1 \\ \text{e} &: \text{clst}_1, \text{clst}_2 \\ \text{w} &: \text{clst}_2 \end{aligned}$$

7.3.5 Results and discussion

This section describes several experiments that were conducted to assess the *cX* acoustic set. First, several *cX* with different number of clusters are compared in terms of their phone recognition performance and quantity of learning within the models. Second, an analysis

of the change in the assignment of the original acoustic set and the new *cX* clusters is presented. Third, several state-tying question generation strategies are compared, in terms of their performance in a phone recognition task using an undiacritised grapheme acoustic set.

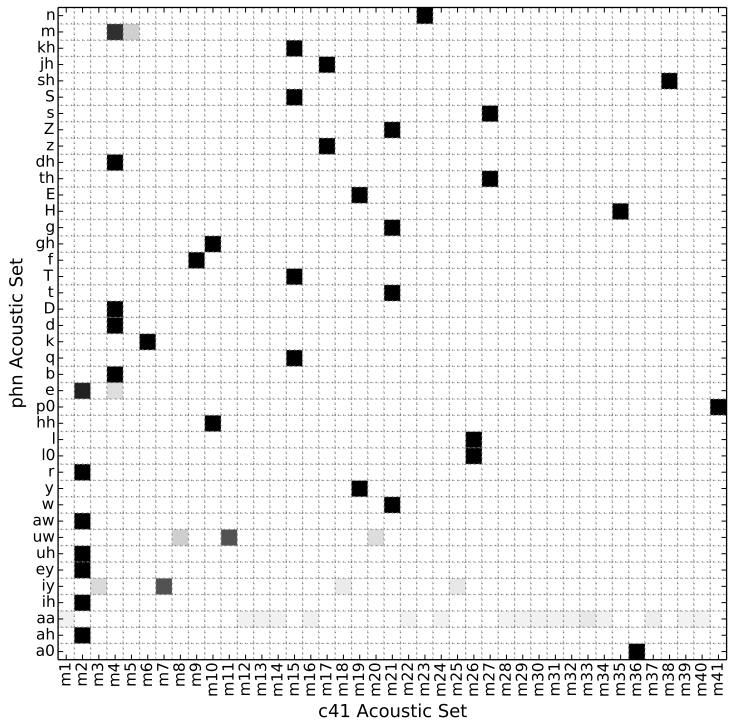
Hierarchical structure and number of clusters

Using the acoustic models as described in Section 7.2.5, an HMM similarity matrix was computed on the context-dependent level. Based on that similarity matrix, hierarchical clustering using graph degree linkage was performed where four cluster sets are extracted from the hierarchical structure using fixed height cut. These sets are *c41*, *c60*, *c80* and *c100* containing 41, 60, 80 and 100 units respectively. Transcriptions are transformed from *phn* to *cX* by substituting each context-dependent *phn* model with its corresponding *cX* unit. If two or more consecutive context-dependent *phn* units belonged to the same cluster, only one instance of the corresponding *cX* unit are used to substitute these *phn* units. As a results, the total number of units in the training transcriptions after transforming them from *phn* acoustic models to their equivalent *cX* acoustic models decreased in relation to the chosen number of clusters; such that it decreased by 5.4% relative when *c41* is chosen whereas this ratio decreased to be less than 0.06% for *c60* units. This suggests that a smaller number of clusters will merge more adjacent sounds and model longer durations while a greater number of clusters will have more separations than merges as shown in Figures 7.15 and 7.16 which illustrate the assignment of phonemes in the *phn* acoustic set to the extracted clusters. In these figures, the darker an assignment is, the more triphones, were clustered together with the same centre phone as the corresponding phoneme. For example, all triphones with centre phoneme "m" are clustered into two clusters: {m4, m5} when 41 clusters were used (Figure 7.15a), with more triphones in the cluster m4. But when 80 clusters were used, it was clustered into three clusters: {m15, m64, m65} (Figure 7.16), with more triphones in the cluster m15.

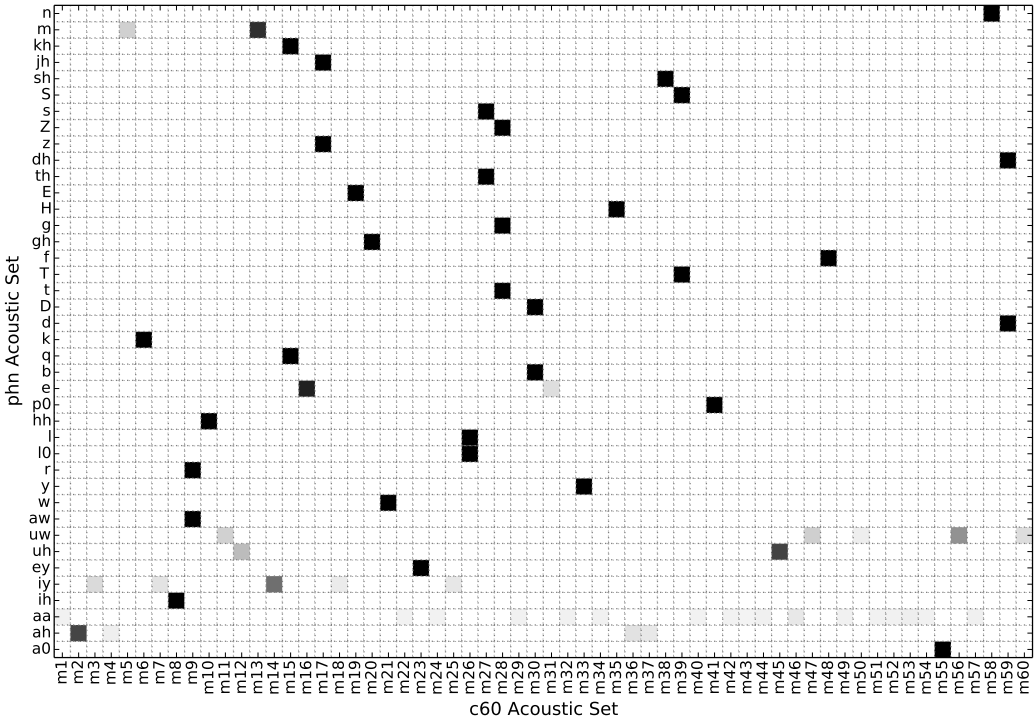
These four *cX* model sets were evaluated in a phone recognition task, with a bigram PLM. Results in terms of UER cannot be compared across systems due to the difference in the number of units across these four sets. Instead, two metrics based on information theory are employed, in addition to UER: the normalised information transfer (NIT) factor and entropy-modulated accuracy (EMA) (Valverde-Albacete and Peláez-Moreno, 2014). These are intended to quantify how much a system learnt during the training. EMA and NIT are computed to evaluate a system of input set x and output set y as follows:

$$\text{EMA} = 2^{-H_{x|y}} \tag{7.24}$$

$$\text{NIT} = 2^{MI_{xy} - \log_2(k)}, \tag{7.25}$$



(a) c41



(b) c60

Figure 7.15: Assignment of units in the *phn* acoustic set to the *cX* acoustic set when using a different number of clusters: (a) 41 clusters, (b) 60 clusters. The darker a cell, the more triphones occupy it

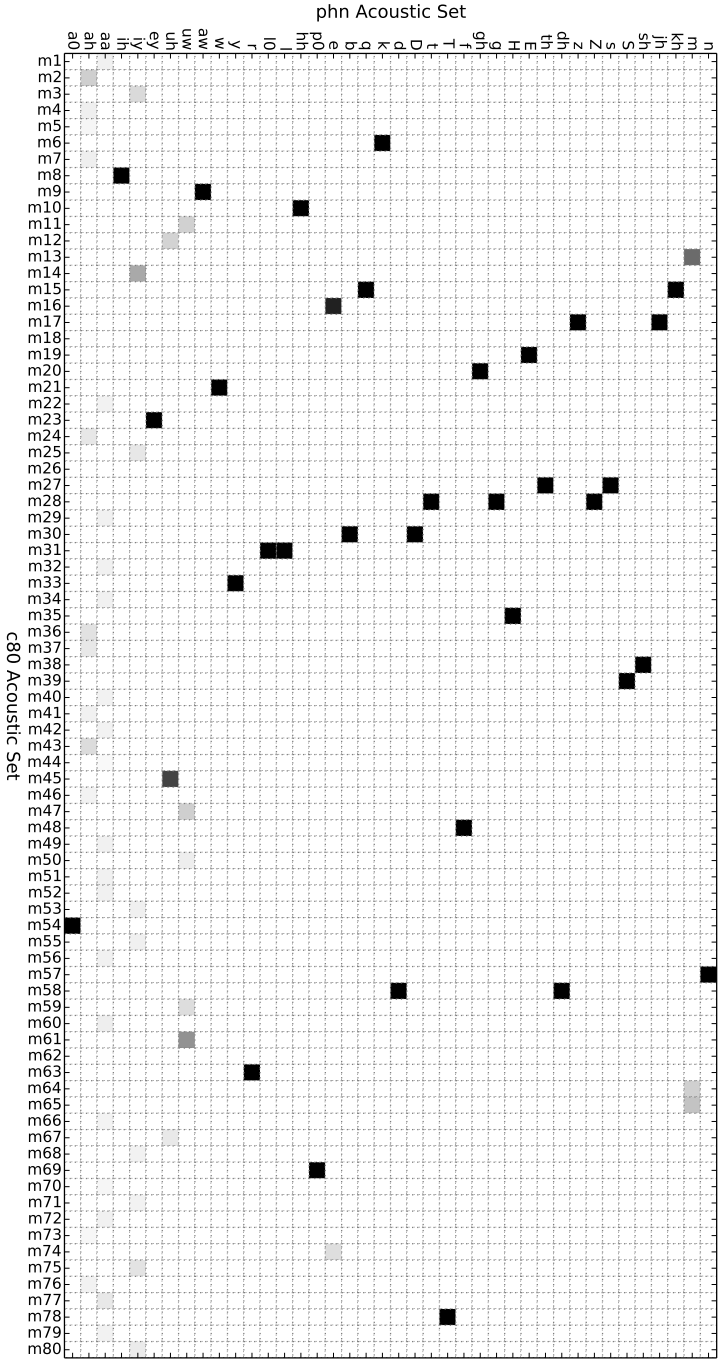


Figure 7.16: Assignment of units in the *phn* acoustic set to the *cX* acoustic set when using 80 clusters. The darker a cell, the more triphones occupy it.

where k is the number of number of classes, and $H_{x|y}$ and MI_{xy} are the conditional entropy and the expected mutual information respectively. These which are computed as follows:

$$H_{x|y} = H_x - MI_{xy} \tag{7.26}$$

$$MI_{xy} = H_x + H_y - H_{xy} \tag{7.27}$$

where H_x, H_y and H_{xy} are the input entropy, output entropy and the joint entropy. Generally the entropy is computed as:

$$H = -\sum_i P^{(i)} \log_2 P^{(i)}, \tag{7.28}$$

where $P^{(i)}$ is the probability distribution of the unit i, which can be computed from a confusion matrix as follows:

$$P_{xy}^{(i,j)} = \frac{c_{ij}}{\sum_k \sum_l c_{kl}} \tag{7.29}$$

$$P_x^{(i)} = \frac{\sum_j c_{ij}}{\sum_k \sum_l c_{kl}} \tag{7.30}$$

$$P_y^{(i)} = \frac{\sum_j c_{ji}}{\sum_k \sum_l c_{kl}} \tag{7.31}$$

where $P_{xy}^{(i,j)}$ is the joint probability which is computed from the number of times that unit i was recognised as unit j, this amount is c_{ij}. $P_x^{(i)}$ is the input probability for the unit i and $P_y^{(i)}$ is the output probability for the symbol i. The term $\sum_k \sum_l c_{kl}$ represents the total number of units in the test set.

Table 7.7 compares the performance when using these sets. There are two types of confusability observed in ASR systems: acoustic confusability is observed when two models compete over the frame assignment, while the pronunciation confusability is when more than one words share the same pronunciation in the dictionary. Since the current evaluation was a phone recognition task, only the former could be observed. It is apparent that increasing the number of clusters improves the performance and the learning process until the performance started to degrade as the number of clusters increased. The increased number of substitutions suggests that the degradation might be because these clusters are not far enough apart to be acoustically distinguishable. However, there is also a high substitution rate found in the acoustic set with the lowest number of clusters, *c41*, by comparing the assignment between the clusters (shown in Figure 7.15), there is more

Table 7.7: Results for phone recognition task using bigram PLM on AppenLCA testing set across several acoustic sets.

	AM	#units	NIT	EMA	UER	FER
graphemic	graph	33	0.0853	0.1605	56.6	46.5
	crfdiac	36	0.0687	0.1621	54.3	49.9
	mandiac	36	0.0736	0.1676	53.3	48.0
derived	phn	41	0.0740	0.1765	52.5	48.3
restructured	c41	41	0.0575	0.1331	57.3	52.7
	c60	60	0.0546	0.1007	59.1	56.0
	c80	80	0.0494	0.0862	61.2	56.9
	c100	100	0.0412	0.0760	61.7	59.2

merging of phonemes in *c41* than in *c60*. For example, the phonemes {m, dh, D, d, b, e} where merged in one clusters in *c41* but the same set was clustered into four clusters instead in *c60* as {m}, {dh, d}, {D, b} and {e}. Such merging was due to the smaller number of clusters. Based on the number of units to be recognised, *mandiac* with *crfdiac* and *phn* with *c41* are comparable in terms of their UER while the rest of acoustic sets are not due to the difference in the number of units. Phone recognition using the *phn* set outperformed using *c41* which is caused from merging some acoustically similar but phonemic units into one which increased the number of substitutions.

Generating state-tying questions

In order to assess whether the proposed method for generating questions is beneficial, the undiacritised grapheme acoustic set was chosen for this experiment. This was tested in a phone recognition task with bigram PLM where the UER is employed as an evaluation metric.

The state-tying questions were generated using four methods. Two of the question generation methods are based on some information either from prior knowledge about the mapping between the units and their phonetic classes, or derived from the similarity between the states using the hierarchical clustering. The other two methods do not depend on any knowledge but the identity of the units themselves.

For the *phonetic classes* method, each grapheme is represented by one nominated phoneme (even if the grapheme is a multi-phoneme) based on the phonological knowledge. The questions are related to the phonetic class of that nominated phoneme. For example, the grapheme "v" has the phoneme /θ/ as its nominated phoneme and the chosen questions are phonological questions about the phoneme /θ/. Such a question can be "is Unvoiced-Fricative on the left?" When assuming that such assignment to a phonetic class is not clear or unavailable, the grapheme itself is considered as a question, hence, the

Table 7.8: Comparison of different strategies in generating state-tying decision trees for undiacritised grapheme acoustic models in terms of UER and the relative difference (% difference) to the basline UER.

Questions		UER	% difference
Strategy	Count		
identity	34	58.42	baseline
phonetic classes	59	56.59	-3.1
random	50	57.99	-0.7
	100	58.29	-0.2
	200	58.05	-0.6
	500	58.40	0.0
	1000	58.39	-0.1
	5000	58.57	0.3
hierarchical clustering	59	57.98	-0.7

identity based questions. For example, "is v on the left?" . In addition, to the proposed method in generating question through finding the *hierarchical clustering* and extracting questions based on the similarity between the states. *random* generation of questions is also tested.

Table 7.8 shows the UER when tying the states of undiacritised graphemes into 1000 shared states and the same clustering configurations[8]. The number of questions would differ according to the generation method; however the actual number of questions will be doubled because a question will be combined with the immediate context. For example, in the identity question of grapheme "H", there will be two questions: "Is H on the right?" and "Is H on the left?".

