

A Formalism of Arabic Phonetic Grammar, and Application on the Automatic Arabic Phonetic Transcription of Transliterated Words

Muhammad Attia^{1,2}, Mohsen A. A. Rashwan^{1,2}, Galaal Khallaaf²

¹Dept. of Electronics & Electrical Communications, Faculty of Engineering, Cairo University

Faculty of Engineering, Cairo Univ., Giza, Egypt.

²The Engineering Company for the Development of Computer Systems; RDI.

171 Al-Haram main st., 6th floor, Giza, Egypt

{m_Atteya, mRashwan, Galal}@RDI-eg.com

Abstract

As the applications based on Text-To-Speech - esp. the telecom ones – are persistently growing into billions-of-USD business, the need for highly reliable large-scale Phonetic Transcriptors - i.e. Diacritizers - arises. It has been found that about 7.5% of the words in news domain Arabic text are transliterated words – mostly foreign proper nouns. This significant ratio of text are handled by none of the layers in the conventional ladder of linguistic layers that starts with morphological processing layer up to the pragmatic one.

In this paper we introduce an industrial-quality Diacritizer of Arabic Transliterated Strings based on an A* search methodology guided by long m-grams statistical model and constrained by a compact Arabic Phonetic Grammar (APG). In addition to presenting assertive formalization of APG, this paper adds APG to the theoretical ladder of Arabic linguistic processing layers and proposes adding Phonetic Grammar to other languages' ladders as well.

1.Introduction

The main motivation to ignite this work was the need to make our automatic Arabic phonetic transcriber *ArabDiac*[®] (RDI's *ArabDiac*[®], 2004), (Attia et al, 2002), (RDI's *ArabTalk*[®], 2004), (Hifny et al, 2003) - hence the based upon Arabic Text-To-Speech systems - effectively handle foreign names and terminology that frequently appear as transliterated Arabic strings in real-life Arabic text esp. in news domain. Our statistical measures made on several hundreds of thousand words Arabic news domain corpora shows that the ratio of transliterated foreign words is as high as 7.5%.

While other groups – including ourselves in the early stages - follow the traditional simple approach of building custom look-up tables; i.e. dynamic dictionaries (Sproat, 1998), of transliterated strings versus their manually crafted phonetic transcriptions, we later realized the apparent shortcomings of this approach that:

- 1- Due to the time variant nature of the occurrence of transliterated words in news domain text (Sproat, 1998); costly – and dirty - manual intervention is continuously needed.
- 2- Moreover, the completeness of those custom dictionaries can never be guaranteed.
- 3- Tolerability to spelling differences of transliterated strings is weak, hence, the matching process against the custom dictionary pushes the coverage even poorer.
- 4- Arabic infixes – prefixes and suffixes – are frequently added to the transliterated strings, and usually alter their spelling and/or phonetic

transcription. This is hard to account for using the aforementioned look-up technique.

- 5- Even when a hit of a given string against the look-up table occurs, no guarantee of the compliance of the obtained phonetic transcription with the Arabic phonology, which leads to crashing Arabic Text-To-Speech systems built over while the syllabification process.

In brief, the problems of that approach are i) High cost, ii) Poor coverage, iii) Poor matching, and iv) Fragility.

2.Statistical approach

To overcome these problems; we gave up all word based look-up tables, and instead built a statistical database of phoneme sequences; i.e. m-grams, so that our system records more generic - i.e. more time invariant – entities. Given that we collected enough statistics offline, we then build online the disambiguation lattice – see figure 1 below - of all the possible diacritizations of the given string.

The diacritization path with maximum likelihood probability is then obtained online using the admissible and optimal A* search algorithm (Nilsson, 1971), and using a combination of *Bayes'-Good-Turing discount-Back-off* techniques to estimate the probability of long phoneme m-gram path segments from the sparse statistical database built offline (Attia et al, 2002), (Jurafsky & Martin, 2000), (Katz, 1987), (Nadas, 1985), (Schutze & Manning, 2000).

