Arabic Language resources in HIAST

Oumayma Al-Dakkak, Nada Ghneim, Afaf Alshalaby, Riad Sonbol, Mhd. Said Desouki

Higher Institue for Applied Sciences and Technology (HIAST)
P.o. Box 31983, Damascus
SYRIA
odakkak@hiast.edu.sy
nada.ghneim@hiast.edu.sy
afaf.shalaby@hiast.edu.sy
riad.sonbol@hiast.edu.sy
said.desouki@hiast.edu.sy

Abstract

Arabic Language Processing is gaining increasing importance all around the world. This language is spoken by nearly 300 millions in the Arab World, and is an interesting language for the 1.3 billion Muslims. Arabic is becoming also a focal point of interest in many universities all around the world. The positive aspect of this interest in Arabic -which was relatively ignored when compared with other live languages-, is the intensive work on language technologies, and an increasing amount of digital contents on the Internet.
Arabic Language is one of the most promoted research axis in HIAST since its foundation in 1983 (Ghneim & Al-DAkkak, 2006). In the introduction of this paper, we mention the most relevant works, in both speech and text aspects, which can be of interest in many applications, and can be a subject of resource sharing. In the second section we detail our acoustic database of semi-syllables and in the third one we develop our morphological analyzer.

1-Introduction

HIAST (Higher Institute for Applied Sciences and Technology) was founded in 1983 in the aim of forming high qualified engineers and researchers. Early headed by Dr A. W. Shaheed a member of Arabic Academy, then by Dr M. Mrayati a regional advisor in science and technology; Arabic language processing emerged as an interesting research axis in the Institute. A group of Arabic linguists and Information Technology researchers worked and is still working on the issue. Our efforts in textual processing included a number of projects from which we mention the following:

- The Arabic morphology system (Bawab et al., 1984). The derivation system is based on a dictionary of 5588 trilateral-root verbs, 1932 quadrilateral-root verbs, 11790 of non standard infinitives. In fact standard infinitives are derived by the system. The enumeration of the basic roots in the above dictionary was also done in HIAST previously.
- A database incorporating 1200 selected extraction from poetry and prose (Bakeer, 2005), with the syntactic classification (I'IRAB), of words and phrases. The database covers all the syntactic classification in the Arabic language.
- Arabic lexical database gathering the information of "Al Wasseet dictionary", in a structured form, using Microsoft Access software. The database is formed of sixteen tables, covering the morphological categories in Arabic and related information: verbs, nouns, infinitives, plural of nouns, particles, special combinations (idioms), examples of use …etc. The total number of the studied items is about 200 thousands (Attar et al., 2007).
- Arabic Morphological analyzer: a new approach for Arabic root extraction (Sonbol et al., 2008), in which different levels of reliability and performance is provided to support the needs of different applications.
- Arabic Optical Character Recognition system, capable of recognizing various typewriting fonts. The project was sponsored by UNESCO.

In addition, there are ongoing works, concerning Arabic Part-Of-Speech tagger, automatic vocalization (Safadi et al., 2006), a construction of a database of more than 4000 Arabic famous words combinations ("tarakeeb") and a prototype of a web based Arabic interactive dictionary.
Concerning speech processing, our efforts focused on speech synthesis, speech and speaker identification and verification. We mention hereafter the principal projects:

- Emotional audio-visual text-to-speech (TTS) system for Arabic Language (Al-Dakkak et al., 2005; Abou-Zliekha et al., 2006). The system is based on two entities: an emotional audio text-to-speech system, which generates speech depending on the input text and the desired emotion type, and an emotional visual model which generates the talking heads, by forming the corresponding visemes. The phonemes to visemes mapping, and the emotion shaping use a 3-paramertic face model, based on the Abstract Muscle Model. We have thirteen viseme models and five emotions (Joy, sadness, fear, surprise and anger) as parameters to the face model. The TTS produces the speech corresponding to the input text with the suitable prosody to include the prescribed emotion. In parallel, the system generates the visemes and sends the controls to the facial model to get the animation of the talking head in real time. An expert system performs the orthographic phonetic transcription

(Ghneim & Habash, 2003). A rough prosody is generated based on the punctuations (Al-Dakkak et al., 2006), and then modulated by the selected emotional type. The actual system generates speech using MBROLA Arabic diphones (Dutoit et al., 1996). However, we are building our own semi-syllables units for the synthesizer (see secion 2).