As expected, providing knowledge about mapping between acoustic units (in this case undiacriticsed graphemes) and the phonetic classes shows the best phone recognition results. Interestingly, generating random questions provided some improvement at generating 50 questions which suggested that increasing the number of questions would enhance the performance; however, no further improvement was gained. In addition, the same experiments were replicated for the random question generation, but as the method implies, the occurrence of performance gain was random as well, unlike the hierarchical clustering method which always provides the same set of questions. The number of questions used in hierarchical clustering was set to be equal to the number of questions based on phonetic classes; however, from the observation of the similarity matrix (shown in Figure 7.17), increasing the number of questions will result in better clustering. This makes using hierarchical clustering for generating state-tying questions a valid alternative in the absence

[8]Clustering configurations here are tree-branching (TB) and removing outliers (RO) which are set to 500 and 1000 respectively.

of the phonological knowledge, even when the number of clusters as small as 59.

7.3.6 Pronunciation generation

As stated in Section 7.3.3, the original mapping between the units and their phonetic classes is not clear after extracting the *cX* acoustic sets from the hierarchical structure of the *phn* acoustic set. Consequently, the relationship between the word and its found pronunciation in *phn* units, which was based on the defined heuristics in Section 7.2, no longer exists. Therefore, the grapheme-to-phoneme (G2P) mapping should be learned in order to generate pronunciations for unseen words in the acoustic training set.

G2P conversion has been discussed previously in Section 6.3, but in the context of diacritisation. Here, G2P conversion is used for its main purpose. As training materials for the G2P model, pronunciations were extracted from the *cX* transcriptions where each training sample is a pair of a word in a graphemic form and its pronunciation using *cX* units.

Since the transcription transformation from *phn* units to *cX* units was made based on context-dependent units, a baseform might have several pronunciations where they only differ in the initial and last units based on the context. For example, the word "$w" (what) is pronounced as "sh uw" using the *phn* acoustic models and represented as "XX-sh+uw sh-uw+YY" using context-dependent *phn* units where XX and YY can be any units from the *phn* acoustic set; however, all the models representing XX-sh+uw are clustered together in the *c60* acoustic set, within a cluster number 38, unlike those of sh-uw+YY where they are divided into two clusters, cluster number 50 and 60. Consequently, the single pronunciation "sh uw" will be transformed into two pronunciations using the *c60* acoustic set: "m38 m50" and "m38 m60"[9]. This increases the number of pronunciations per word in the extracted dictionary considerably, and increasing as the number of clusters increases. As a consequence, the ambiguity in the G2P conversion model increases and degrades its performance.

Two methods can be used in order to reduce the number of pronunciations extracted per word. First, an alignment after (at least) one round of acoustic model re-estimation will reduce the number of pronunciations found. Alternatively, pronunciation probabilities, computed based on the frequency within the training data, can be used as a filtering method by which pronunciations having probabilities which are lower than a chosen threshold are removed from the dictionary.

[9] A model in *cX* acoustic set is named using the cluster number preceded by the letter "m".

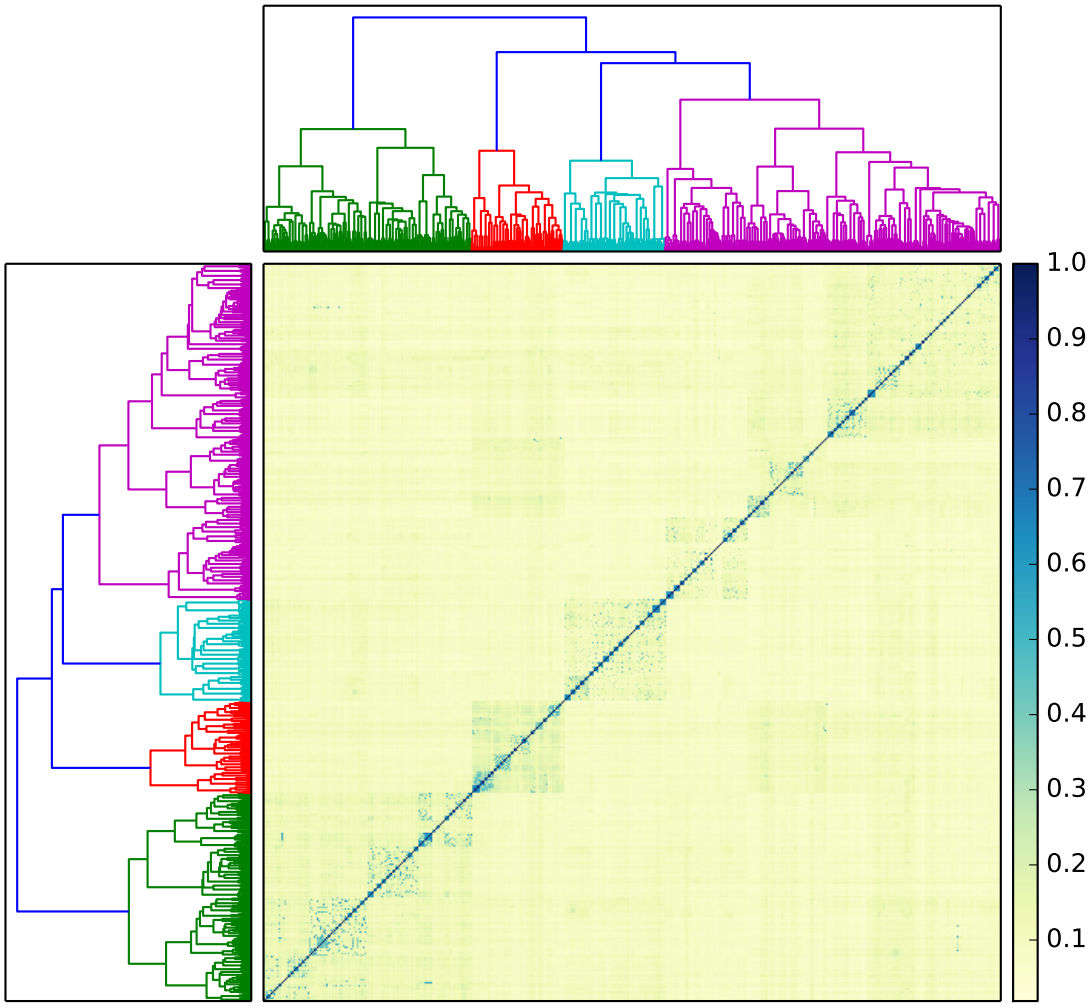


Figure 7.17: Similarity matrix based on the Cauchy-Schwarz divergence between 1038 states of undiacriticsed grapheme set. These states were estimated from 9 hours of AppendLCA training.

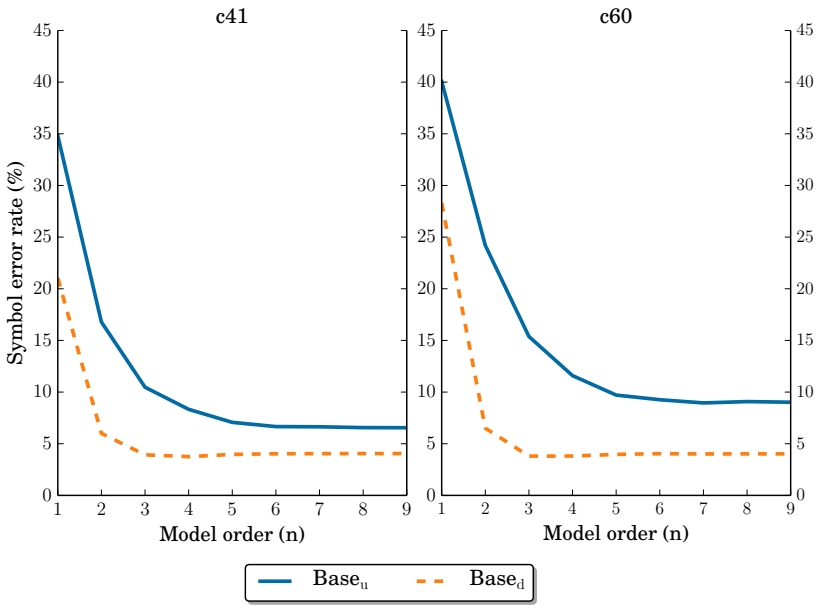


Figure 7.18: Symbol error rate (y-axis) for G2P converters based on dictionaries in *cX* models when using different depth of context (x-axis) without any pronunciation reduction strategies such as filtering or force-alignment. Two dictionaries where used for each system: words in undiacritised form Base$_u$ (solid line) or in diacritised form Base$_d$ (dashed line).

7.3.6.1 Results and discussion

In order to assess the found mapping between the written form and the *cX*-based pronunciations, training materials were extracted from the 9-hour AppenLCA training set previously chosen (detailed in Appendix B). The chosen numbers of clusters to be tested were 41 and 60.

As training material, pronunciations were extracted from aligning transformed transcriptions in the *cX* acoustic set with the associated word-level transcriptions. All word fragments and disfluency markers were excluded from the transcriptions beforehand. Two dictionaries were created from extracted pronunciations based on the written form for the word entries that either included diacritics, Base$_d$, or exclude them, Base$_u$. The number of pronunciations per word is relative to the number of clusters, as discussed previously. Thus, when using *c60* acoustic units, the average of pronunciations per word for Base$_d$ and Base$_u$ are 1.3 and 1.7 respectively, and 1.3 and 1.5 when using *c41* acoustic set. A held out set of randomly chosen 5% of the original word list in Base$_d$ and Base$_u$ was chosen from all resulted dictionaries for evaluation.

Based on each dictionary, nine G2P converters were trained with context ranges from unigram to 9-gram using the Sequintur G2P toolkit (Bisani and Ney, 2008) with similar settings as used previously in Section 6.3.3. Figure 7.18 shows the the performance of

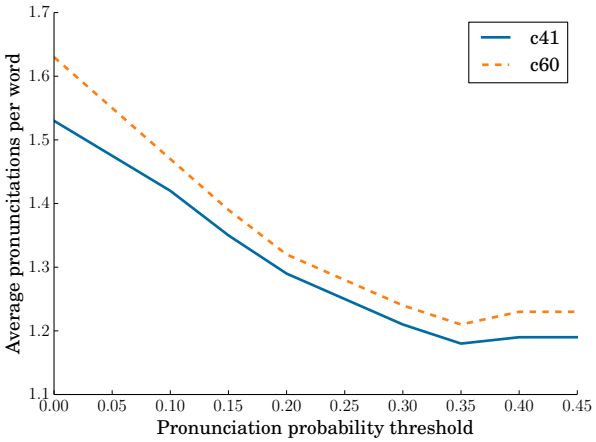


Figure 7.19: Pronunciations per word (y-axis) for Base_u dictionaries in both *c41* (solid line) and *60* (dashed line) acoustic units as the pronunciation probability threshold increased (x-axis).

Table 7.9: The relative reduction in the number of pronunctions (Reduction%) and the pronunciations per word (Prons/word) when applying pronunciation reduction strategies.

	c41			c60		
		Prons/word			Prons/word	
	Reduction%	Base_u	Base_d	Reduction%	Base_u	Base_d
before filtering	*baseline*	1.5	1.3	*baseline*	1.7	1.3
force-aligned	-4.2%	1.5	1.2	-4.1%	1.6	1.2
filtered (0.35<pp)	-11.3%	1.4	1.1	-12.0%	1.4	1.1
filtered (best pp)	-17.2%	1.3	1	-18.0%	1.3	1

these G2P converters in terms of symbol error rate (SER). SER is the Levenshtein distance divided by the number of phonemes in the reference pronunciation. Generally, performance improved as the context depth increased up to 5-gram where no further improvement was prominent. It was evident that using words in diacritised form in the training dictionary, (Base_d), resulted in a better G2P models, especially when the number of clusters increased. There was an average of 40.8% relative SER improvement in the *c41* acoustic set over G2P models trained the using dictionaries with undiacriticsed words, Base_u. This improvement reached an average of 58.5% relative SER in the *c60* acoustic set. When using more clusters in the acoustic set, the G2P performance degrades for the undiarcritised dictionary, Base_u, by an average of 26.7% relative SER. Such a difference was not observed in Base_d owning to the fact that most of the additional clusters were in the vowel units (as shown in Figure 7.15) where these vowels are not represented in undiacritised words and consequently the mapping between the two representations becomes more ambiguous.

To evaluate which of the pronunciation reduction strategies is the most effective, the average number of pronunciations per word of the resulting dictionaries are compared. First,

diacritised word based transcriptions was force-aligned using the *c41* and *c60* acoustic sets and their corresponding dictionaries. The resulting dictionaries only kept pronunciations used in the force-alignment, and pronunciations probabilities were computed based on the frequency in the aligned transcriptions. The force-alignment strategy on its own reduced the average number of pronunciations per word by a 0.1 absolute for both *cX* acoustic sets using Base_d dictionaries (as shown in Table 7.9). Further reduction was obtained when setting a threshold on the pronunciation probabilities (as shown in Figure 7.19). This decreased the average number of pronunciations per word until it reached a slight increase in the average number of pronunciations per word beyond the pronunciation probability of 0.35, which was then chosen as a cut-off point. Additional dictionaries were extracted by including only the most frequent pronunciation for each diacritised word within the training data, i.e. with the highest pronunciation probability. Table 7.9 summarises the average number of pronunciations per word for different dictionaries using several pronunciation reduction strategies.

Based on these new dictionaries, another set of G2P converters were trained. By considering the previously trained G2P converters as baseline systems, Figure 7.20 illustrates the relative difference in SER for each system. The observed increase in SER might be due to the reduction in training examples (relative reduction in dictionaries is shown in Table 7.9), especially when converting diacritised words (Base_d). However, the SER improved with adding more context generally.

7.4 CTS ASR Experiments

The main objective of this chapter is to redefine the acoustic inventory for CA to improve the recognition performance for ASR tasks. In order to evaluate the impact of the proposed acoustic sets on recognition performance, an LCA CTS ASR task based on AppenLCA corpus was performed where the system configuration and resources were the same as those used previously for evaluation in Section 6.6.

For all systems in the following experiments, the same language model was employed where the recognition hypothesis is in undiacritised form. Based on each acoustic set, a recognition dictionary was implemented as described previously.

For completeness, undiacritised graphemes (*graph*) results are also included in this section. Table 7.10 compares the performance in terms of WER across systems. An absolute improvement of 1.1-1.0% in recognition performance was observed when using the *phn* acoustic set over diacritised graphemes, with fewer deletion errors compared to using either of the graphemeic-based acoustic sets. In both cases, this improvement shows to be statistically significant with $p < 0.001$ using MAPSSWE. However, using the *c41*

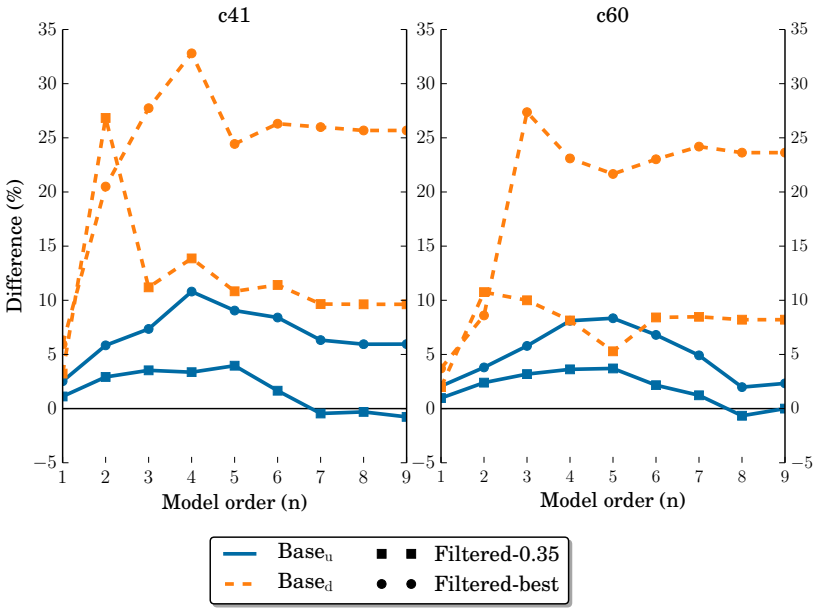


Figure 7.20: The change in the G2P performance (y-axis) using different depth of context, i.e. model order, (x-axis) when training on filtered dictionaries, either by removing pronunciations with a probability lower than 0.35 (square markers) or by just selecting the pronunciation with highest probability (round markers). Pronunciation probabilities are computed based on frequency in the training transcriptions. Filtering the dictionary before training with a pronunciation probability, computed based on the frequency in the training transcriptions, cut-off at 0.35 or by select one best pronunciation for the training dictionary. Baseline systems are presented in Figure 7.18.