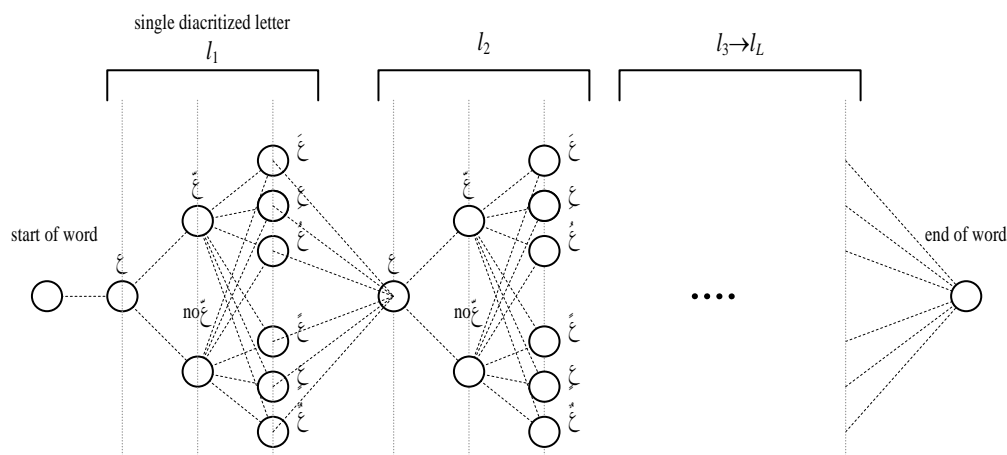


Figure 1; Search lattice for disambiguating diacritics of a given Arabic string using A* Search or Beam Search algorithm.

This statistical approach reduces the needed continuous manual intervention into cleanly and economically building enough diacritized corpus (see the last section of this paper) of transliterated Arabic strings for building the statistical phoneme m-grams database, and then incrementally adapting and refining this database at long intervals (annually, say). Due to the decomposition into word segments of phoneme m-grams as well as the ability of backing-off to even shorter m-grams, problems no. 2, 3 and 4 of the look-up tables are also recovered. Moreover, the statistically dominant long m-grams in the statistical approach preserve the main virtue of retrieving exact phonetic transcriptions in the look-up tables approach.

However, it remains the problem of guaranteeing the compliance of the obtained most likely diacritization path with the Arabic phonology.

3. Formalized Arabic Phonetic Grammar (APG)

To eliminate the threat of incompliance with Arabic phonology, we had to test each expanded path while the

searching process against a *formal Arabic Phonetic Grammar (APG)* of Arabic words. If the test fails that intermediate path is eliminated, else the path is added to the of open paths stack of A*.

Despite the rich literature on classic Arabic phonology (Mukhtaar Umar, 1990), (Anees, 1971), (Al-Aany, 1983), a formal APG written in BNF format was not available to enable the computational validation process mentioned above. Upon surveying the literature of classic Arabic phonologists for scanning the rules of Arabic phonology, we discovered one interesting point that *Arabic phonetic rules are conventionally stated negatively* (e.g. No Arabic word can start with two consecutive consonants) while formulating them in BNF needs *stating these rules assertively* which was a major bulk of our work in this regard.

After many iterations, we managed to formulate the compact – yet comprehensive – formal APG shown below in table 1.

$$\begin{aligned}
\mathbf{W} &:= \mathbf{y}_{\text{start}}[\mathbf{y}_{\text{mid}}\#][\mathbf{y}_{\text{end}}] \\
\mathbf{y}_{\text{start}} &:= \mathbf{c}_{\text{start}} \mathbf{f}_{\text{vowel}} \\
\mathbf{y}_{\text{mid}} &:= \mathbf{y}_{\text{mid,regular}}|\mathbf{y}_{\text{mid,sokoon}}|\mathbf{y}_{\text{mid,silent}} \\
\mathbf{y}_{\text{end}} &:= \mathbf{y}_{\text{end,sokoon}}|\mathbf{y}_{\text{end,silent}}|\mathbf{y}_{\text{end,layyina}}|\mathbf{y}_{\text{end,tanween}} \\
\mathbf{y}_{\text{mid,regular}} &:= \mathbf{c}_{\text{mid}}[\mathit{SHADDA}]\mathbf{f}_{\text{vowel}} \\
\mathbf{y}_{\text{mid,sokoon}} &:= \mathbf{c}_{\text{mid}} \mathit{SOKOON} \mathbf{c}_{\text{mid}} \mathbf{f}_{\text{vowel}} \\
\mathbf{y}_{\text{mid,silent}} &:= \mathbf{c}_{\text{mid}} \mathit{BYPASS} \\
\mathbf{y}_{\text{end,sokoon}} &:= (\mathbf{c}_{\text{end}} \mathit{SOKOON})|(\mathbf{c}_{\text{mid}} \mathit{SOKOON} \mathbf{c}_{\text{end}} \mathit{SOKOON})|(\mathbf{c}_{\text{mid}} \mathit{SHADDA} \mathit{SOKOON}) \\
\mathbf{y}_{\text{end,silent}} &:= \mathbf{c}_{\text{mid}} (\mathit{SOKOON}|\mathbf{f}_{\text{vowel}}|\mathbf{f}_{\text{tanween}}|(\mathit{SHADDA} \mathbf{f}_{\text{tanween}})) \mathbf{c}_{\text{end}} \mathit{BYPASS} \\
\mathbf{y}_{\text{end,layyina}} &:= \mathbf{c}_{\text{mid}}[\mathit{SHADDA}]\mathbf{f}_{\text{layyina}} \\
\mathbf{y}_{\text{end,tanween}} &:= \mathbf{c}_{\text{end}}[\mathit{SHADDA}]\mathbf{f}_{\text{tanween}} \\
\mathbf{c}_{\text{start}} &:= (\mathit{HMZA}|\mathit{BAA}|\mathit{TAA}|\dots|\mathit{HA}|\mathit{WAW}|\mathit{YAA})|(\mathit{ALIF}|\mathit{HMZe}) \\
\mathbf{c}_{\text{mid}} &:= (\mathbf{c}_{\text{start}} - \{\mathit{ALIF}, \mathit{HMZe}\})|(\mathit{HMZs}|\mathit{HMZy}|\mathit{HMZw}) \\
\mathbf{c}_{\text{end}} &:= \mathbf{c}_{\text{mid}}|\mathit{Yend}|\mathit{TAAM} \\
\mathbf{f}_{\text{vowel}} &:= (\mathit{FATEHA}[\mathit{ALIF} \mathit{VWL}])|(\mathit{KASRA}[\mathit{YAA} \mathit{VWL}])|(\mathit{DHAMMA}[\mathit{WAW} \mathit{VWL}]) \\
\mathbf{f}_{\text{layyina}} &:= \mathit{FATEHA} \mathit{YAA} \mathit{YAAL} \\
\mathbf{f}_{\text{tanween}} &:= \mathit{TNWa}|\mathit{TNWo}|\mathit{TNWe}
\end{aligned}$$