- In parallel, we work on Text normalization especially numbering, and on necessary tagging in view of building Arabic SSML system (Shaker et al., 2008).
- Several works on words recognition and speaker identification and verification have been also undertaken, based on Gaussian Mixture Modeling of Mel Cepstrum features. The system works in a quasi real time (Al-Marashli & Al-Dakkak, 2008).

In the following sections, we give more details about a speech application; the acoustic database of semi syllables, and a textual application; a morphological analyzer.

2- Acoustic Database of semi-syllables

Many high quality multilingual Text-to-Speech (TTS), don't support Arabic. Arabic TTS, whose demos are available on Internet are not accessible for development by researchers.

In this section, we present the recorded corpus, from which the semi-syllables are extracted, the segmentation of its logatoms and the incorporation of the extracted semi-syllables in an acoustic data-base.

2-1 The Corpus of the Semi-Syllables

With the objective of building a complete system of standard spoken Arabic, we defined the set of phonemes that must be present in the system:

- 28 consonants, which correspond to the letters of the alphabet, replacing the first letter by "hamza" and the last two letters by the semi vowels /w/ and /y/.
- 5 vowels which are: /a/, /u/, /i/ and the open /o/ and open /e/. In fact, most Arabic synthesizers do not take into account the presence of /o/ and /e/; though they do exist in the standard Arabic spoken in the Middle East countries. On the other hand, many synthesizers differentiate between short vowels and long vowels. We do not adopt this vision as the only difference between a short vowel and its corresponding long one is just in duration.
- The same 5 vowels in an emphatic version. The presence of an emphatic adjacent consonant changes the acoustic features of the vowels.
- Emphatic /l/ and /r/.

Table 1 shows the set of these phonemes, with their adopted transcription in our system.

As the Arabic syllables are only of 4 types: V, CV, CVC, CVCC, where V stands for vowels and C stands for consonants; the semi-syllables are of 5 types: #CV, VC#, VCC# (# is silence). Other combinations such as VCV and VCCV are also added; hence the logatoms from which those semi-syllables are extracted are respectively: Cvsasa, satVC, satVC$_1$C$_2$, tV$_1$CV$_2$sa, tV$_1$C$_1$C$_2$V$_2$sa (Chenfour et al., 2000), where the small letters are pronounced as they are, V, V$_1$, V$_2$ scans all the vowels and C, C$_1$, C$_2$ scan all the consonants. Some combinations never occur in the language, they are excluded.

This corpus is being recorded twice: a female, and a male voice. The total number of recorded logatoms is 10304 for each speaker.

2-2 Segmentation and Semi-Syllables Extraction

Once the corpus is recorded, we've proceeded to its segmentation to extract the semi syllables.

There are several possibilities for speech segmentation algorithms. The adopted algorithm on successive overlapped frames is the following:

- Calculating the Cepestrum LPC Coefficients.
- Calculating the cesptral acoustic distances vector, for every frame, by taking the cepstral distance between the current frame and the three previous frames, and between the current frame and the three following ones. Depending on the values of the distance vectors, the frame is in one of the three cases: either it is part of the preceding phoneme, or it is part of the following phoneme, or it is a separating frame between two phonemes.

Depending on the previous step, the boundaries of each phoneme are defined. In fact, the above algorithm permits to define the frames which are the most stable inside each phoneme too. These stables frames enable to define the cutting points in the vowels of the logatoms to extract the semi-syllables.

The stable frame inside a phoneme is the frame which has the minimal cepstral distance from the centroide of all the frames of that phoneme.

In order to ensure high quality of synthesis, the results of segmentation were checked by human expert, and corrected when needed.