Table 7.10: Recognition performance for CTS ASR experiments using the AppenLCA corpus using different sets of acoustic sets: Graphemic sets: undiacritised graphemes (*graph*) and diacritised graphemes, manually diacritised (*mandiac*) and automatically diacritised (*crfdiac*), phonemic set (*phn*) and acoustically clustered sets (*c41* and *c60*).

AM	WER	Substitutions	Deletions	Insertions
graph	71.4	48.5	20.0	2.7
crfdiac	70.2	47.4	20.9	1.9
mandiac	70.3	48.4	18.4	3.5
phn	69.2	52.0	13.6	3.6
c41	75.0	58.2	14.2	2.6
c60	70.0	54.0	12.7	3.4

models resulted in higher WER due to increase of substitution errors. This confirms that using a small number of clusters resulted in merging acoustically similar but phonemically different units. Increasing the number of clusters in *c60* acoustic set slightly outperformed the diacritised graphemes set by 0.3-0.2% absolute WER, which shows to be statistically significant with $p < 0.001$ using MAPSSWE.

No pronunciation reduction was applied during the training of the G2P converters of the *cX* acoustic sets in the listed results in Table 7.10. As described in Section 7.3.6, the resulting dictionaries for *cX* systems suffer from high numbers of average pronunciations per word which were dealt with by using force-alignment and cut-off pronunciation probabilities based on the frequency within the training data. Consequently, two additional dictionaries were generated for each set, for both training and testing where their baseform entries were the undiacritised representations. Tables 7.11a and 7.11b show recognition performance when using different combinations of these dictionaries for the *c41* and *c60* acoustic sets respectively. Generally, the best performance for both acoustic sets was accomplished when using both training and recognition dictionaries that were generated from a G2P converter that was trained using a dictionary that only kept the pronunciation with the highest frequency in the training transcriptions. The absolute reduction was 0.1% WER for the *c41* acoustic set, while it reached 0.6% absolute WER for the *c60* acoustic set, where the latter outperformed the baseline diacritised grapheme system by 0.8-0.9% absolute WER. Again, for both *cX* acoustic sets, the worst performance was obtained when training dictionaries were generated using G2P converters trained on unreduced dictionaries while the recognition dictionaries were generated using G2P converters trained on dictionaries that only kept the most frequent pronunciations.

A more realistic evaluation scenario is when no diacritisation is provided such that all derivation procedures for the new acoustic set will be based on automatically diacritised transcriptions as a starting point. To evaluate such a scenario, a system based on undi-

Table 7.11: Recognition performance in WER for CTS ASR experiments using AppenLCA with several combination of training and testing dictionaries using either (a) the *c41* or (b) the *c60* acoustic sets. Dictionaries can be either unreducted (all) or reduced in terms of number of pronunciations by removing all pronunciations where their pronunciation probabilities are less than 0.35 (filtered-0.35) or by only including the pronunciation with the highest pronunciation probability (filtered-best).

(a) *c41*.

Training \ Testing	all	filtered-0.35	filtered-best
all	75.0	75.9	76.3
filtered-0.35	75.3	75.6	75.8
filtered-best	75.6	75.5	74.9

(b) *c60*

Training \ Testing	all	filtered-0.35	filtered-best
all	70.0	70.7	71.0
filtered-0.35	69.9	70.1	70.0
filtered-best	69.4	69.7	69.4

Table 7.12: Recognition performance for CTS ASR experiments using the FisherLCA corpus with different acoustic sets: Graphemic sets: undiacritised graphemes (*graph*) automatically diacritised graphemes (*crfdiac*), phonemic set (*phn*) and acoustically clustered sets (*c41*, *c60*, *c80* and *c100*).

AM	WER	Substitutions	Deletions	Insertions
graph	60.4	45.0	11.3	4.1
crfdiac	60.1	44.9	11.2	4.0
phn	60.3	44.0	13.4	2.9
c41	62.4	46.2	12.8	3.4
c60	61.9	45.4	13.5	3.0
c80	62.5	45.8	13.8	3.0
c100	63.1	45.8	14.4	2.9

acritised resource (FisherLCA, described in Appendix B), was trained. Table 7.12 lists the evaluation results. A marginal degradation in the performance was observed when deriving the *phn* from an automatically diacritised transcription of 0.2% WER absolute. This was resulted from the strong reliance of the derivation procedure for the *phn* acoustic set on the existed diacritisation. Again, as restructuring the context-dependent *phn* model space into the *cX* accompanied by dictionaries generated from a G2P model trained on all extacted pronunciations did not yield any improvement but the observed degradation ration (of 3.5% WER relative) was smaller than that observed previously in the AppenLCA system (of 8.4% WER relative). That may be caused by the nature of the data sets since AppenLCA set contains more acoustic noise than FisherLCA, consequently, the computed HMM similarity will be based on better estimated models. Recognition performance was improved from increasing the number of cluster for the *cX* acoustic set to be 60 instead of 41; however, a further increase of number of clusters resulted in a steady degradation in the recognition performance which could be caused by the lack of training examples.

7.5 Summary and conclusion

This chapter addressed questions raised by Objective 4 of this thesis by investigating the possibility of defining a wider acoustic set than diacritised graphemes and defined mapping between the derived acoustic sets and the original written form to generalise the pronunciation generation process.

Using diacritised graphemes as acoustic units was not acoustically representative for CA in ASR tasks for several reasons. These include several infrequent acoustic units sharing the same phonetic realisation while some units have multiple distinctive phonetic realisation where the chosen value can be decided based on the context. For some of these multi-phoneme units, context does not contribute to choosing the acoustic value. Each of these issues was addressed through the course of this chapter. Two acoustic sets were proposed: *phn* and *cX*.

The *phn* acoustic set was the result of applying a sequence of phonological rules followed by disambiguating context-dependent multi-phoneme units using the acoustic context by means of force-alignment. Potentially silent graphemes were modelled using an HMM with a special topology that contains a skip, prior to the force-alignment process. This acoustic set contained 42 units and showed a more balanced frequency distributions on the training data, in comparison to the skewed frequency distribution observed for diacritised graphemes.

Second, context-independent multi-phoneme units were addressed by restructuring the context-dependent *phn* model space. Based on the computed similarity between the

context-dependent units, a hierarchical structure was constructed for the *phn* model space using graph degree linkage hierarchical clustering. A *cX* acoustic set can be extracted from the resulting hierarchical structure that contains X units where X corresponds to the number of extracted clusters. It was found that using lower numbers of clusters resulted in the merging of phonemic units into one, which increased the number of homophone words and consequently increased the number of substitution errors. Using larger numbers of clusters can lead to under-trained units due to the lack of examples in the training set. Therefore, the number of clusters must be optimised.

For both proposed acoustic sets, a mapping was found from diacritised graphemes, whether the diacritisation was manual or automatic. The mapping to the *phn* acoustic set was by applying phonological rules owing to the less ambiguous relationship between each grapheme and its corresponding acoustic unit. In the *cX* acoustic unit case, such direct mappings did not exist; therefore, a G2P conversion method using joint-sequence statistical modelling was employed which showed consistent results with high accuracy up to 96% when the source was in diacritised form, and up to 94% when source baseform was undiacritised. This accuracy dropped as the chosen number of clusters increases.

The empirical results of an LCA CTS ASR task showed that using the *phn* acoustic set outperformed diacritised graphemes by at least 1% absolute WER when the derivation was based on manually diacritistion. This improvement was found to be statistically significant with $p < 0.001$ based on MAPSSWE. Using *cX* with a small number of clusters degraded the recognition performance due to the increase in substitution errors which indicates that some of the clustered units merge phonemic units and increase the possibility of homophone words whereas using larger number of clusters shows statistically significant improved performance ($p < 0.001$) over diacritised graphemes but did not outperform a system using *phn* acoustic set. The performance was slightly reduced if the derivation of the proposed acoustic inventories was based on noisy diacritisation.

These results confirmed that using the proposed *phn* acoustic set or *cX* (with higher number of clusters) is acoustically closer to the acoustic realisation than diacritised graphemes, hence, systems using them obtained better recognition performance in ASR tasks. More optimisation methods are required for extracting *cX* acoustic set from the hierarchical structure in order to obtain further improvement.

Chapter 8

Conclusions and future work

Contents

The main goal of this research was to investigate and highlight the main issues for the observed under-performance of CA ASR systems compared to MSA when using conventional ASR modelling technologies. It has to be emphasised that all issues discussed in this work in implementing an ASR system for CA do exist for MSA as well. However, the impact of these on the performance of an MSA system was not as prominent as was observed with CA. This is mainly because of the rich resources in MSA compared to CA and other linguistic and differences between the two variants.

This chapter summarises the main contributions of this thesis and emphasises the significance of them in relation to the existing research. Finally, it outlines some suggestions for future work on this area.

8.1 Scientific contributions

To fully understand the challenges posed by the Arabic language itself, a detailed comparison of CA and MSA in terms of their phonetic, orthographic, morphological and syntactic structures was presented in Chapter 2. This was followed by a discussion of the existing studies in developing ASR systems for CA in Chapter 3, highlighting the main issues. In order to address these issues, the work of this thesis was focused in two main directions: first, investigating the limited lexical coverage and data sparsity in written CA and its implication on language modelling (which was addressed in Chapters 4 and 5); and second, narrowing the gap between the written and spoken forms of CA in the absence of conventional phonetic dictionaries to obtain better models for the acoustics and pronunciations (addressed in Chapters 6 and 7). This work contributed with the following original findings and outcomes as a result of the aforementioned directions of investigation:

8.1.1 Decomposing CA and sub-lexical unit CA LMs

In general, Arabic is a morphologically rich language with a high vocabulary growth rate. Consequently, a selected list of vocabulary would have limited lexical coverage on a given text, hence, high OOV rates are observed. Several approaches were investigated in order to decompose a full word into its sub-lexical units: supervised, semi-supervised and unsupervised decomposition methods. It was shown that:

- Word decomposition limited the vocabulary growth in CA, and reduced the vocabulary size by different degrees depending on the decomposition method used, and whether parts of the vocabulary were excluded from the decomposition process.
- Regardless of the chosen decomposition method, word decomposition reached its best potential in reducing the vocabulary size when the whole set of observed vocabulary was included in the decomposition process.
- A considerable reduction in OOV rates was observed in general such that OOV rates decreased with an increase of linguistic knowledge or statistical constraints involved in the decomposition process.
- Generally, morph-based LMs outperformed word-based LMs in CA, by an average of 3% relative improvement in character-level perplexity, regardless of the decompo-

sition method employed. As the level of CA-related linguistic constraints increased in the decomposition method, the performance improved in terms of character-level perplexities.

8.1.2 Inducing data-driven classes from decomposed data in CA LM development

In order to overcome the data sparsity issue in CA, data-driven clustering on the sub-lexical level was incorporated in the development of language models. Two approaches were employed: For the first approach, classes were induced from the decomposed CA data using Brown et al.'s word clustering algorithm and then used to estimate class LMs (CLMs) (Brown et al., 1992). For the second approach, word histories were clustered in a randomised decision tree model and then used to estimate random forest LMs (RFLMs) (Xu and Jelinek, 2004). It was shown that:

- CA morph-based CLMs did not yield any improvement over standard morph-based LMs or word-based LMs.
- CA morph-based RFLMs outperformed the standard morph-based LMs, with an average improvement in character-level perplexity from RFLMs of 8% relative over word-based LMs.

8.1.3 Colloquialising MSA to be used in CA LM development

The transfer of CA properties to MSA, i.e. normalisation of CA, has been investigated in the context of machine translation. The transfer of MSA properties to CA, i.e. colloquialisation of MSA, was mostly explored as an issue of acoustic modelling, by using either pooling or adaptation approaches. However, work in this thesis can be considered as a first attempt in colloquialisaing MSA in order to generate additional CA data by using existing MSA resources. A colloquialisation system was developed based on an SMT approach, where MSA was cast as a source language and CA as a target language. The main outcomes and findings were as follows:

- An annotated resource, a MSA-LCA parallel corpus, was created as requirement for training the colloquialisation system. Selected LCA sentences were normalised to MSA and validated using a crowdsourcing framework. The corpus contains 30135 pairs of LCA and its normalised variant(s), with an average of 1.3 MSA variants per LCA sentence and an average of 1.2 LCA variants per MSA sentence.

- A novel colloquialisation model was estimated from the parallel MSA-LCA corpus, using an SMT framework. Afterwards, several MSA resources were colloquialised into CA using the colloquialisation model and an SMT decoder.

- Considerable perplexity reduction was achieved with LMs estimated from colloquialised MSA, 68% relative, as compared to LMs estimated from MSA resources.

- The perplexity reduction obtained from interpolating CA LM with a colloquialised MSA LM was twice that of interpolating with any MSA LMs.

8.1.4 Using cross-dialect paraphrastic LMs

Paraphrastic LMs (ParaLMs) (Liu et al., 2012) have been used to capture cross-domain lexical and syntactic structures. Rather than casting MSA and CA as two different dialects, they can be considered as two different domains for the same language; for example politics and medicine domains. Each domain uses a different syntactic structure, which was captured by ParaLMs. It was shown that:

- Excluding disfluency words (such as hesitation and backchannel markers) can improve the quality of the induced paraphrase pairs for text using conversational style.

- Using a dialect-targeted ParaLM outperformed general ParaLMs. For instance, a relative perplexity reduction of between 1.5% and 1.7% was obtained when using ParaLMs estimated from MSA resources but this was outperformed by dialect-targeted ParaLMs, reaching a 2% relative perplexity reduction. Further improvement was achieved when interpolating ParaLMs with standard LMs, giving a 2.8% relative reduction.

8.1.5 Development of CA automatic diacritisation systems

In general, Arabic transcriptions lack short vowels and gemination information which are represented by diacritics. Unlike an English dictionary, a standard Arabic dictionary does not provide pronunciations but only diacritisation variants. A variant is chosen according to the given context. Since diacritics hold one third of the acoustic information, it is crucial for acoustic modelling to retrieve those diacritics. Two novel CA automatic diacritisation systems were implemented in this work:

- A diacritiser based on a grapheme-to-phoneme (G2P) framework was implemented. It requires a small amount of diacritised seeding data (5000 diacritised CA words) to achieve a consistent and highly accurate performance.

- A CRF-based diacritiser allowed the incorporation of contextual and extralinguistic information in the prediction process.

  In addition, it was shown that:

- The incorporation of extralinguistic information, such as speaker's dialect and gender, improved diacritisation performance by a significant margin, reaching 14% relative DWER and 15% relative DER, compared to depending on the grapheme only.

- Using phonological information as a multi-stream feature, place-voicing-manner properties, slightly degraded the diacritisation performance owing to insufficient training samples resulting from increasing the number of parameters to estimated.

- Using automatically diacritised transcriptions as training samples was equivalent to, and sometimes better than, manually transcribed transcriptions for diacritised grapheme acoustic models.

- Whether diacritisation was performed automatically or manually, acoustic models trained using diacritised graphemes outperformed those using undiacritised graphemes, which was found to be statistically significant ($p < 0.001$) based on MAPSSWE.