Table 1; Formalized APG in BNF format where terminals are written in italic capitals.

Besides guaranteeing the compliance of the resulting most likely diacritization with the phonology of Arabic words; validating against formal APG enhances the inherent efficiency of A* by early pruning many invalid intermediate paths, and guarantees an original Arabic

flavor of the pronunciation of transliterated Arabic strings.

For clear understanding, table 2 shown below explains the accurate significance of the terminals in the formal APG.

ID	Mnemonic	Orthography
1	<i>HMZA</i>	أ
2	<i>BAA</i>	ب
3	<i>TAA</i>	ت
4	<i>THAA</i>	ث
5	<i>JEEM</i>	ج
6	<i>HAA</i>	ح
7	<i>KHAA</i>	خ
8	<i>DAL</i>	د
9	<i>ZAL</i>	ذ
10	<i>RAA</i>	ر
11	<i>ZAE</i>	ز
12	<i>SEEN</i>	س
13	<i>SHEEN</i>	ش
14	<i>SSAD</i>	ص
15	<i>DHAAD</i>	ض
16	<i>TTAA</i>	ط

17	<i>DZAA</i>	ظ
18	<i>EIN</i>	ع
19	<i>GHEEN</i>	غ
20	<i>FAA</i>	ف
21	<i>QAAF</i>	ق
22	<i>KAF</i>	ك
23	<i>LAM</i>	ل
24	<i>MEEM</i>	م
25	<i>NOON</i>	ن
26	<i>HA</i>	هـ
27	<i>WAW</i>	و
28	<i>YAA</i>	ي
29	<i>Yend</i>	ى
30	<i>ALIF</i>	ا
31	<i>TAAM</i>	ة
32	<i>HMZe</i>	!

34	<i>HMZy</i>	ئ
35	<i>HMZw</i>	ؤ
36	<i>HMZs</i>	ء
37	<i>SHADDA</i>	ع
38	<i>FATEHA</i>	ع
39	<i>KASRA</i>	ع
40	<i>DHAMMA</i>	ع
41	<i>SOKOON</i>	ع
42	<i>TNWa</i>	ع
43	<i>TNWe</i>	ع
44	<i>TNWo</i>	ع
45	<i>VWL</i>	Non printable; the symbol @ is used for visualization.
46	<i>YAAL</i>	Non printable; the symbol ~ is used for visualization.
48	<i>BYPASS</i>	Non printable; the symbol × is used for visualization.

Table 2; Explaining the significance of terminals in the APG of figure 2.

4. Phonetic Grammar as the most bottom

NLP layer

Except for the formal APG, there is nothing specific to Arabic in the approach we presented to phonetically transcribing Arabic transliterated strings. Consequently, this approach is language independent given that phonetic grammars of different languages are computationally formalized as we did for Arabic.