2-1-3 Acoustic Database

We have created an acoustic database, containing the wave files of the extracted semi-syllables. Each record has the following fields:

- ID
- The path of the corresponding wave file, for the semi-syllables and for the logatoms from which the semi-syllables are extracted.
- The transcription of the semi-syllable in Latin.
- The transcription of the semi-syllable in Arabic.

Table 2 shows an example of some semi-syllables extracted from the acoustic database. In fact, we have fives tables, one for each type of semi-syllables mentioned above.

The present work has been tested on several sentences, using the semi-syllables as acoustic units. Compared to the quality of our old diphone based TTS, listeners approved the significant improvement in synthesized speech naturalness, even with rough prosody. The ongoing work now is towards high quality automatic prosody generation and the analysis of the acoustic database.

3- Application-Oriented Arabic Morphological Analyzer

Several approaches have been proposed for Arabic stemming; many papers survey and classify these techniques (Al-Sughaiyer et al., 2004; Larkey et al., 2001).

A promising approach to build a flexible and application oriented Arabic morphological analyzer has been proposed in HIAST. This approach is designed to satisfy various requirements of most applications which need morphological processing controlling a balance point between : performance, accuracy, and generality of solutions (i.e. getting all possible roots).

3-1 Algorithm

The algorithm has a number of steps (Sonbol et al., 2008):

Step1: Check if the word is a particle or a foreign word using a dictionary of particles and common foreign words.

Step2: Apply normalization steps.

Step3: Apply *Original Letters Detection Algorithm* which consists of two stages:

a. *Initialization:* an encoding process converting each letter to its initial "Morphological State" (see Table 3).

O	letter is surely part of the root. { ث، ج، ح، خ، د، ذ، ر، ز، ش، ص، ض، ط، ظ، ع، غ، ق}
A	letter is always considered as additional. {ة }
P	letter can only be added in prefix. { ب، ف، س، ل}
S	letter can only be added in suffix. {هـ }
T	letter can be added in both suffix or prefix. { ك، م، ن}
U	letter can be added anywhere in the word. { ت، و، ي، ا، أ}

Table 3 Morphological State for each letter

After encoding each letter by its initial morphological state code, we obtain an encoded word that can be more useful for morphological analysis. The root can be extracted directly in some cases, like when we have 3 Os (or more) in the encoded word, and in this case they represent root letters.

b. *Applying Transformation Rules:* taking into account the context of each letter in the word. The aim of these "*transformation rules*" is to move word's letters from its morphological state to a higher one with less ambiguity.

We use the following transformation Rules1:

R1) Change each 'P' after 'O' to 'O'.
R2) Change each 'S' before 'O' to 'O'.
R3) Change each 'P' after 'S' to 'O', and each 'S' before 'P' to 'O'.
R4) Change each 'T' before 'P' to 'P'.
R5) Change each 'T' before 'O' to 'P'.
R6) Change each 'T' after 'S' to 'S'.
R7) Change each 'T' after 'O' to 'S'.
R8) Change the first letter to 'P' if it is not 'O' or 'A'.
R9) Change the last letter to 'S' if it is not 'O' or 'A'.
R10) *Cutting Rule:* Let n_r the maximum length of the root, n_o the number of O letters in the encoded word, n_p the index of the first P letter, n_s the index of the last S letter, len the word length. Change each letter at index $i \in [0, n_p] \cup [n_s, len]$ to 'A' if $\min(|i - n_p|, |i - n_s|) \geq (n_r - n_o)$

Step4: Generate a bank of solutions which consist of each sequence of letters satisfying the following conditions:

a. Contains all Original letters (letters in the state O).
b. Does not contain any Additional letters (letters in the state A).
c. The pre-string (i.e. string of letters that are situated before the first root's letter in the word (Faa AL-Fel)) is valid. For example, in the word 'المكاتب" we consider 'المـ" as a pre-String, while the classical prefix is only 'الـ".
d. The suf-string (i.e. string of letters that are found after the last root's letter in the word (Lam AL-Fel)) is valid. We consider a string as a valid suf-string if it satisfies the following conditions:
   i. There is no letters in the state P.
   ii. If we have the letter Meem 'م' in the suf-string, it should be one of the following suf-string: {تم، كم، هم، ما}.
   iii. If we have the letter Taa Marbuta 'ة' in the suf-string, it should be one of the following suffixes: {ة ، ية، اتية ، انية}
   iv. If we have the letter Hamza in the suf-string, the previous letter of Hamza should be Alef 'ا".