8.1.6 Identifying pronunciation modelling issues in CA

In this work, it was empirically shown that using diacritised graphemes as acoustic units was not acoustically representative for CA ASR tasks, for several reasons. These include the existence of infrequent acoustic units sharing the same phoneme. In addition, some units have multiple distinctive phonemes and the chosen phoneme can be decided based on the context. For some of these multi-phoneme units, context does not contribute to the choice of the acoustic realisation.

8.1.7 Derivation of CA phonetic transcription

It was found that diacritised graphemes are not the optimal acoustic units and should not be considered equivalent to phonemes. This work proposed a derivation procedure to obtain phonetic transcriptions from diacritised graphemes along with more suitable acoustic units. The new acoustic set is denoted as *phn*. The main outcomes and findings were as follows:

- A derivation process to obtain a pronunciation dictionary along with new acoustic units was implemented from fully diacritised transcriptions. This process based on applying a sequence of phonological rules and disambiguating context-dependent

multi-phoneme units by using the acoustic context through the means of forced alignment.

- The new acoustic set consists of 42 units and shows a more balanced frequency distribution in the training data than diacritised graphemes.
- It was empirically shown that using the new acoustic set outperformed diacritised graphemes by at least 1% absolute WER in an LCA CTS ASR task, which was found to be statistically significant with $p < 0.001$ using MAPSSWE.

8.1.8 Derivation of new CA acoustic units

Another issue in using diacritised graphemes was that of context-independent multi-phoneme units. This was addressed by restructuring the context-dependent *phn* model space. It was denoted as *cX* where *X* was the number of units in that set. The main contributions were as follows:

- A framework was implemented for building a hierarchical structure of the context-dependent model space based on Hartmann et al. (2013)'s HMM similarity metric, using graph degree linkage hierarchical clustering (Zhang et al., 2012). From the resulting hierarchical structure, a *cX* acoustic set was extracted where X corresponds to the number of extracted clusters.
- A framework was implemented for generating a set of questions to be used in building state-tying decision trees for the *cX* acoustic models. This framework employed graph degree linkage hierarchical clustering to construct the state hierarchical structure based on the Cauchy-Schwarz divergence (Kampa et al., 2011) between GMMs.
- Pronunciations were generated using a G2P framework and different configurations for training the conversion model were examined. It was shown that using diacritised baseforms in the training improved accuracy. In addition, the G2P conversion performance dropped as the number of the units in the *cX* increased.
- It was found that using lower numbers of clusters resulted in the merging of phonemic units into one unit which increased the number of homophone words and consequently increased the number of substitution errors, whereas using larger numbers of clusters might lead to under-trained units due to the lack of examples in the training set.
- It was shown that a CTS ASR using *cX* models outperformed those using diacritised graphemes marginally but was found to be statistically significant with $p < 0.001$ but not those using *phn* models.

8.2 List of publications

Some of the findings contributed by this work have been published or to be submitted as international conference and journal papers and technical reports.

- S. Al-Shareef and T. Hain (2011). An Investigation in Speech Recognition for Colloquial Arabic. In Proc. Interspeech'11
- S. Al-Shareef and T. Hain (2012). CRF-based Diacritisation of Colloquial Arabic for Automatic Speech Recognition. In Proc. Interspeech'12
- S. Al-Shareef and T. Hain (2012). Conditional Random Fields Based Diacritisation of Colloquial Arabic. In Proc. the 6th Saudi International Conference (SIC)
- S. Al-Shareef (2013). Conversational Arabic Automatic Speech Recognition: Literature Review. Technical report
- S. Al-Shareef and T. Hain (to be submitted in 2015). Towards a Universal Acoustic Inventory for Colloquial Arabic ASR. Speech Communication.
- S. Al-Shareef and T. Hain (to be submitted in September 2015). Colloquialising Modern Standard Arabic Text for Improved Speech Recognition. ICASSP 2016.
  This paper shows than using the colloquialised MSA resources did not just improved the language model quality (in terms of perplexity) but also in recognition performance in CTS LCA ASR task with a significant 5.4% WER relative improvement over using MSA resources directly.

8.3 Relation to other research work

The work presented in this thesis focused on addressing issues in the following components of CA ASR systems: language model, acoustic and pronunciation model. Several studies have been performed on one component or another using different approaches and mostly designed for MSA originally. In this section, a summary is presented of how the aforementioned outcomes and findings relate to these studies.

Two main issues in language modelling have been addressed. These are the limited lexical coverage and data sparsity of written CA. Limited lexical coverage in written CA is caused by the complex morphology of the language. The proposed methods addressing this issue were evaluated in terms of perplexity and OOV rate reductions. Character-level perplexity and normalised OOV rate were employed when comparing the performance of two LMs where they used different linguistic unit, such as word in opposed to morpheme.

Because of the different linguistic unit used, few studies reported such evaluation and used WER instead which have not been used in this thesis. Unlike any of the previous studies, this work provided a detailed comparison of the use of sub-lexical unit LMs using several word decomposition methods. These methods ranged from linguistic-based, as in MADA, to completely unsupervised, as in Morfessor Baseline.

MSA textual data have been used in previous studies to address the data sparsity issue of CA, using two main approaches: First, pooling CA text with MSA data to be used in estimating a CA LMs (Kirchhoff et al., 2003; Elmahdy et al., 2013); second, an LM was estimated on each Arabic variant individually and then both LMs are linearly interpolated with optimised weights (Kirchhoff et al., 2003; Nguyen et al., 2009). Both approaches reduced perplexity only insignificantly. None of the approaches perform any kind of processing on the MSA data prior using it in LM training. In contrast to previous studies, the proposed methods in this work allows the use of MSA textual resources to reduce the data sparsity issue and improve the performance of an LM on a CA test set, reaching perplexity by 68% relative compared to an MSA LM and 7% relative compared to an interpolated LM of MSA and CA.

In this thesis, issues in acoustic and pronunciation modelling have been addressed. These issues include diacritics omitted from written CA and the fact that a fully diacritised form of CA does not represent its phonetic realisation. The first issue has been already addressed by several studies. Vergyri et al. (2005) built a system that uses POS information. Their method achieved 30% DWER absolute. Both diacritisation systems implemented in this work outperformed Vergyri et al.'s system, reaching 16-23% DWER absolute, with a smaller training set. These results may not be comparable due to the difference in the employed test set. However, the achieved improvement in terms of WER, of 1-1.5% relative, is similar. This recognition improvement also agreed with other studies that used manually diacritised graphemes, instead of undiacritised graphemes as acoustic units (Kirchhoff et al., 2002a; Gales et al., 2007; Diehl et al., 2008; Soltau et al., 2009).

CRF-based diacritisation has been proposed previously by Schlippe et al. (2008). Their work differ from the proposed system in this work in terms of the chosen features in the prediction process. Since the task was diacritising MSA, Schlippe et al. used POS tags which were derived from an expert-made resource. However, the proposed system did not rely on any linguistic resources. Instead, extralinguistic information was used. This included speaker's dialect and gender, both of which can be derived using automatic approaches. Speaker-dependent information has not been used in previous studies for vowelisation or diacritisation of both MSA and CA.

Two new acoustic sets were introduced in this work: one is derived from applying phonological rules and forced-alignment; the other is derived from extracting clusters from

the hierarchical structure of context-dependent models. The obtained recognition improvement from the first acoustic set was 3% WER relative in an LCA CTS ASR task. Biadsy et al. (2009a) had also achieved 4% relative WER improvement on MSA recognition task. Their approach differ from the proposed method here in two aspects: first, phonological rules were applied on MADA-based diacritised transcriptions; second, morphological information were used to disambiguate silent and ambiguous graphemes instead of using forced-alignment.

The second proposed acoustic set followed a similar approach proposed by Hartmann et al. (2013) for acoustic unit discovery. In their work, spectral-based clustering was used on a similarity matrix derived from context-dependent grapheme units. Spectral-based clustering generated acoustically separated clusters when applied to CA. The generated clusters merged many phonemic units into one unit. As a consequence, many homophone words appeared and increased the amount of substitutions in the recognition output. Using hierarchical agglomerative clustering based on graph degree linkage distinguish this work from Hartmann et al. (2013)'s. The generated clusters merged acoustically similar context-dependent units while retaining phonemic separation between these units. A further distinction lies in the set of questions chosen for creating state-tying decision tree to be used for the context-dependent units of these models. These questions were generated based on the computed similarity between the states instead of using just the identity questions.

8.4 Future work

In this thesis, two directions of work were investigated to address the under-performance issues in CA ASRs. During the course of these investigations, several research directions remained unexplored and might serve as potential starting points for future work.

8.4.1 Comparison between MSA and CA using the proposed techniques

As it was established through the course of this work, MSA and CA differ in acoustics, pronunciations. Such acoustic differences are hidden in the undiacritised form of writing. Therefore, some of the presented findings above might be valid for the MSA as well.

8.4.2 Optimising the un-decomposed word in morph-based LMs

Although the best OOV rate reduction was obtained when including all observed vocabulary in the word decomposition process, the character-level perplexity was increased compared to excluding the most frequent words from the decomposition. The number of

excluded words was chosen based on a reported experiment in MSA (El-Desoky et al., 2010). Because such a choice is highly dependent on the corpus and language used, it would be preferable to find an automatic optimisation method for this selection.

8.4.3 Adaptive acoustic set extraction

In this work, the hierarchical structure context-dependent acoustic model space based on HMM similarity was extracted using graph degree linkage hierarchical clustering. From this structure a finite set of X clusters was extracted. Each cluster represents a set of acoustically similar acoustic models based on the computed distance between their HMMs. The empirical results showed that increasing the number of extracted clusters, i.e. X, was not beneficial. The extraction of these clusters from the dendrogram was performed using a fixed height cut method. The objective of fixed height cut extraction is to retrieve X clusters with the best separation between them. Such clusters may not be the best choice for acoustic modelling because two clusters might be acoustically close to each other but they are phonemic and must remain separate. Therefore, an alternative strategy to fixed height cut should be employed for defining the cut-off points in the dendeogram while retaining the desired number of clusters. In addition, finding the most suitable X can be automatically optimised.

Such technique will allow creating an acoustic set that fits a certain dialect or language using some initial pronunciation based on acoustic units from a different language as long as they provide some phonetic separation.

8.5 Summary

This thesis addressed the under-performance issues in CA ASR systems by undertaking a thorough investigation in two directions. The first direction focused on the limited lexical coverage and insufficient training samples of written CA. As a result of addressing these issues, methods in language modelling of CA were proposed. The second direction investigated narrowing the gap between written and spoken CA in the absence of phonetic dictionaries. Several original contributions resulted from each direction which improved the system by at least 3% WER relative. Finally, some suggestions for further research in this topic were presented.

Appendix A

MSA and CA phonemes

Table A.1: MSA consonants and their pronunciation in CA

Class	MSA grapheme	MSA phoneme	CA phoneme	Notes
Bilabial	م	m	m	
	ب	b	b	
			p	Used for load words between educated members
Labiodental	ف	f	f	
			v	Used for loan words between educated members
Dental	ث	θ	θ	In nomadic dialects in the Arabian Peninsula, Tunisia, Palestine, Syria, and Mesopotamia.
			t	In some Morocco dialects
			s	In Egypt, the large cities of Syria and Lebanon and many neighbouring areas.
			f	In southern Anatolian Siirt.
	ذ	ð	ð	In nomadic dialects in the Arabian Peninsula, Tunisia, Palestine, Syria, and Mesopotamia.
			z	In Egypt, the large cities of Syria and Lebanon and many neighbouring areas.

Continued on next page

Class	MSA grapheme	MSA phoneme	CA phoneme	Notes
	ظ	ðˤ	ðˤ	In nomadic dialects in the Arabian Peninsula, Tunisia, Palestine, Syria, and Mesopotamia.
			dˤ	In Egypt, the large cities of Syria and Lebanon and many neighbouring areas.
			z	In Egypt, the large cities of Syria and Lebanon and many neighbouring areas.
Alveolar	ت	t	t	
	د	d	d	
	ط	tˤ	tˤ	
	ض	dˤ	dˤ	
	ن	n	n	In some modern dialects such as Lebanese
			ɾ	Some of Egyptian words
	س	s	s	
			ʃ	Few dialects spoken in the Morocco, in Farafra and Central Bahariyya spoken in the oases of the western desert of Egypt
	ش	ʃ	ʃ	
	ز	z	z	
			ʒ	Few dialects spoken in the Morocco, in Farafra and Central Bahariyya spoken in the oases of the western desert of Egypt
	ص	sˤ	sˤ	
			s	in Sudan and in some dialects spoken in the western mountain range of northern Yemen
	ل	l	l	
	لّ		ɬ	Exclusively in Allah ”God” and its derivatives

Continued on next page

Class	MSA grapheme	MSA phoneme	CA phoneme	Notes
	ج	ʤ	ʤ	In Bedouin dialects, in many rural Syrian, Jordanian, Palestinian, and Mesopotamian dialects, in the central region of northern Yemen and many parts of the Arabian Peninsula
			g	In Egyptian and Yamani dialects
			j	In the Syrian desert, Khuzistan, Hadramawt, Dhofar, and the Gulf dialects
			ʒ	In many areas of the Levant, especially the major cities of Beirut, Damascus, and Jerusalem and in the majority of Maghribi dialects
Palatal	ج	j	j	
			i	Between consonants
Velar	ك	k	k	
			tʃ	in some Gulf dialects
	غ	ʁ	ʁ	
	خ	χ	χ	
Uvular	ق	q	q	in many Syrian and North African dialects, in the North Mesopotamian.
			g	in the west and south of the Arabian Peninsula.
			ʔ	in the large cities around the Mediterranean, Cairo, Jerusalem, Damascus, and Beirut. [but certain religious and Standard Arabic words are pronounced with q]
			ʒ	in some Gulf dialects
			k	in some dialects in Iraq and Palestine.
			ʁ	in Sudanese
Pharyngeal	ح	ħ	ħ	
			h	In Nigerian and Chadian
	ع	ʕ	ʕ	

Continued on next page

Class	MSA grapheme	MSA phoneme	CA phoneme	Notes
			ʔ	In Nigerian and Chadian
Glottal	ء	ʔ	ʔ	Some Arabian Peninsula dialects
			vowel	replaced by compensatory lengthening of the vowel in preconsonantal position
			glide	between two vowels of differing quality the glottal stop is usually replaced by a glide.
			-	
	هـ	h	h	
Glide	و	w	w	
			u	Between consonants

Appendix B

Resouces and experimental paradigms

Contents

B.1 Resources

In this thesis there were three sets of experiments: CTS experiments, phone recognition and language modelling experiments.

B.1.1 CTS ASR experiments

Two sets from LCA were included: AppenLCA (Appen, 2006e; 2007) and FisherLCA (Maamouri et al., 2006; 2007). These sets were distributed by the Linguistic Data Consortium (LDC). They represent conversations by native speakers talking with their friends and families, and unrelated people about topics suggested by the corpus collectors. AppenLCA was collected by Appen and FisherLCA was collected by Fisher. All speech segments have been preprocessed where all segments with speaker changes, foreign speakers or overlapped speech were excluded.

Table B.1: Data specifications for CTS ASR experiments.

(a) Speech data

Data set	FisherLCA hours	AppenLCA hours	Total hours
trainLCA	143.3	41.2	184.5
testLCA	5.1	0.8	5.9

(b) Textual data

Data set	FisherLCA		AppenLCA		Total	
	sentences	words	sentences	words	sentences	words
trainLCA	18,726	77,848	11,409	77,194	30,135	155,042
testLCA	10,121	52,253	1,224	9,294	11345	50,202

AppenLCA defined a training and testing sets but not FisherLCA. Hence, around 5 hours of FisherLCA were chosen as a test set. To maintain homogeneous and balanced recording conditions, FisherLCA test set was constructed by the random selection of conversation sides from the FisherLCA with the objective to have good coverage of all sub-dialects and speakers. A selection on a side-level instead of conversation-level was preferred for better speaker coverage, while allowing speaker set separation between training and test sets.