One theoretical point deserves mentioning here regarding the famous abstraction of the NLP problem as mutually interacting successive linguistic processing layers ladder with the lower layers imposing constraints on the higher ones. (Rich & Knight, 1991), (Winston, 1992) While diacritizing strings corresponding to original words (Arabic or else) is constrained by the *Lexical* and may be the *Syntactic* processing layers (Attia et al, 2002), (Rich & Knight, 1991), diacritizing transliterated strings has no constraining layers but the *Phonetic Grammar* which locates it in the most bottom place in the NLP layers ladder as layer no. 0 as shown in figure 2 below.

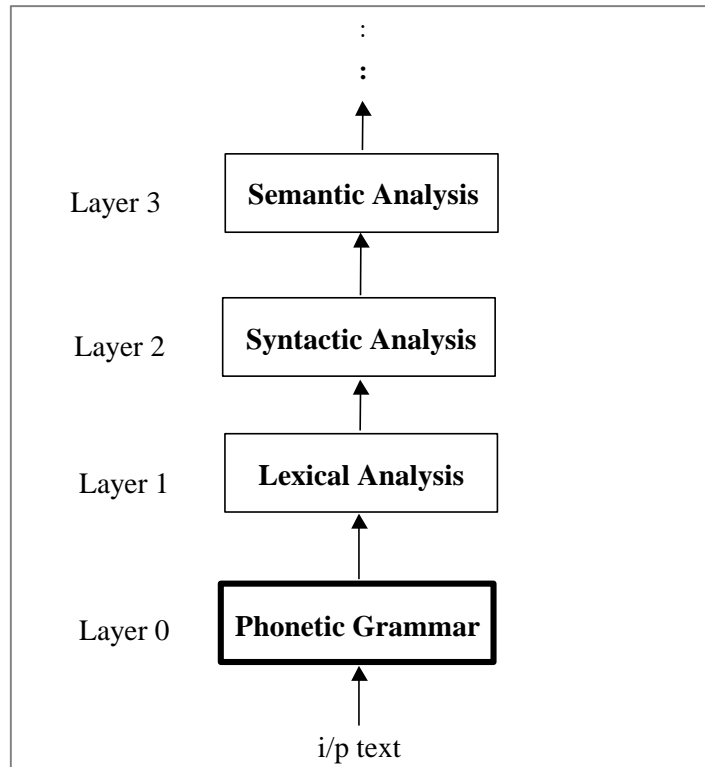


Figure 2; Phonetic grammar added as layer no. 0 in the theoretical ladder of linguistic processing layers.

5. Performance evaluation of the approach

Different writers in Arabic (and in other languages too) do not necessarily agree to the same phonetic transcription for the same transliterated word, and they do not even necessarily agree to the same spelling. So, there is no single correct answer that can be referenced while

evaluating the output of our *APG constrained statistical A* search* approach.

We hence followed an MOS-like approach that a committee of 3 persons (or any other odd number) are asked to evaluate the Arabic speech synthesized from the resulting phonetic transcription of each given transliterated word into one of the following ranks:

Rank	Significance
Perfect	Majority reports no errors
Very Good	Majority reports one error
Intelligible	Majority reports more than one error, but they can still understand the word
Unintelligible	Majority reports so many errors that they can not understand the word

While the results of our evaluation experiments are presented in the table below, readers can try online RDI's Arabic diacritizer *ArabDiac*[®] as well as Arabic Text-To-

Speech *ArabTalk*[®] with Arabic strings corresponding to transliterated and/or original Arabic words. (RDI's *ArabDiac*[®], 2004), (RDI's *ArabTalk*[®], 2004)

Size of training corpus	Max length of m-gram	Size of language model	Size of test sample	Evaluation rank	Ratio
7,123 transliterated words from about 100,000 news-domain words corpus.	15 phonemes	5.8 M.Bytes	1,106 transliterated words from about 14,000 news-domain words sample.	Perfect	43.9%
				Very Good	30.2%
				Intelligible	16.5%
				Unintelligible	9.4%
14,345 transliterated words from about 200,000 news-domain words corpus.	15 phonemes	7.3 M.Bytes	1,057 transliterated words from about 14,000 news-domain words sample.	Perfect	51.3%
				Very Good	35.1%
				Intelligible	9.4%
				Unintelligible	4.2%

Table 3; Results of evaluation experiments.

Finally, some few expressive examples are presented in the table below in order to give a concrete idea about the

quality of resulting diacritizations we get through *ArabDiac*[®] for Arabic transliterated strings.