Step5: Generate solutions that represent shadda case, elimination case, and non-trilateral roots **(optional step)**.

Step6: Correcting solutions in the bank of solution: we can make a balance between the three metric (reliability, performance, generality) by applying the next **optional steps**:

a. Pattern existence test using a list of available patterns.
b. Root existence test using a list of available roots.
c. Apply Ebdal and Ealal rules: we do this step only for invalid roots to check if it is invalid because of a special case.

To control the balance point (accuracy, performance, generality) we use the next two controlling techniques:

- Adding some parameters to control the different modules of the system, these parameters are: StopWord Test, ForignWords Test, RootExistance, Patterns Test, Ebdal Test, Ealal Test, Shadda Test, Eliminating Test, Satisfying the best solution, Root_Max_Length.
- Ranking the solutions by its "accuracy".

3-2 Evaluation

The evaluation of the system was performed using two different corpuses:

[1] Note that when using the words "before" and "after" in the transformation rules, we consider the direction of Arabic reading (right to left).

- The first is a list of word-root pairs (about 167000 pairs) extracted from HIAST Arabic lexical database (Al-Attar et al., 2007), and was used to evaluate the accuracy.
- The second is a collection of 585 Arabic articles (more than 375000 words) covering different fields (politics, economy, culture, science, and sport), and was used to evaluate the performance (speed of processing).

Using the last ten parameters we achieve different balance points, each balance point has certain characteristics which support the needs of a group of application.

Figure 1 shows the different balance points we achieved using our analyzer. The Horizontal axis represents the processing speed, while the vertical one represents the accuracy. Filled circles represent *Rn* states in which we try to find all possible solutions, where empty circles represent *Rn+* states in which we get the best solution. In this way we represent the accuracy, performance, and generality.

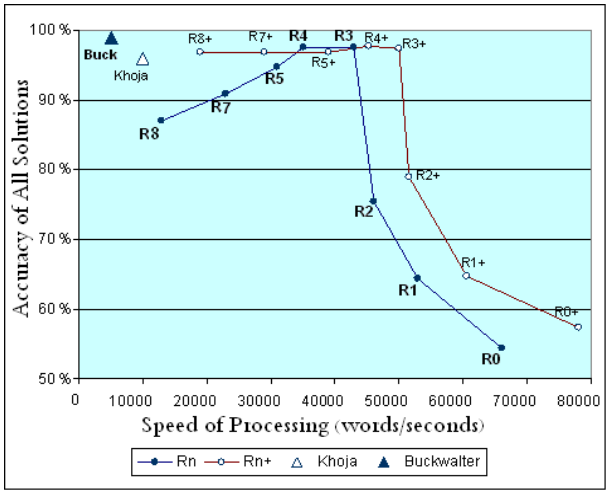

Figure 1 Evaluation of the system

We can see clearly that this approach provides different balance points which can support the needs of most applications. It provides states like (R0+, R1+, R2+, R3+, R4+) which have the advantage of high performance.

States R7+, R6+, R5+ are high performance and high accurate balance points. Their accuracy (about 97%) can be compared to high accurate rule-based stemmers like khoja one (Khoja & Garside, 1999), with better performance.

Rn states outperform Rn+ ones in generality. We did not notice much difference in accuracy in states R0, R1, R2, R3, R4 where the generality affects mainly the performance (as we do not cover solutions that represent elimination, shadda, and Ealal cases). These three cases are difficult in Arabic especially if we solve them without using a dictionary. To avoid this problem we provide two kinds of states:

- R5+, R6+, R7+ solve these problems without great effect on the performance, and still outperform other stemmers, because in these states we look for the best solution which is, generally, not one of the three difficult cases.
- In addition, we provide R5, R6, R7, R8 in which we try to include all right solutions even those representing elimination and shadda, which affects the accuracy. For example, the accuracy of R8 (where we solve all special cases) is about 87%. We can use these states for learning systems or lexical dictionaries where the stored data in these systems help to correct the result and raise the accuracy. In this case, we expect to reach both accuracy and generality close to Dictionary-based systems like Buckwalter's (Buckwalter 2002).