The orthographic transcripts were generated by LDC in a semi-diacritised form using standard MSA scripts. All diacritics were discarded for consistency. All transcriptions were normalised by mapping different backchannel tags into one tag and the same was applied for hesitation tags.

A master LCA sets were constructed from merging FisherLCA and AppenLCA training sets and test sets, obtaining trainLCA and testLCA, respectively. Table B.1 shows the characteristics of these sets and Table B.2 shows the gender and sub-dialect distributions within the whole trainLCA and testLCA.

B.1.2 Phone recognition experiments

A dialect and gender balanced 30 hours of speech was constructed from three dialectical corpora: Gulf (Appen, 2006a;b) , Iraqi (Appen, 2006c;d) and Levantine CA collected by Appen. As AppenLCA, these sets represent conversations by native speakers talking with their friends and families, and unrelated people about topics suggested by the corpus collectors. Similar preprocessing were performed on these sets to AppenLCA. A test set was constructed by the random side-level selection of 10% from the conversation sides of each data set. The Arabic orthographic transcripts were provided by Appen in a fully-diacritised form using standard MSA scripts and Buckwalter transliteration.

Table B.2: Amount of data (hours) for dialect and gender on training set "trainLCA" and test set "testLCA". Lev: general Levantine dialect, UNK: unknown dialect, which could be non-Levantine.

	trainLCA				testLCA		
		Gender				Gender	
dialect	size	m	f	unk	size	m	f
SYR	2.84	0.92	1.86	0.07	1.32	0.46	0.86
PAL	19.30	13.28	6.02	0	1.26	0.75	0.52
JOR	29.49	17.30	10.24	1.95	1.32	0.67	0.66
LEB	91.69	58.17	33.52	0	1.20	0.60	0.60
Lev	23.54	11.49	12	0.05	0	0	0
UNK	4.09	0	0	4.09	0	0	0
Total	170.96	101.16	63.63	6.17	5.10	2.46	2.64

Finally, a 10-hour modern standard Arabic (MSA) data set was included from the down-sampled WestPoint corpus (LaRocca and Chouairi, 2002). The transcripts were provided by the corpora distributor in a fully-diacritised form.

A similar approach for defining the Arabic set was adopted for defining its English counterpart. 50 hours of North American English telephone conversations were drawn randomly from a pool of Fisher corpora (Cieri et al., 2004a;b), which exhibited similar condition to the selected Arabic corpora. The selected English data set was divided into 5 dialects based on the speaker information: Canadian (CAN), midland (MID), northern (NTH), southern (STH) and western (WST) dialects. Phone-level transcriptions were generated through forced-alignment using a pronunciation dictionary which is mostly written by human experts. This data will be used for the contrastive experiments in English.

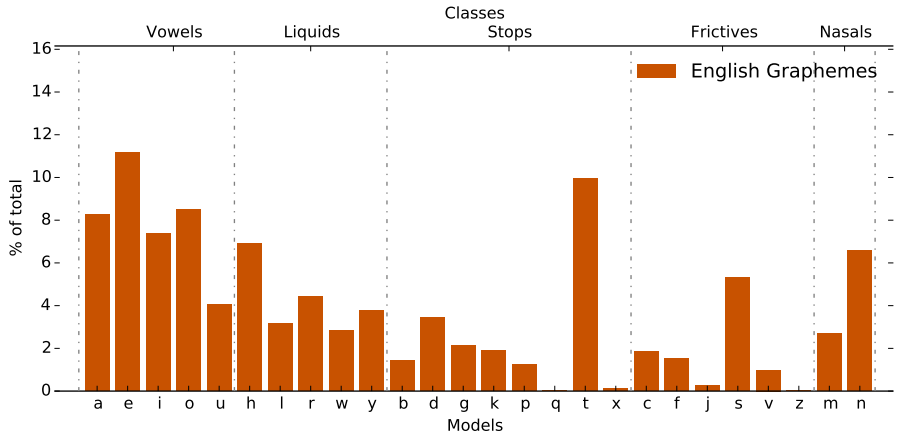
Figure B.1 shows the frequency of different acoustic units in the transcriptions.

B.1.3 Language modelling experiments

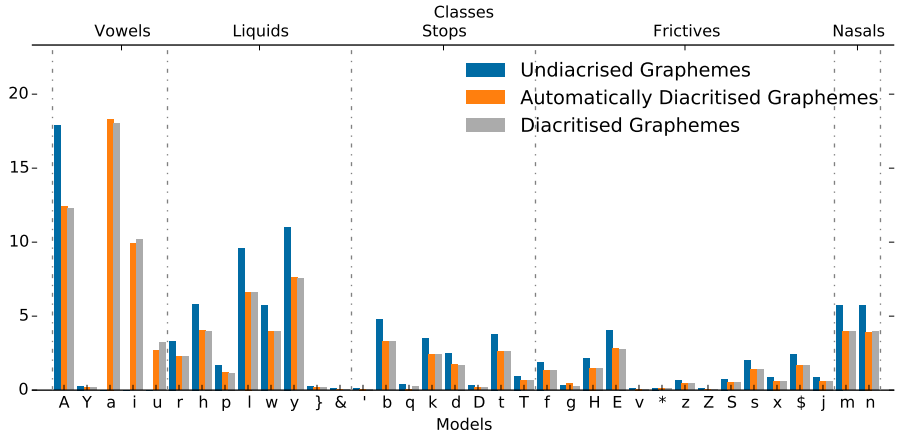
For these experiments, two Arabic variants were included: CA and MSA. CA data were based on the transcriptions of the corpora mentioned previously: AppenLCA, FisherLCA, ICA and GCA. But without any segment exclusion. All these transcriptions were provided in undiacritised form and were normalised by removing all word fragments and hesitations from the transcriptions. All backchannel tags were mapped into one tag instead.

In addition, two textual MSA resources were included: NW10 and BC. NW10 refer An-Nahar subset of the Arabic Gigaword corpus that is a newswire resource for a Lebanese newspaper. BC is collection of transcriptions from Arabic broadcast news shows which were collected under GALE project. Both were distributed by LDC.

Table B.3 lists some basic statistics for each of these sets.



(a) English



(b) Arabic

Figure B.1: Graphemic units average frequency distribution across 9-hour sets from different dialect in English and Arabic.

Table B.3: Data specifications for language modelling experiments.

	trainLCA	GCA	ICA	BC	NW10
sentences	433076	54921	24148	89816	1477544
words	1906286	344383	167263	1433932	15779447
average word/sentence	4.4	6.3	6.9	16.0	10.7
unique words	81636	29862	17976	102629	424922
average word length	5.8	5.5	5.4	6.07795	6.7

B.2 Experimental paradigm

B.2.1 Phone recognition system configuration

Throughout this thesis, all phone recognition experiments shared an identical front-end and training framework, which employed HTK (Young et al., 2006). The audio data was segmented using the time boundaries provided by the corpora transcriptions. A 13 dimensional perceptual linear prediction coefficients (PLP) features, in addition to their first and second derivatives, were extracted every 10ms using a 25ms Hamming window. Cepestral mean subtraction and variance normalisation are applied to each segment during training and testing. Gender independent models are trained using a standard mix-up maximum likelihood regime with left and right context tri-units. Left-to-right HMMs with three emitting states were used and clustered at the state level using a binary decision tree with phonologically motivated graphemic questions (Odell and Woodland, 1994) to gradually obtain output distributions with 32 mixture components for each state with 900 clustered states. None of the advanced acoustic modelling techniques, such as adaptation and discriminative features, were employed.

Evaluation was performed through three tasks. First assessment was a context-free phone recognition task in which the number of recognized units is not controlled, where insertion and deletion errors can occur. Secondly, to assess the inter-class confusion with the unit sets, forced-alignment was used on unit-level, where each reference grapheme model can be forced-aligned to any grapheme. Finally, a more constrained experiment was performed to assess the intraclass confusion, where a reference grapheme model can be force-aligned with graphemes belonging to the same class of the reference grapheme. In the latter two experiments, the number of recognized units per utterance is controlled to the reference.

Unit-level references with time boundaries were generated by force aligning the reference units using the corresponding units. Four error rates were considered in the assessment for these tasks: unit error rate, frame error rate, class error rate and class frame error rate. In the class-based metrics, units in hypothesis and references were mapped onto their corresponding phonetic class before computing their error rate. Class-based error rates are more flexible by which errors are counted only if the incorrectly assigned unit does not belong to the same phonetic class as the reference unit.

B.2.2 CTS ASR system configuration

The systems used for tasks under this category have a similar front-end as those described in Section B.2. Due to the higher amount of data than that described in the unit recognition tasks, after a gradual increase of Gaussian mixture components models contained 16-

mixture components for each state with 2400 clustered states when training on AppenLCA with 3800 clustered states when training on FisherLCA.

For each training set, a language model was constructed using SRILM toolkit (Stolcke, 2002) based on the vocabulary of 41.69k words. Both language models are standard 3-gram model that are trained using modified Kneser-Ney discounting and backoff. An interpolated language model estimated from these two language models were used in all speech recognition experiments.

Appendix C

The use of crowdsourcing in standardising LCA

الترجمة الحرفية من العربية الشامية إلى الفصحى

السلام عليكم و رحمة الله و بركاته

نشكركم جميعا مقدما على وقتكم الثمين في معاونتنا في هذه المهمة...

حسن .. **ما المطلوب بالضبط منكم؟**

باختصار التحويل من العربية باللهجة الشامية إلى العربية الفصحى

مثال:

(قبل التحويل)	لأ إنت مع العائلة هيك أنا بسألك إنت مع مين بالبيت
(بعد التحويل)	لا أنت مع العائلة لذلك أنا أسألك أنت مع من بالبيت

مثال آخر:

(قبل التحويل)	لأ جد أها بس كيف أثرت عليك عطفولتك يعني أعطيني مثال (يعن)
(بعد التحويل)	لا جد أها لكن كيف أثرت عليك على طفولتك يعني أعطيني مثال (يعن)

ما سيتم تحويله...
محادثات هاتفية بين أشخاص قد تكون هناك بينهم معرفة أو لا بمجموعة من اللهجات الشامية (و التي تشمل اللبنانية و الأردنية و الفلسطينية و السورية).. و قد قيل لأغلبهم أن هذه المكالمة هي للاشتراك في مسابقة ما و أن عليهم التحدث لمدة معينة في موضوع محدد (يختلف من محادثة للأخرى).. فستجد أن أغلب المحادثات تبدأ بالتعارف ثم التساؤل عن المسابقة (والتشكيك أحياناً في مآربها) ثم محاولة خلق الحوار في الموضوع المطروح ... ما ستجدونه هو غالبا طرف واحد من الحوار.. و ليس كل الأطراف.. و ستجدون الكثير من المقاطعة و التأكيد و الإيجاب (مثل: إيه .. أها)
الرجاء ملاحظة أنه قد تم تطبيق التحويل على النص بشكل آلي.. لذلك لا تستغرب وجود بعض الكلمات المحولة مسبقا..

القوانين المتبعة للتحويل ...

- تحويل الكلمات دون إعادة ترتيبها حتى لو كانت الجملة الناتجة غير مفهومة... أي أن المطلوب هو الترجمة الحرفية لا بالمعنى
- من الممكن أن يكون التحويل من كلمة إلى جملة
  - مثال: عطفولتك << على طفولتك
- من الممكن أن يكون التحويل من جملة إلى كلمة
  - مثال: عشان هم << لأنهم

- قد لا يكون للكلمة مرادف بالفصحى كأسماء الأشخاص و الأماكن و المصطلحات و الكلمات الأجنبية.. فلا تتغير
- الرجاء عدم التنويع في المرادفات و اختيار مرادفة واحدة للتحويل .(إلا في حال الحاجة لاتمام المعنى) .فالهدف الحصول على أقل عدد ممكن من الكلمات بالفصحى و المرادفة للشامية..
- بعض الكلمات تتغير ترجمتها الفصحى حسب موضعها .. و إليك مثال لكلمة بس:
  - بس لازم نروح << لكن يجب أن نذهب
  - عندي خمس ولاد بس << عندي خمس أولاد فقط
- أرجو عدم استخدام التشكيل
- إذا كانت الكلمة بحد ذاتها تنتمي إلى الفصحى .. فلا داعي إلى تغييرها

Figure C.1: Instructions for the annotators (first page)

التطبيق:

- أفضل طريقة للكتابة في الملف هي باستخدام برنامج المفكرة (notepad) على نظام الويندوز
- مثال لمحتوى الملفات:

#l06_0399_B_M358320J_110.17_112.11 كيف أها قديه صار أها #l06_0399_B_M358320J_116.05_117.50 أيوه

الرجاء عدم تغير الجزء الأول من كل سطر و المكتوب باللغة الانجليزية.. حيث أنه يمثل هوية السطر في الدراسة (من أي محادثة و هوية المتحدث و لهجته إلخ) ..

- في حال كان المحتوى معكوس (من اليسار إلى اليمين) .. يمكنك تصحيح بالضغط على زري shift و ctrl في جهة اليمين من لوحة المفاتيح
- الرجاء كتابة جميع الكلمات التي تم تحويلها للفصحى مع مرادفها في ملف منفصل كالأتي:

مين : من
ولاد : أولاد
بس : فقط
بس : لكن
عطفولتك : على طفولتك
عشان هم : لأنهم

ما ستم تسليمه

- ملف المحادثات المحول إلى الفصحى
- ملف المرادفات

ملاحظة :

اعتذر مسبقا اذا صادفت في بعض المحادثات كلمات و ألفاظ نابية .. فقد حاولت ازالتها بشكل آلي قدر الإمكان.. و لكن قد يصدف أن يسقط بعضا سهوا..