Input string of transliterated word	Diacritized string using RDI's <i>ArabDiac</i> [®] method	Quality judgment
الديمقراطيون	آل×دِّي@مُقْرَأ@طَبِيو@ن	No errors-Perfect
ماساشوستس	مآ@سآ@شُو@سِتْسُنْ	No errors-Perfect
بوليفارد	بُو×لِفَا@رَدْ	No errors-Perfect
تويوتا	تُو×يُو×تَا@	No errors-Perfect
فالكرو موسومات	فَا@لُكْرُو@مُو×سُو@مَآ@تْ	4 errors-Unintelligible
للبياردو	لُلبِيَا@رَدُو×	1 error-Very Good
التراجيدية	آل×تْرَا@جِي@دِيَهْ	No errors-Perfect
انطونيو	آنْطُو×تِيُو@	No errors-Perfect
شنغهاي	شِنْغْهَآ@يْ	No errors-Perfect
رونالدو	رُو×نَا@ل×دُو@	3 errors-Still Intelligible

Table 4; Expressive examples of Arabic transliterated strings diacritized by *ArabDiac*[®].

- فونولوجيا العربية، د/سَلْمَان حَسَن العاني، ترجمة ياسر الملاح،

دار النادي الأدبي بجدة - المملكة العربية السعودية، ١٩٨٣ م.

Al-Aany (1983)

References in Arabic

- دراسة الصوت اللغوي، د/أحمد مختار عمر، عالم الكتب-

مصر، ١٩٩٠. Mukhtaar Umar (1990)

- الأصوات اللغوية، د/إبراهيم أنيس، مكتبة الأنجلو المصرية-

القاهرة، ١٩٧١ م. Anees (1971)

References in English

- An online trial version of the mentioned system RDI's *ArabDiac*[®] is found at: <http://www.RDI-eg.com> under the sub menu item Arabic NLP under the main menu item Technologies, (2004). (MS-Explorer[®] version 6 or later, and Arabic enabled MS-Windows[®] are needed)
- An online trial version of an Arabic Text-To-Speech system; RDI's *ArabTalk*[®] which is based on the Arabic diacritizer mentioned in this paper; RDI's *ArabDiac*[®] is found at: <http://www.RDI-eg.com> under the sub menu item Speech under the main menu item Technologies, (2004). (MS-Explorer[®] version 6 or later, and Arabic enabled MS-Windows[®] are needed)
- Attia, M., Rashwan, M., Khallaaf, G., (2002) *On Stochastic Models, Statistical Disambiguation, and Applications on Arabic NLP Problems*, The Proceedings of the 3rd Conference on Language Engineering; CLE'2002, the Egyptian Society of Language Engineering (ESLE). This paper is also downloadable from the following web pages; <http://www.NEMLAR.org/ScientificPapers/Index.htm> and <http://www.RDI-eg.com> under the menu sub item *Arabic NLP* under the main menu item *Technologies*.
- Hifny, Y., Qurany, S., Hamid, S., Rashwan, M., Attia, M., Ragheb, A., Khallaaf, G., (2003) *ArabTalk*[®]; *An Implementation for Arabic Text To Speech System*, The proceedings of the 4th Conference on Language Engineering; CLE'2003, the Egyptian Society of Language Engineering (ESLE), and published also in the News Letter of Evaluation of Language Resources and Distribution Agency (ELDA) May 2004 issue.
- Jurafsky, D., Martin, J. H., (2000) *Speech and Language Processing; An Introduction to Natural Language Processing, Computational Linguistics, and Speech Processing*, Prentice Hall.
- Katz, S.M., (1987) *Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer*, IEEE Transactions on Acoustics, Speech and Signal Processing, vol. ASSP-35 no. 3, March 1987.
- Nadas, A., (1985) *On Turing's Formula for Word Probabilities*, IEEE Transactions on Acoustics, Speech and Signal Processing, vol. ASSP-33 no. 6, December 1985.
- Nilsson, N.J., (1971) *Problem Solving Methods in Artificial Intelligence*, McGraw-Hill.
- Rich, E., Knight, K., (1991) *Artificial Intelligence 2nd edition*, McGraw-Hill.
- Schutze, H., Manning, C.D., (2000) *Foundations of Statistical Natural Language Processing*, the MIT Press.
- Sproat, R., (1998) *Multilingual Text-To-Speech Synthesis*, Kluwer Academic Publishers.
- Van Santen, J.P.H., Sproat, R.W., Olive, J.P., Hirschberg, J., (1998) *Progress in Speech Synthesis*, Springer Publishers.
- Winston, P.H., (1992) *Artificial Intelligence 3rd edition*, Addison Wesley.