4- Conclusion

HIAST is a pioneer Institute for research and higher education. Arabic Natural Processing is one of its main research axes. Our projects seek to produce resources for Arabic, which can either be used to feed future researches or be incorporated in stand alone tools.

Acknowledgements

Authors are greatful to engineers: Dreresh. A., Mohammad **S.** and Mansour L. for their efforts in participating in the acoustic dictionary segmentation

Bibliographical References

Abou-Zliekha M., Al-Moubayed S., Al-Dakkak O., Ghneim N. (2006). Emotional Audio-Visual Arabic Text-to-Speech. Proceedings of EUSIPCO 2006, 4-8 September, Florence, Italy.

Al-Attar S., Bawab M. & Al-Dakkak O. (2007). Arabic lexical database. ANLP, ICTIS 2007, fes, Morocco.

Al-Dakkak O., Ghneim N., Abou-Zliekha M., Al-Moubayed S. (2006). Prosodic Feature Introduction and Emotion Incorporation in an Arabic TTS. Proceedings of ICTTA 2006, pp.1317-1322, April, Damascus, SYRIA.

Al-Dakkak O., Ghneim N., Abou-Zliekha, & Al-Moubayed S. (2005). Emotion Inclusion in an Arabic Text-to-Speech. Proceedings of EUSIPCO 2005, 4-8 September, Antalya, Turkey.

Al-Marashli A.& Al-Dakkak O. (2008). Automatic speaker independent speaker identification and verification system using mel cepstrum and GMM. Proceedings of ICTTA 2008, Damascus, SYRIA.

Al-Sughaiyer I. & Al-Kharashi I.A. (2004). Arabic morphological analysis techniques: A comprehensive survey. Journal of the American Society For Information Science and Technology, 55(3):189–213.

Bakeer A. (2005). I'IRAB database. Internal Report, HIAST.

Bawab M., Meer Alam Y., Tayan H.& Mrayati, M. (1984). Arabic morphology system and Arabic phonetic transcription. Internal Report, HIAST.

Buckwalter. T. (2002). Buckwalter Arabic Morphological Analyzer Version 1.0. Linguistic Data Consortium (LDC) catalog number LDC2002L49 and ISBN 1-58563-257-0.

Chenfour N., Benabbou A. & Mouradi A. (2000). Etude et Evaluation de la di-syllabe comme Unité Acoustique pour le Système de Synthèse Arabe PARADIS. Second International Conference on language resources and evaluation, Athenes, Greece, 31 May-2 June.

Dutoit T., Pagel V., Pierret N., Bataille F. and van der Vrecken O. 1996. The MBROLA project: towards a set of high quality speech synthesizers free of use for non-

commercial purposes, Proceedings of ICSLP'96, pp. 1393-1396.

Ghneim N. & Al-Dakkak O. (2006). Arabic Language and the Computer. 5th Conference of the Arabic Academy, 20-22 November, Damascus, Syria.

Ghneim N. & Habash H. (2003). Transcription of Arabic texts into phonetic symbols. Damascus University Journal for essential sciences, Vol 19, Nb. 1, 2003.

Khoja S. & Garside R. (1999). Stemming Arabic text. Computing Department. Lancaster University. United Kingdom.

Larkey. L.S & Connell. M.E. (2001). Arabic information retrieval at UMass in TREC-10. In Proceedings of the 10th Text Retrieval Conference, TREC2001, pp. 562–570. Gaithersburg, Maryland.

Safadi H., Al-Dakkak O., Ghneim N. (2006). Computational Methods to Vocalize Arabic Texts. Second Workshop on Internationalizing SSML, Crete, Greece, 30-31 May.

Shaker N., Abou-Zliekha M.& Al-Dakkak O. (2008). SSML for Arabic languages, TSD2008, Czech.