و لكم جزيل شكري و خالص دعواتي..
محبتكم .. سارة الشريف
sarah.alshareef : SkypeID

Figure C.2: Instructions for the annotators (second page)

Appendix D

Derivation of the intergal of a product of two Gaussian distributions

A Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ is defined as:

$$\mathcal{N}(\mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}}\ e^{-\frac{(x-\mu)^2}{2\sigma^2})} \tag{D.1}$$

An integral for a product of two Gaussian distributions $\mathcal{N}(\mu_1, \sigma_1^2)$ and $\mathcal{N}(\mu_2, \sigma_2^2)$, can be computed as:

$$\begin{aligned}
\int \mathcal{N}(\mu_1, \sigma_1^2)\ .\ \mathcal{N}(\mu_2, \sigma_2^2) &= \int \frac{1}{\sigma_1\sqrt{2\pi}}\ e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}}\ .\ \frac{1}{\sigma_2\sqrt{2\pi}}\ e^{-\frac{(x-\mu_2)^2}{2\sigma_2^2}} \\
&= \int \frac{1}{2\pi\sigma_1\sigma_2}\ e^{-\left[(\frac{1}{2\sigma_1^2}x^2 - \frac{\mu_1}{\sigma 1^2}x + \frac{\mu_1^2}{2\sigma_1^2}) + (\frac{1}{2\sigma_2^2}x^2 - \frac{\mu_2}{\sigma 2^2}x + \frac{\mu_2^2}{2\sigma_2^2})\right]} \\
&= \int \frac{1}{2\pi\sigma_1\sigma_2}\ e^{-\left[(\frac{1}{2\sigma_1^2} + \frac{1}{2\sigma_2^2})x^2 - (\frac{\mu_1}{\sigma 1^2} + \frac{\mu_2}{\sigma 2^2})x + (\frac{\mu_1^2}{2\sigma_1^2} + \frac{\mu_2^2}{2\sigma_2^2})\right]}
\end{aligned} \tag{D.2}$$

Assume that exponential in Equation D.2 is written as a quadratic form as follows:

$$(\frac{1}{2\sigma_1^2} + \frac{1}{2\sigma_2^2})x^2 - (\frac{\mu_1}{\sigma 1^2} + \frac{\mu_2}{\sigma 2^2})x + (\frac{\mu_1^2}{2\sigma_1^2} + \frac{\mu_2^2}{2\sigma_2^2}) \quad = \quad A(x-B)^2 + C \tag{D.3}$$

Since the integral of an arbitrary Gaussian function [] is:

$$\int_{-\infty}^{\infty} e^{-A(x-B)^2} dx = \sqrt{\frac{\pi}{A}} \tag{D.4}$$

then Equation D.2 is rewritten as:

$$\begin{aligned}
\int \frac{1}{2\pi\sigma_1\sigma_2}\, e^{-\left[(\frac{1}{2\sigma_1^2}+\frac{1}{2\sigma_2^2})x^2-(\frac{\mu_1}{\sigma 1^2}+\frac{\mu_2}{\sigma 2^2})x+(\frac{\mu_1^2}{2\sigma_1^2}+\frac{\mu_2^2}{2\sigma_2^2})\right]} &= \int \frac{1}{2\pi\sigma_1\sigma_2}\, e^{-[A(x-B)^2+C]} \\
&= \frac{1}{2\pi\sigma_1\sigma_2} \int e^{-[A(x-B)^2+C]} \\
&= \frac{1}{2\pi\sigma_1\sigma_2} \int e^{-A(x-B)^2-C} \\
&= \frac{1}{2\pi\sigma_1\sigma_2}\, e^{-C} \sqrt{\frac{\pi}{A}}
\end{aligned} \tag{D.5}$$

To find the A and C, from Equation D.3:

$$\begin{aligned}
(\frac{1}{2\sigma_1^2}+\frac{1}{2\sigma_2^2})x^2-(\frac{\mu_1}{\sigma_1^2}+\frac{\mu_2}{\sigma_2^2})x+(\frac{\mu_1^2}{2\sigma_1^2}+\frac{\mu_2^2}{2\sigma_2^2}) &= A(x-B)^2+C \\
&= Ax^2-2ABx+AB^2+C
\end{aligned} \tag{D.6}$$

then:

$$A = \frac{1}{2\sigma_1^2}+\frac{1}{2\sigma_2^2} \tag{D.7}$$

$$2AB = \frac{\mu_1}{\sigma_1^2}+\frac{\mu_2}{\sigma_2^2} \tag{D.8}$$

$$AB^2+C = \frac{\mu_1^2}{2\sigma_1^2}+\frac{\mu_2^2}{2\sigma_2^2} \tag{D.9}$$

By substituting the A from Equation D.7 into Equation D.8, then solve the resulting equation to get B:

$$
\begin{aligned}
2(\frac{1}{2\sigma_1^2}+\frac{1}{2\sigma_2^2})\ B &= \frac{\mu_1}{\sigma_1^2}+\frac{\mu_2}{\sigma_2^2} \\
2\ \frac{1}{2}\ (\frac{\sigma_2^2+\sigma_1^2}{\sigma_1^2\sigma_2^2})\ B &= \frac{\mu_1\sigma_2^2+\mu_2\sigma_1^2}{\sigma_1^2\sigma_2^2} \\
B &= \frac{\mu_1\sigma_2^2+\mu_2\sigma_1^2}{\sigma_2^2+\sigma_1^2}
\end{aligned} \tag{D.10}
$$

By substituting the A from Equation D.7 and B from Equation into Equation D.9, then solve the resulting equation to get C:

$$
\begin{aligned}
AB^2+C &= \frac{\mu_1^2}{2\sigma_1^2}+\frac{\mu_2^2}{2\sigma_2^2} \\
\left(\frac{1}{2\sigma_1^2}+\frac{1}{2\sigma_2^2}\right)\left(\frac{\mu_1\sigma_2^2+\mu_2\sigma_1^2}{\sigma_2^2+\sigma_1^2}\right)^2+C &= \frac{1}{2}\left(\frac{\mu_1^2\sigma_2^2+\mu_2^2\sigma_1^2}{\sigma_1^2\sigma_2^2}\right) \\
\frac{1}{2}\left(\frac{\sigma_2^2+\sigma_1^2}{\sigma_1^2\sigma_2^2}\right)\left(\frac{(\mu_1\sigma_2^2+\mu_2\sigma_1^2)^2}{(\sigma_2^2+\sigma_1^2)^2}\right)+C &= \frac{1}{2}\left(\frac{\mu_1^2\sigma_2^2+\mu_2^2\sigma_1^2}{\sigma_1^2\sigma_2^2}\right) \\
\frac{(\mu_1\sigma_2^2+\mu_2\sigma_1^2)^2}{2\sigma_1^2\sigma_2^2(\sigma_2^2+\sigma_1^2)}+C &= \frac{\mu_1^2\sigma_2^2+\mu_2^2\sigma_1^2}{2\sigma_1^2\sigma_2^2} \\
C &= \frac{\mu_1^2\sigma_2^2+\mu_2^2\sigma_1^2}{2\sigma_1^2\sigma_2^2}-\frac{(\mu_1\sigma_2^2+\mu_2\sigma_1^2)^2}{2\sigma_1^2\sigma_2^2(\sigma_2^2+\sigma_1^2)} \\
C &= \frac{1}{2\sigma_1^2\sigma_2^2}\left(\mu_1^2\sigma_2^2+\mu_2^2\sigma_1^2-\frac{(\mu_1\sigma_2^2+\mu_2\sigma_1^2)^2}{\sigma_2^2+\sigma_1^2}\right)
\end{aligned}
$$

$$
\begin{aligned}
C &= \frac{1}{2\sigma_1^2\sigma_2^2}\left(\frac{\mu_1^2\sigma_2^4+\mu_1^2\sigma_2^2\sigma_1^2+\mu_2^2\sigma_1^2\sigma_2^2+\mu_2^2\sigma_1^4-\mu_1^2\sigma_2^4-2\mu_1\mu_2\sigma_1^2\sigma_2^2-\mu_2^2\sigma_1^4}{\sigma_2^2+\sigma_1^2}\right) \\
&= \frac{1}{2\sigma_1^2\sigma_2^2}\left(\frac{\mu_1^2\sigma_2^2\sigma_1^2+\mu_2^2\sigma_1^2\sigma_2^2-2\mu_1\mu_2\sigma_1^2\sigma_2^2}{\sigma_2^2+\sigma_1^2}\right) \\
&= \frac{1}{2\sigma_1^2\sigma_2^2}\left(\frac{\sigma_1^2\sigma_2^2(\mu_1^2+\mu_2^2-2\mu_1\mu_2)}{\sigma_2^2+\sigma_1^2}\right) \\
&= \frac{(\mu_1-\mu_2)^2}{2(\sigma_2^2+\sigma_1^2)}
\end{aligned} \tag{D.11}
$$

By substituting A and C from Equations D.7 and D.11 respectively in Equation D.5:

$$
\begin{aligned}
\frac{1}{2\pi\sigma_1\sigma_2}\; e^{-C}\; \sqrt{\frac{\pi}{A}} &= \frac{1}{2\pi\sigma_1\sigma_2}\; e^{-\frac{(\mu_1-\mu_2)^2}{2(\sigma_2^2+\sigma_1^2)}}\; \sqrt{\frac{\pi}{\frac{1}{2}\left(\frac{\sigma_2^2+\sigma_1^2}{\sigma_1^2\sigma_2^2}\right)}} \\
&= \frac{1}{2\pi\sigma_1\sigma_2}\; e^{-\frac{(\mu_1-\mu_2)^2}{2(\sigma_2^2+\sigma_1^2)}}\; \sqrt{\frac{2\pi\sigma_1^2\sigma_2^2}{\sigma_2^2+\sigma_1^2}} \\
&= \frac{1}{2\pi\sigma_1\sigma_2}\; e^{-\frac{(\mu_1-\mu_2)^2}{2(\sigma_2^2+\sigma_1^2)}}\; \frac{\sigma_1\sigma_2\sqrt{2\pi}}{\sqrt{\sigma_2^2+\sigma_1^2}} \\
&= \frac{1}{\sqrt{2\pi(\sigma_2^2+\sigma_1^2)}}\; e^{-\frac{(\mu_1-\mu_2)^2}{2(\sigma_2^2+\sigma_1^2)}} \\
&= \mathcal{N}(x=0;\mu_2-\mu_1,\sigma_1^2+\sigma_2^2) \\
&= \mathcal{N}(\mu_1;\mu_2,\sigma_1^2+\sigma_2^2) \qquad \text{(D.12)}
\end{aligned}
$$

References

Abdur-Rasheed, A. (2008). *"Al-Hidayah fi Al-Nahw" by Ibn-Hayyan - the guidance in Arabic grammar*. Academy of Islamic Sciences.

Abudalbuh, M. (2010). *Effects of gender on the production of emphasis in Jordanian Arabic: A sociophonetic study*. PhD thesis, University of Kansas.

Adda-Decker, M. and Adda, G. (2000). Morphological decomposition for ASR in german. In *Workshop on Phonetics and Phonology in Automatic Speech Recognition*, pages 129–143.

Afify, M., Nguyen, L., Xiang, B., Abdou, S., and Makhoul, J. (2005). Recent progress in Arabic broadcast news transcription at BBN. In *Proc INTERSPEECH*, pages 1637–1640, Lisbon, Portugal.

Afify, M., Sarikaya, R., and Kuo, H. (2006). On the use of morphological analysis for dialectal Arabic speech recognition. In *Proc INTERSPEECH*, pages 1444–1447.

Afify, M., Siohan, O., and Sarikaya, R. (2007). Gaussian mixture language models for speech recognition. In *Proc ICASSP*, volume 4.

Al-Farahidi, I. A. A.-K. (1980). *Kitab Al-'Ayn*. Dar wa Maktabat Al-Hilal, Baghdad.

Al-Haj, H., Hsiao, R., Lane, I., Black, A., and Waibel, A. (2009). Pronunciation modeling for dialectal Arabic speech recognition. In *Proc ASRU*, pages 525–528.

Al-Sabbagh, R. and Girju, R. (2010). Mining the web for the induction of a dialectical Arabic lexicon. In *Proc LREC*, pages 288–293, Valletta, Malta.

Al-Wer, E. (2002). Jordanian and Palestinian dialects in contact: vowel raising in Amman. *Contributions To The Sociology Of Language*, 86:63–80.

Alghamdi, M. (2000). *AlSawTiyat AlEarabyap*. Attaoobah, Riyadh.

Alghamdi, M. and Muzaffar, Z. (2007). KACST Arabic diacritizer. In *Proc International Symposium on Computers and Arabic Language*, pages 25–28, Riyadh, Saudi Arabia.

Ali, A. I. (1756). *Marāḥ al-arwāḥ*. Dar Alkutub.

Ali, M., Elshafei, M., Al-Ghamdi, M., Al-Muhtaseb, H., and Al-Najjar, A. (2008). Generation of Arabic phonetic dictionaries for speech recognition. In *Proc Int Conf Innovations in Information Technology*, pages 59–63, Al Ain, United Arab Emirates.

Allauzen, C., Riley, M., Schalkwyk, J., Skut, W., and Mohri, M. (2007). OpenFst: a general and efficient weighted finite-state transducer library. In Holub, J. and Žďárek, J., editors, *Implementation and Application of Automata*, pages 11–23. Springer.

Aminian, M., Ghoneim, M., and Diab, M. (2014). Handling OOV words in dialectal Arabic to English machine translation. In *Proc EMNLP Workshop on Language Technology for Closely Related Languages and Language Variants*, page 99, Doha, Qatar.

Ananthakrishnan, S., Narayanan, S., and Bangalore, S. (2005). Automatic diacritization of Arabic transcripts for ASR. In *ICON-05*, Kanpur, India.

Appen (2006a). Gulf Arabic Conversational Telephone Speech (LDC2006S43).

Appen (2006b). Gulf Arabic Conversational Telephone Speech, Transcripts (LDC2006T15).

Appen (2006c). Iraqi Arabic Conversational Telephone Speech (LDC2006S45).

Appen (2006d). Iraqi Arabic Conversational Telephone Speech, Transcripts (LDC2006T16).

Appen (2006e). Levantine Arabic Conversational Telephone Speech (LDC2007S01).

Appen (2007). Levantine Arabic Conversational Telephone Speech, Transcripts (LDC2007T01).

Arısoy, E., Kurimo, M., Saraçlar, M., Hirsimäki, T., Pylkkönen, J., Alumäe, T., and Sak, H. (2008). Statistical language modeling for automatic speech recognition of agglutinative languages. In Mihelic, F. and Zibert, J., editors, *Speech Recognition, Technologies and Applications*. I-Tech.

Aubert, X. and Ney, H. (1995). Large vocabulary continuous speech recognition using word graphs. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, volume 1, pages 49–52. IEEE.

Bahl, L., Baker, J., Jelinek, F., and Mercer, R. (1977). Perplexity – a measure of the difficulty of speech recognition tasks. *Journal Acoustical Society of America*, 62:S63.

Bahl, L. R., Brown, P., de Souza, P. V., and Mercer, R. (1989). A tree-based statistical language model for natural language speech recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37(7):1001–1008.

Bahl, L. R., Jelinek, F., and Mercer, R. L. (1983). A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5:179–190.

Baker, J. (1975). The DRAGON system – an overview. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 23(1):24–29.

Bakr, H. A., Shaalan, K., and Ziedan, I. (2008). A hybrid approach for converting written Egyptian colloquial dialect into diacritized Arabic. In *Proc 6th Int Conf Informatics and Systems*, pages 27–33, Cairo, Egpyt.

Baum, L. E., Petrie, T., Soules, G., and Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171.

Beesley, K. R. (1996). Arabic finite-state morphological analysis and generation. In *Proceedings of the 16th conference on Computational linguistics*, volume 1, pages 89–94. Association for Computational Linguistics.

Bellman, R. (1957). *Dynamic Programming*. Princeton University Press, New York.

Berton, A., Fetter, P., and Regel-Brietzmann, P. (1996). Compound words in large-vocabulary German speech recognition systems. In *Proc ICSLP*, volume 2, pages 1165–1168. IEEE.

Beulen, K. and Ney, H. (1998). Automatic question generation for decision tree based state tying. In *Proc ICASSP*, volume 2, pages 805–808. IEEE.

Biadsy, F., Habash, N., and Hirschberg, J. (2009a). Improving the Arabic pronunciation dictionary for phone and word recognition with linguistically-based pronunciation rules. In *Proc NAACL HTL*, pages 397–405.

Biadsy, F., Hirschberg, J., and Habash, N. (2009b). Spoken Arabic dialect identification using phonotactic modeling. In *Proc EACL Workshop on Computational Approaches to Semitic Languages*, pages 53–61. Association for Computational Linguistics.

Biadsy, F., Moreno, P. J., and Jansche, M. (2012). Google's cross-dialect Arabic voice search. In *Proc ICASSP*, pages 4441–4444. IEEE.

Billa, J., Noamany, M., Srivastava, A., Liu, D., Stone, R., Xu, J., Makhoul, J., and Kubala, F. (2002). Audio indexing of Arabic broadcast news. In *Proc ICASSP*, pages I5– I8.

Bilmes, J. and Kirchhoff, K. (2003). Factored language models and generalized parallel backoff. In *Proc NAACL HTL*, volume 2, pages 4–6.

Bisani, M. and Ney, H. (2002). Investigations on joint-multigram models for grapheme-to-phoneme conversion. In *Proc INTERSPEECH*, pages 105–108, Denver, CO.

Bisani, M. and Ney, H. (2008). Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication*, 50(5):434–451.

Bourlard, H. A. and Morgan, N. (1993). *Connectionist speech recognition: a hybrid approach*. Kluwer Academic Publishers, Norwell, MA.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and regression trees*. Wadsworth International Group, Belmont, CA.

Brown, P. F., deSouza, P. V., Mercer, R. L., Pietra, V. J. D., and Lai, J. C. (1992). Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

Buckwalter, T. (2002). Buckwalter Arabic Morphological Analyzer Version 1.0 (LDC2002L49).

Buckwalter, T. (2004a). Buckwalter Arabic Morphological Analyzer Version 2.0 (LDC2004L02).

Buckwalter, T. (2004b). Issues in Arabic orthography and morphology analysis. In *Proc Workshop on Computational Approaches to Arabic Script-based Languages*, pages 31–34, Geneva, Switzerland.

Byrne, W., Hajič, J., Ircing, P., Krbec, P., and Psutka, J. (2000). Morpheme based language models for speech recognition of Czech. In Ivan Habernal, V. M., editor, *Text, speech and dialogue*, pages 211–216. Springer.

Carki, K., Geutner, P., and Schultz, T. (2000). Turkish LVCSR: towards better speech recognition for agglutinative languages. In *Proc ICASSP*, volume 3, pages 1563–1566. IEEE.

Chen, S. F. et al. (2003). Conditional and joint models for grapheme-to-phoneme conversion. In *Proc INTERSPEECH*, pages 2033–2036, Geneva, Switzerland.

Chen, S. F. and Goodman, J. (1996). An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 310–318. Association for Computational Linguistics.

Chiang, D., Diab, M., Habash, N., Rambow, O., and Shareef, S. (2005). Parsing Arabic dialects. In *Final Report, 2005 Johns Hopkins University Summer Workshop*.

Choueiter, G., Povey, D., Chen, S., and Zweig, G. (2006). Morpheme-based language modeling for Arabic LVCSR. In *Proc ICASSP*, pages 1053–1056.

Cieri, C., Graff, D., Kimball, O., Miller, D., and Walker, K. (2004a). Fisher English Training Speech Part 1 Speech (LDC2004S13).

Cieri, C., Graff, D., Kimball, O., Miller, D., and Walker, K. (2004b). Fisher English Training Speech Part 1 Transcripts (LDC2004T19).

Coulmas, F. (1996). *The Blackwell encyclopedia of writing systems*. Oxford : Blackwell.

Creutz, M., Hirsimäki, T., and Kurimo, M. (2007a). Analysis of morph-based speech recognition and the modeling of out-of-vocabulary words across languages. In *Proc NAACL HTL*, pages 380–387.

Creutz, M., Hirsimäki, T., Kurimo, M., Puurula, A., Pylkkönen, J., Siivola, V., Varjokallio, M., Arisoy, E., Saraçlar, M., and Stolcke, A. (2007b). Morph-based speech recognition and modeling of out-of-vocabulary words across languages. *ACM Transactions on Speech and Language Processing (TSLP)*, 5(1).

Creutz, M. and Lagus, K. (2005). *Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor 1.0*.

Dahl, G., Yu, D., Deng, L., and Acero, A. (2012). Context-dependent pre-trained deep neural networks for large vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):30–42.

Darwish, K. (2002). Building a shallow Arabic morphological analyzer in one day. In *Proc ACL Workshop on Computational Approaches to Semitic Languages*, pages 1–8. Association for Computational Linguistics.

Davis, S. B. and Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 28(4):357–366.

Dedina, M. J. and Nusbaum, H. C. (1991). PRONOUNCE: a program for pronunciation by analogy. *Computer speech & language*, 5(1):55–64.

Deligne, S., Yvon, F., and Bimbot, F. (1995). Variable-length sequence matching for phonetic transcription using joint multigrams. In *Proc EUROSPEECH*, pages 2243–2246, Madrid, Spain.

Deng, Y. and Khudanpur, S. (2003). Latent semantic information in maximum entropy language models for conversational speech recognition. In *Proc NAACL HLT*, volume 1, pages 56–63. Association for Computational Linguistics.

Diab, M., Ghoneim, M., and Habash, N. (2007). Arabic diacritization in the context of statistical machine translation. In *Proceedings of MT-Summit*.

Diab, M., Habash, N., Rambow, O., Altantawy, M., and Benajiba, Y. (2010). COLABA: Arabic dialect annotation and processing. In *Proc LREC Workshop on Semitic Language Processing*, pages 66–74.

Diehl, F., Gales, M., Tomalin, M., and Woodland, P. (2008). Phonetic pronunciations for Arabic speech-to-text systems. In *Proc ICASSP*, pages 1573–1576.

Diehl, F., Gales, M. J., Tomalin, M., and Woodland, P. C. (2009a). Morphological analysis and decomposition for Arabic speech-to-text systems. In *Proc INTERSPEECH*, pages 2675–2678.

Diehl, F., Gales, M. J. F., Tomalin, M., and Woodland, P. C. (2009b). Morphological analysis and decomposition for Arabic speech-to-text systems. In *Proc INTERSPEECH*, pages 2675–2678. ISCA.

Dietterich, T. G. and Bakiri, G. (1995). Solving multiclass learning problems via error-correcting output codes. *arXiv preprint cs/9501101*.

El-Desoky, A., Gollan, C., Rybach, D., Schlüter, R., and Ney, H. (2009). Investigating the use of morphological decomposition and diacritization for improving Arabic LVCSR. In *Proc INTERSPEECH*, pages 2679–2682. ISCA.

El-Desoky, A., Schlüter, R., and Ney, H. (2010). A hybrid morphologically decomposed factored language models for Arabic LVCSR. In *Proc NAACL HTL*, pages 701–704. Association for Computational Linguistics.

El-Desoky, A., Schluter, R., and Ney, H. (2012). Investigations on the use of morpheme level features in language models for Arabic LVCSR. In *Proc ICASSP*, pages 5021–5024. IEEE.

El-Desoky, A. M., Kuo, H.-K., Mangu, L., and Soltau, H. (2013). Morpheme-based feature-rich language models using deep neural networks for LVCSR of Egyptian Arabic. In *Proc ICASSP*, pages 8435–8439. IEEE.

El-Imam, Y. A. (2004). Phonetization of Arabic: rules and algorithms. *Computer Speech & Language*, 18(4):339–373.

El-Sadany, T. A. and Hashish, M. A. (1988). Semi-automatic vowelization of Arabic verbs. In *10th NC conference*.

El-Sadany, T. A. and Hashish, M. A. (1989). An Arabic morphological system. *IBM Systems Journal*, 28(4):600–612.

Elfardy, H. and Diab, M. T. (2012). Simplified guidelines for the creation of large scale dialectal Arabic annotations. In *Proc LREC*, pages 371–378.

Elmahdy, M., Gruhn, R., Minker, W., and Abdennadher, S. (2010). Cross-lingual acoustic modeling for dialectal Arabic speech recognition. In *Proc INTERSPEECH*.

Elmahdy, M., Hasegawa-Johnson, M., and Mustafawi, E. (2013). A transfer learning approach for under-resourced Arabic dialects speech recognition. In *Workshop on Less Resourced Languages, new technologies, new challenges and opportunities (LTC 2013)*.

Elshafei, M., Al-Muhtaseb, H., Al-Ghamdi, M., et al. (2006). Machine generation of Arabic diacritical marks. *MLMTA*, 2006:128–133.

Emam, O. and Fischer, V. (2004). Hierarchical approach for the statistical vowelization of Arabic text. US Patent App. 10/948,443.

Emami, A. and Jelinek, F. (2005). Random clusterings for language modeling. In *Proc ICASSP*, volume 1, pages 581–584. IEEE.

Farghaly, A. and Shaalan, K. (2009). Arabic Natural Language Processing: Challenges and Solutions. *ACM Transactions on Asian Language Information Processing*, 8(4).

Ferguson, C. A. (1959). Diglossia. *Word*, 15:325–40.

Forney, G. (1973). The Viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278.

Gal, Y. (2002). An HMM approach to vowel restoration in Arabic and Hebrew. In *Proc ACL Workshop on Computational Approaches to Semitic Languages*, pages 1–7. Association for Computational Linguistics.

Gales, M., Diehl, F., Raut, C., Tomalin, M., Woodland, P., and Yu, K. (2007). Development of a phonetic system for large vocabulary Arabic speech recognition. In *Proc ASRU*, pages 24–29.

Galescu, L. and Allen, J. F. (2001). Bi-directional conversion between graphemes and phonemes using a joint n-gram model. In *4th ISCA Tutorial and Research Workshop (ITRW) on Speech Synthesis*.

Geutner, P. (1995). Using morphology towards better large-vocabulary speech recognition systems. In *Proc ICASSP*, volume 1, pages 445–448. IEEE.

Ghaoui, A., Yvon, F., Mokbel, C., and Chollet, G. (2005). On the use of morphological constraints in n-gram statistical language model. In *Proc INTERSPEECH*.

Goldsmith, J. (2001). Unsupervised learning of the morphology of a natural language. *Computational linguistics*, 27(2):153–198.

Habash, N. and Rambow, O. (2005). Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In *ACL*, pages 573–580.

Habash, N. and Rambow, O. (2006). MAGEAD: a morphological analyzer and generator for the Arabic dialects. *Proc ACL*.

Habash, N. and Rambow, O. (2007). Arabic diacritization through full morphological tagging. In *Proc NAACL HTL*, volume of short papers, pages 53–56. Association for Computational Linguistics.

Habash, N., Soudi, A., and Buckwalter, T. (2007). On Arabic transliteration. *Arabic Computational Morphology*, pages 15–22.

Hain, T. (2002). Implicit pronunciation modelling in ASR. *ISCA Tutorial and Research Workshop on Pronunciation Modeling and Lexicon Adaptation for Spoken Language Technology*.

Hammarström, H. and Borin, L. (2011). Unsupervised learning of morphology. *Computational Linguistics*, 37(2):309–350.

Hammo, B., Sleit, A., and El-Haj, M. (2008). Enhancing retrieval effectiveness of diacritisized Arabic passages using stemmer and thesaurus. In *The 19th Midwest Artificial Intelligence and Cognitive Science Conference (MAICS2008)*.

Harris, Z. (1954). Distributional structure. *Word*, 10(23):146–162.

Hartmann, W., Roy, A., Lamel, L., and Gauvain, J.-L. (2013). Acoustic unit discovery and pronunciation generation from a grapheme-based lexicon. In *Proc ASRU*, pages 380–385. IEEE.

Hattab, A. M. and Hussain, A. K. (2012). Hybrid statistical and morpho-syntactical Arabic language diacritizing system. *International Journal of Academic Research*, 4(4).

Hawkins, P. (1983). Diglossia revisited. *Language sciences*.

Heddaya, A. (1985). Qalam: A convention for morphological Arabic-Latin-Arabic transliteration. *URL http://eserver. org/langs/qalam. txt*.

Hermansky, H. (1989). Perceptual linear predictive (PLP) analysis of speech. *Journal Acoustical Society of America*, 87(4):1738–1752.

Hermansky, H., Ellis, D., and Sharma, S. (2000). Tandem connectionist feature extraction for conventional HMM systems. In *Proc ICASSP*, volume 3, pages 1635–1638.

Hestenes, M. R. and Stiefel, E. (1952). Methods of conjugate gradients for solving linear systems. *Research of the National Bureau of Standards*, 49(6).

Hifny, Y. (2012a). Higher order n-gram language models for Arabic diacritics restoration. In *Proceedings of the 12th Conference on Language Engineering (ESOLEC'12), Cairo, Egypt*.

Hifny, Y. (2012b). Smoothing techniques for Arabic diacritics restoration. In *Proceedings of the 12th Conference on Language Engineering (ESOLEC'12), Cairo, Egypt*.

Hifny, Y. (2013). Restoration of Arabic diacritics using dynamic programming. In *Computer Engineering & Systems (ICCES), 2013 8th International Conference on*, pages 3–8. IEEE.

Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T., and Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29(6):82–97.

Hirsimäki, T., Creutz, M., Siivola, V., Kurimo, M., Virpioja, S., and Pylkkönen, J. (2006). Unlimited vocabulary speech recognition with morph language models applied to finnish. *Computer Speech & Language*, 20(4):515–541.

Holes, C. (2004). *Modern Arabic: structures, functions, and varieties*. Georgetown University Press.

Ibn Khaldoun, A. (2001). *Al-Muqaddimah*. Dar Al-Fikr, Beirut.

IPA (1999). *Handbook of the International Phonetic Association*. Cambridge University Press, Cambridge.

Ircing, P., Krbec, P., Hajic, J., Psutka, J., Khudanpur, S., Jelinek, F., and Byrne, W. (2001). On large vocabulary continuous speech recognition of highly inflectional language–Czech. In *Proc INTERSPEECH*.

Jelinek, F. (1976). Continuous speech recognition by statistical methods. In *Proc IEEE*, volume 64, pages 523–556.

Jelinek, F. (1980). Interpolated estimation of markov source parameters from sparse data. *Pattern recognition in practice*.

Jelinek, F. (1997). *Statistical Methods for Speech Recognition*. MIT Press.

Kampa, K., Hasanbelliu, E., and Principe, J. C. (2011). Closed-form cauchy-schwarz PDF divergence for mixture of Gaussians. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 2578–2585. IEEE.

Kawahara, T., Lee, A., Kobayashi, T., Takeda, K., Minematsu, N., Sagayama, S., Itou, K., Ito, A., Yamamoto, M., Yamada, A., et al. (2000). Free software toolkit for Japanese large vocabulary continuous speech recognition. In *Proc ICSLP*, volume 4, pages 476–479. IEEE.

Kiecza, D., Schultz, T., and Waibel, A. (1999). Data-driven determination of appropriate dictionary units for Korean LVCSR. In *Proc ICASSP*, pages 323–327.

Kilany, Hanaa, Gadalla, H., Arram, H., Yacoub, A., El-Habashi, A., and McLemore, C. (2002). Egyptian Colloquial Arabic Lexicon (LDC99L22).

Killer, M., Stüker, S., and Schultz, T. (2003). Grapheme based speech recognition. In *Proc EUROSPEECH*, pages 3141–3144.

King, B. (1967). Step-wise clustering procedures. *Journal of the American Statistical Association*, 62(317):86–101.

Kirchhoff, K., Bilmes, J., Das, S., Duta, N., Egan, M., Ji, G., He, F., Henderson, J., Liu, D., and Noamany, M. (2003). Novel approaches to Arabic speech recognition: report from the 2002 Johns-Hopkins summer workshop. In *Proc ICASSP*, volume 1, pages 344–347.

Kirchhoff, K., Bilmes, J., Henderson, J., and Schwartz, R. (2002a). Novel speech recognition models for Arabic. *… Workshop*.

Kirchhoff, K., Bilmes, J., Henderson, J., Schwartz, R., Noamany, M., Schone, P., Ji, G., Das, S., Egan, M., He, F., Vergyri, D., Liu, D., and Duta, N. (2002b). Novel speech recognition models for Arabic. Technical report, Johns Hopkins University.

Kirchhoff, K. and Vergyri, D. (2004). Cross-dialectal acoustic data sharing for Arabic speech recognition. In *Proc ICASSP*, volume 1, pages 765–768.

Kirchhoff, K. and Vergyri, D. (2005). Cross-dialectal data sharing for acoustic modeling in Arabic speech recognition. *Speech communication*, 46(1):37–51.

Kirchhoff, K., Vergyri, D., Bilmes, J., Duh, K., and Stolcke, A. (2006). Morphology-based language modeling for conversational Arabic speech recognition. *Computer Speech & Language*, 20(4):589–608.

Kittur, A., Chi, E. H., and Suh, B. (2008). Crowdsourcing user studies with Mechanical Turk. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 453–456. ACM.

Kneissler, J. and Klakow, D. (2001). Speech recognition for huge vocabularies by using optimized sub-word units. In *Proc INTERSPEECH*, pages 69–72.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al. (2007). Moses: Open source toolkit for statistical machine translation. In *Proc ACL*, pages 177–180. Association for Computational Linguistics.