Sonbol R., Ghneim N., Desouki M. (2008). Arabic Morphological Analysis: a new approach, ICTTA 2008, Damascus, SYRIA.

ASCII code	Phoneme Representation	Arabic Grapheme	pronunciation
98	b	ب	phoneme in st١ "book"
116	t	ت	phoneme in st١ "table"
120	x	ث	phoneme in st١ "thumb"
106	j	ج	phoneme in st١ "giraffe"
72	H	ح	Arabic
88	X	خ	last phoneme in "auch" German
100	d	د	"do" phoneme in st١
118	v	ذ	"that" phoneme in st١
114	r	ر	"run" phoneme in st١
122	z	ز	"zoo" phoneme in st١
115	s	س	"see" phoneme in st١
74	J	ش	"she" phoneme in st١
83	S	ص	(/Arabic (emphatic /s
68	D	ض	(/Arabic (emphatic /d
84	T	ط	(/Arabic (emphatic /t
90	Z	ظ	(/Arabic (emphatic /v
67	C	ع	Arabic
71	G	غ	phoneme in st١ "rue" French
102	f	ف	"food" phoneme in st١

ASCII code	Phoneme Representation	Arabic Grapheme	pronunciation
113	q	ق	(/Arabic (emphatic /k
107	k	ك	"kit" phoneme in st١
108	l	ل	"long" phoneme in st١
109	m	م	phoneme in st١ "moon"
110	n	ن	"no" phoneme in st١
104	h	ه	"hat" phoneme in st١
99	c	ء	"at" phoneme in st١
76	l	ل (مفخمة)	(/Arabic (emphatic /l
82	R	ر (مفخمة)	(/Arabic (emphatic /r
65	A	ـَ (مفخمة)	(/Arabic (emphatic /a
85	U	ـُ (مفخمة)	(/Arabic (emphatic /u
73	I	ـِ (مفخمة)	(/Arabic (emphatic /i
101	e	ـِ (ممالة)	"egg" phoneme in st١
111	o	ـُ (ممالة)	"on" phoneme in st١
119	w	و (غير مدية)	phoneme in st١ "what"
121	y	ي (غير مدية)	"yes" phoneme in st١
97	a	ـَ (فتحة)	last phoneme in "la" French
117	u	ـُ (ضمة)	last phoneme in "cout" French
105	i	ـِ (كسرة)	last phoneme in "qui" French

Table 1: Set of phonemes in our system

CV_Paths				
ID	**Semi_Path**	**Loga_Path**	**Latin_script**	**Arab_script**
1	D:\SemiSyllables\CV\1..8_bVsasa\1\bI.wav	D:\SemiSyllables\CV\1..8_bVsasa\1\bIsasa.wav	#bIsasa#	#بإساسا#
2	D:\SemiSyllables\CV\1..8_bVsasa\2\be.wav	D:\SemiSyllables\CV\1..8_bVsasa\2\besasa.wav	#besasa#	#بىساسا#
3	D:\SemiSyllables\CV\1..8_bVsasa\3\bo.wav	D:\SemiSyllables\CV\1..8_bVsasa\3\bosasa.wav	#bosasa#	#بأساسا#
4	D:\SemiSyllables\CV\1..8_bVsasa\4\bA.wav	D:\SemiSyllables\CV\1..8_bVsasa\4\bAsasa.wav	#bAsasa#	#بآساسا#
5	D:\SemiSyllables\CV\1..8_bVsasa\5\bU.wav	D:\SemiSyllables\CV\1..8_bVsasa\5\bUsasa.wav	#bUsasa#	#بةساسا#
6	D:\SemiSyllables\CV\1..8_bVsasa\6\ba.wav	D:\SemiSyllables\CV\1..8_bVsasa\6\basasa.wav	#basasa#	#باساسا#
7	D:\SemiSyllables\CV\1..8_bVsasa\7\bu.wav	D:\SemiSyllables\CV\1..8_bVsasa\7\busasa.wav	#busasa#	#بوساسا#

Table 2: Extraction of the acoustic database.