Kschischang, F. R., Frey, B. J., and Loeliger, H.-A. (2001). Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519.

Kumar, G., Cao, Y., Cotterell, R., Callison-Burch, C., Povey, D., and Khudanpur, S. (2014). Translations of the CALLHOME Egyptian Arabic corpus for conversational speech translation. In *IWSLT*.

Kuo, H.-K., Mangu, L., Emami, A., Zitouni, I., and Lee, Y.-S. (2009). Syntactic features for Arabic speech recognition. In *Proc ASRU*, pages 327–332.

Kuo, H.-K. J., Mangu, L., Emami, A., and Zitouni, I. (2010). Morphological and syntactic features for Arabic speech recognition. In *Proc ICASSP*, pages 5190–5193. IEEE.

Kwon, O.-W. (2000). Performance of LVCSR with morpheme-based and syllable-based recognition units. In *Proc ICASSP*, volume 3, pages 1567–1570. IEEE.

Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data, 2001. In *Proc. ICML*.

Lagally, K. (1992). ArabTEX, a system for typesetting Arabic. In *3rd Int Conf Multilingual Computing: Arabic and Roman Script*, volume 9.

Lamel, L., Messaoudi, A., and Gauvain, J.-L. (2008). Investigating morphological decomposition for transcription of Arabic broadcast news and broadcast conversation data. In *Proc INTERSPEECH*, pages 1429–1432. ISCA.

Lamel, L., Messaoudi, A., and Gauvain, J.-L. G. (2007). Improved acoustic modeling for transcribing Arabic broadcast data. In *Proc INTERSPEECH*, pages 2077–2080.

LaRocca, S. and Chouairi, R. (2002). West Point Arabic Speech (LDC2002S02).

Larson, M., Willett, D., Köhler, J., and Rigoll, G. (2000). Compound splitting and lexical unit recombination for improved performance of a speech recognition system for German parliamentary speeches. In *Proc INTERSPEECH*, pages 945–948.

Lee, Y., Papineni, K., Roukos, S., Emam, O., and Hassan, H. (2003). Language model based Arabic word segmentation. *Proc ACL*, pages 399–406.

Lewis, M. P., editor (2009). *Ethnologue: Languages of the World*. SIL International, Dallas, Tex, sixteenth edition edition.

Liu, X., Gales, M. J., and Woodland, P. C. (2012). Paraphrastic language models. In *Proc INTERSPEECH*.

Liu, X., Gales, M. J., and Woodland, P. C. (2014). Paraphrastic language models. *Computer Speech & Language*, 28(6):1298–1316.

Lowerre, B. T. (1976). *The Harpy Speech Recognition System.* PhD thesis, Pittsburgh, PA.

Maamouri, M., Bies, A., Buckwalter, T., and Mekki, W. (2004). The Penn Arabic Treebank: Building a large-scale annotated arabic corpus. In *NEMLAR conference on Arabic language resources and tools*, pages 102–109.

Maamouri, M., Buckwalter, T., Graff, D., and Jin, H. (2006). Levantine Arabic QT Training Data Set 5 (LDC2006S29).

Maamouri, M., Buckwalter, T., Graff, D., and Jin, H. (2007). Fisher Levantine Arabic Conversational Telephone Speech, Transcripts (LDC2007T04).

Maisel, S. and Shoup, J. A. (2009). *Saudi Arabia and the Gulf Arab states today: an encyclopedia of life in the Arab states*. Greenwood.

Manzur, Y. a.-D. M. I. (1883). *Lisan al-arab*. Al Dar al-Misriyya Li-l-ta'lif wa-l-tarhim.

Mohri, M. (1997). Finite-state transducers in language and speech processing. *Computational Linguistics*, 23(2):269–312.

Murtagh, F. (1983). A survey of recent advances in hierarchical clustering algorithms. *The Computer Journal*, 26(4):354–359.

Nelken, R. and Shieber, S. (2005). Arabic diacritization using weighted finite-state transducers. In *Proc ACL Workshop on Computational Approaches to Semitic Languages*, pages 79–86.

Ney, H. (1990). Acoustic modeling of phoneme units for continuous speech recognition. In Torres, L. and Masgrau, E., editors, *Proceedings of Eusipco-90*, volume 1, pages 65–72, Barcelona, Spain. Elsevier Science Publishers B. V.

Nguyen, L., Ng, T., Nguyen, K., Zbib, R., and Makhoul, J. (2009). Lexical and phonetic modeling for Arabic automatic speech recognition. In *Proc INTERSPEECH*, pages 712–715. ISCA.

NIST (2009). The history of automatic speech recognition evaluations at NIST. Technical report.

Nocedal, J. (1980). Updating quasi-newton matrices with limited storage. *Mathematics of computation*, 35(151):773–782.

Norris, J. R. (1998). *Markov chains*. Number 2. Cambridge university press.

Novotney, S. and Callison-Burch, C. (2010). Cheap, fast and good enough: Automatic speech recognition with non-expert transcription. In *Proc NAACL HLT*, pages 207–215. Association for Computational Linguistics.

Odell, J. and Woodland, P. (1994). Tree-based state clustering for large vocabulary speech recognition. In *International Symposium on Speech, Image Processing and Neural Networks*, volume 2, pages 690–693.

Olive, J., Christianson, C., and McCary, J., editors (2011). *Handbook of Natural Language Processing and Machine Translation*. SpringerLink : Bücher. Springer New York.

Paolacci, G., Chandler, J., and Ipeirotis, P. G. (2010). Running experiments on Amazon Mechanical Turk. *Judgment and Decision making*, 5(5):411–419.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: a method for automatic evaluation of machine translation. In *Proc ACL*, pages 311–318. Association for Computational Linguistics.

Parris, E. S. and Carey, M. J. (1996). Language independent gender identification. In *Proc ICASSP*, volume 2, pages 685–688. IEEE.

Parveen, S. and Green, P. D. (2002). Speech recognition with missing data using recurrent neural nets. In *Advances in neural information processing systems*, volume 2, pages 1189–1194. MIT Press.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible reasoning*. Morgan Kaufmann Publishers, Los Altos.

Phan, X.H. Nguyen, L. (2005). Flexcrfs: Flexible conditional random field toolkit.

Rashwan, M., Al-Badrashiny, M., Attia, M., and Abdou, S. (2009). A hybrid system for automatic Arabic diacritization. In *The 2nd International Conference on Arabic Language Resources and Tools*.

Rashwan, M. A., Al-Badrashiny, M. A. S., Attia, M., Abdou, S. M., and Rafea, A. (2011). A stochastic Arabic diacritizer based on a hybrid of factorized and unfactorized textual features. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(1):166–175.

Richardson, F., Ostendorf, M., and Rohlicek, J. (1995). Lattice-based search strategies for large vocabulary speech recognition. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, volume 1, pages 576–579. IEEE.

Rosenfeld, R. (1996). A maximum entropy approach to adaptive statistical language modelling. *Computer Speech & Language*, 10(3):187–228.

Rosenfield, R. (1995). Optimizing lexical and n-gram coverage via judicious use of linguistic data. In *Proc EUROSPEECH*, page 1763–1766.

Sabou, M., Bontcheva, K., Derczynski, L., and Scharl, A. (2014). Corpus annotation through crowdsourcing: Towards best practice guidelines. In *Proc LREC*, pages 859–866.

Sahraeian, S. M. E. and Yoon, B.-J. (2011). A novel low-complexity HMM similarity measure. *Signal Processing Letters, IEEE*, 18(2):87–90.

Salloum, W. and Habash, N. (2011). Dialectal to standard Arabic paraphrasing to improve Arabic-English statistical machine translation. In *the First Workshop on Algorithms and*

Resources for Modelling of Dialects and Language Varieties, pages 10–21, Edinburgh, Scotland.

Salloum, W. and Habash, N. (2013). Dialectal Arabic to English machine translation: pivoting through Modern Standard Arabic. In *Proc NAACL HTL*, pages 348–358.

Sarikaya, R., Afify, M., and Gao, Y. (2007). Joint morphological-lexical language modeling (JMLLM) for Arabic. In *Proc ICASSP*, volume 4.

Sarikaya, R., Afify, M., and Kingsbury, B. (2009). Tied-mixture language modeling in continuous space. In *Proc NAACL HLT*, pages 459–467.

Sawaf, H. (2010). Arabic dialect handling in hybrid machine translation. In *Proc Conf Assoc Machine Translation in the Americas (AMTA)*, Denver, CO.

Schlippe, T., Nguyen, T., and Vogel, S. (2008). Diacritization as a machine translation problem and as a sequence labeling problem. In *8th AMTA conference, Hawaii*, pages 21–25.

Schlippe, T., Ochs, S., and Schultz, T. (2012). Grapheme-to-phoneme model generation for Indo-European languages. In *Proc ICASSP*, pages 4801–4804. IEEE.

Schuller, B., Steidl, S., Batliner, A., Burkhardt, F., Devillers, L., MüLler, C., and Narayanan, S. (2013). Paralinguistics in speech and language – state-of-the-art and the challenge. *Computer Speech & Language*, 27(1):4–39.

Schwartz, R. and Austin, S. (1991). A comparison of several approximate algorithms for finding multiple (n-best) sentence hypotheses. In *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on*, pages 701–704. IEEE.

Sejnowski, T. J. and Rosenberg, C. R. (1988). *NETtalk: A parallel network that learns to read aloud*. MIT Press.

Shaalan, K., Abo Bakr, H. M., and Ziedan, I. (2009). A hybrid approach for building Arabic diacritizer. In *Proc EACL Workshop on Computational Approaches to Semitic Languages*, pages 27–35. Association for Computational Linguistics.

Sibawayh, A. I. U. (1983). *Kitab Sibawayh*. Alam Al-Kutub, Beirut.

Sneath, P. H., Sokal, R. R., et al. (1973). *Numerical taxonomy. The principles and practice of numerical classification.*

Snow, R., O'Connor, B., Jurafsky, D., and Ng, A. Y. (2008). Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings*

of the conference on empirical methods in natural language processing, pages 254–263. Association for Computational Linguistics.

Soltau, H., Mangu, L., and Biadsy, F. (2011). From Modern Standard Arabic to Levantine ASR: leveraging GALE for dialects. In Nahamoo, D. and Picheny, M., editors, *Proc ASRU*, pages 266–271. IEEE.

Soltau, H., Saon, G., Kingsbury, B., Kuo, H.-K., Mangu, L., Povey, D., and Emami, A. (2009). Advances in Arabic speech transcription at IBM under the DARPA GALE program. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(5):884–894.

Sorokin, A. and Forsyth, D. (2008). Utility data annotation with Amazon Mechanical Turk. *Urbana*, 51(61):820.

Stolcke, A. (2002). SRILM – an extensible language modeling toolkit. In *Proc INTERSPEECH*, pages 901–904.

Stolcke, A., Chen, B., Franco, H., Gadde, V. R. R., Graciarena, M., Hwang, M.-Y., Kirchhoff, K., Mandal, A., Morgan, N., Lei, X., Ng, T., Ostendorf, M., Sonmez, K., Venkataraman, A., Vergyri, D., Wang, W., Zheng, J., and Zhu, Q. (2006). Recent innovations in speech-to-text transcription at SRI-ICSI-UW. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5):1729–1744.

Taylor, P. (2005). Hidden Markov models for grapheme to phoneme conversion. In *Proc INTERSPEECH*, pages 1973–1976.

The Unicode Consortium (2011). The Unicode Standard. Technical Report Version 6.0.0, Unicode Consortium, Mountain View, CA.

Tsakalidis, S., Prasad, R., and Natarajan, P. (2009). Context-dependent pronunciation modeling for Iraqi ASR. In *Proc ICASSP*, pages 4457–4460.

Valverde-Albacete, F. J. and Peláez-Moreno, C. (2014). 100% classification accuracy considered harmful: the normalized information transfer factor explains the accuracy paradox. *PloS one*, 9(1):e84217.

Van Mol, M. (2003). *Variation in Modern Standard Arabic in radio news broadcasts, a synchronic descriptive investigation into the use of complementary particles*. Peeters.

Vazirnezhad, B., Almasganj, F., and Bijankhan, M. (2005). A hybrid statistical model to generate pronunciation variants of words. In *Natural Language Processing and Knowledge Engineering, 2005. IEEE NLP-KE'05. Proceedings of 2005 IEEE International Conference on*, pages 106–110. IEEE.

Vergyri, D. and Kirchhoff, K. (2004). Automatic diacritization of Arabic for acoustic modeling in speech recognition. In *Proc. the Workshop on Computational Approaches to Arabic Script-based Languages*, pages 66–73.

Vergyri, D., Kirchhoff, K., Gadde, R., Stolcke, A., and Zheng, J. (2005). Development of a conversational telephone speech recognizer for Levantine Arabic. In *Proc INTERSPEECH*, pages 1613–1616.

Vergyri, D., Mandal, A., Wang, W., Stolcke, A., Zheng, J., Graciarena, M., Rybach, D., Gollan, C., Schlüter, R., Kirchhoff, K., et al. (2008). Development of the SRI/nightingale Arabic ASR system. In *Proc INTERSPEECH*, pages 1437–1440.

Virpioja, S., Smit, P., Grönroos, S.-A., Kurimo, M., et al. (2013). Morfessor 2.0: Python implementation and extensions for Morfessor baseline.

Vozila, P., Adams, J., Lobacheva, Y., and Thomas, R. (2003). Grapheme to phoneme conversion and dictionary verification using graphonemes. In *Proc INTERSPEECH*.

Ward Jr, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244.

Watson, J. C. E. (2002). *The phonology and morphology of Arabic*. Oxford University Press, New York.

Weninger, F., Geiger, J., Wöllmer, M., Schuller, B., and Rigol, G. (2011). The Munich 2011 CHiME Challenge contribution: NMF-BLSTM speech enhancement and recognition for reverberated multisource environments. In *CHiME 2011 Workshop on Machine Listening in Multisource Environments*, pages 24–29.

Xanthos, A., Hu, Y., and Goldsmith, J. (2006). Exploring variant definitions of pointer length in MDL. In *Proceedings of the Eighth Meeting of the ACL Special Interest Group on Computational Phonology and Morphology*, pages 32–40. Association for Computational Linguistics.

Xiang, B., Nguyen, K., Nguyen, L., Schwartz, R., and Makhoul, J. (2006). Morphological decomposition for Arabic broadcast news transcription. In *Proc ICASSP*, volume 1, pages I–I, Toulouse, France.

Xu, P. and Jelinek, F. (2004). Random forests in language modeling. In *Proceedings of EMNLP*, volume 4, pages 325–332.

Young, S. J., Evermann, G., Gales, M. J. F., Hain, T., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., and Woodland, P. C. (2006). *The HTK Book, version 3.4*. Cambridge University Engineering Department, Cambridge, UK.

Young, S. J. and Woodland, P. C. (1994). State clustering in hidden Markov model-based continuous speech recognition. *Computer Speech & Language*, 8(4):369–383.

Yvon, F. (1996). Grapheme-to-phoneme conversion using multiple unbounded overlapping chunks. In *Proce NeMLaP II*, pages 218–228, Ankara, Turkey.

Zaidan, O. F. and Callison-Burch, C. (2011). Crowdsourcing translation: Professional quality from non-professionals. In *Proc 49th Annual Meeting Assoc Computational Linguistics*, volume 1 - Human language technologies, pages 1220–1229. Association for Computational Linguistics.

Zhang, W., Wang, X., Zhao, D., and Tang, X. (2012). Graph degree linkage: Agglomerative clustering on a directed graph. In *Computer Vision–ECCV 2012*, pages 428–441. Springer.

Zitouni, I., Sorensen, J., and Sarikaya, R. (2006). Maximum entropy based restoration of Arabic diacritics. *ANNUAL MEETING- ….*