Fassieh®; a Semi-Automatic Visual Interactive Tool for Morphological, PoS-Tags, Phonetic, and Semantic Annotation of Arabic Text Corpora

Mohamed Attia[1], Mohsen A. A. Rashwan[1], Mohamed A. S. A. A. Al-Badrashiny[1]
[1] The Engineering Company for the Development of Computer Systems; RDI, Egypt www.RDI-eg.com
{m_Atteya, Mohsen_Rashwan, Mohammed.Badrashiny}@RDI-eg.com

Abstract

Arabic is a language located on the extreme of richness at its low-level processing layers; orthography, phonology, morphology, PoS-tagging, … ,etc. The shortcomings of directly dealing with such richness; e.g. low coverage, triggered us to resort to language factorization esp. as Arabic is inherently based on a relatively compact basis set of building entities along with a concise set of analytic/synthetic rules. Our Arabic morphological, PoS-Tagging, phonetic, and lexical semantics factorization models have hence proved over years to be quite successful in handling that highly generative language. There remains however the ambiguity problem primarily emanating from the lack of higher-level processing layers which in turn should presumably be built over those lower-level ones! Being the currently best known feasible and sound approach, we use statistical disambiguation to get out of this circular dilemma.

No matter how powerful stochastic modeling, training, and disambiguation are, an inevitable error margin resides. While many applications may live with the realized small error margins, other few ones; e.g. building supervised statistical training corpora, are intolerant to whatever error margins. For both kinds we have built Fassieh® which is an annotation tool of Arabic text. This tool is a sophisticated GUI application that enables the automatic factorization of large Arabic text corpora, in addition to a multitude of features enabling a guided, normalized, and efficient proofreading of any part of those factorized corpora.

This paper reviews the aforementioned factorization models along with the associated statistical disambiguation, then presents Fassieh® which is not only an annotation tool, but is also an evaluation, demonstrative, and tutorial means.

1. Introduction

Like huge problems that cannot be tackled once as a whole, language processing is theoretically thought of as a ladder of processing layers escalating from orthography, phonology, morphology … to semantics, discourse analysis, and pragmatics. [2], [7], [11], [20] This is much alike the multi-layers model popularly adopted while studying computer networks. While natural languages tend to be similar at their higher-level layers, their lower layers are distinct. See for example how far one can guess on a language via listening to a short dialogue between two of its speakers; i.e. via its phonology.

Arabic is famous for being a highly sophisticated language. This is due perhaps to its long contiguous history as a major human language for more than 2,000 years! [3] Besides the complexities of Arabic orthography [3] and phonology [2], [5], [23], the Arabic morphology has always remained remarkably difficult to computationally model. Difficulties in this regard typically arise when Arabic is artificially approached as a vocabulary-based language; e.g. English. Even huge dictionaries encompassing millions of full-form words cannot fully cover all the legitimate Arabic words generable via the fertile derivative and inflective mechanisms of this language. [8], [9]

Deep investigation, however, reveals Arabic to be systematically structured around a compact set of basic building entities along with a concise set of analytic/synthetic rules. [2], [7] The whole Arabic morphology, for example, has been factorized in our model into no more than 7,800 morphemes + around 1,100 inflection/derivation rules. [7] As mentioned on subsection 2.1 below; the coverage of this morphological factorization exceeds 99.8%. Similarly, Arabic PoS-tagging reviewed on subsection 2.2 is fully covered via our 62 PoS-tags set [4], the Arabic phonological constraints reviewed on subsection 2.3 have been encapsulated in a 14-statement formal BNF grammar [5], and finally on subsection 2.4; our Arabic lexical semantic mapping relying only on a couple-of-thousands word senses [1] is reviewed.

While some factorization models are pleasantly one-to-one mapping systems, others are ambiguously one-to-many mapping ones. The major source of ambiguity in such systems is ultimately due to the absence of high-level NLP knowledge. The current state-of-the-art of NLP is long years behind providing mature models of the high layers

like semantics and pragmatics on the NLP ladder.[1] [2], [6], [20] Things are made tougher by the sad fact that; due to the limited available knowledge about the interaction among the NLP layers, those layers are simplistically cascaded; i.e. one over/after another. This means the higher layers rely on the analysis results of the lower ones while those lower ones wait for the answers of the higher layers to disambiguate their output! [2]

Stochastic modeling and statistical inference is then resorted to as the suboptimal most feasible and effective approach to get out/around this circular problem. While stochastic modeling strives to build distributions of linguistic entities and their compounds as close as possible to their occurrences in the real linguistic phenomena represented by sample training corpora, statistical disambiguation strives to infer the most likely proposed sequences in light of those distributions. [2], [13], [18], [20] Section 3 reviews our statistical training and disambiguation mechanism based on the *maximum a posteriori* (MAP) probability estimation approach borrowed from the electrical communications engineering where it is one pillar of signals language processing. [2]

Language factorization models like the ones we present here are not only vital for issues like coverage, completeness verifiability, and compactness, but are also impactful on whatever statistical language processing methodology deployed. While stochastic modeling as well as statistical disambiguation might be directly applied on un-factorized (raw) language sequences, their performance gets better and better as they are applied on deeper and deeper linguistic analysis of these sequences given the same algorithms, training corpora, and computational power. Mathematically, as we delve deeper in linguistic analysis; e.g. from morphological, to PoS tagging, to lexical semantics … etc. resolving more and more complex relations, the raw sequences are factorized into more fundamental - and typically less numerous - atomic entities to be dealt with. This in turn reveals more concentrated statistical correlations and reduces the dimensionality of the problem, which both sharpen the effectiveness of the statistical processing. [1], [12], [13], [19], [20]

The stochastic importance of language factorization gets more and more magnified as the vocabulary and structure of the target language get richer where Arabic is an extreme, and fortunately is also on another extreme of the compactness and regularity of its atomic entities and building process. [1], [2], [7], [9], [24]

Producing large factorized Arabic text corpora is made easy and tidy via our text annotation tool *Fassieh*® where Arabic morphological analysis, PoS tagging, phonetic transcription, and lexical semantics are all automatically enabled. While presenting *Fassieh*® on section 4, we also mention a group of auxiliary linguistic tools; e.g. morpheme dictionaries, and illustrative GUI tools; e.g. character/word status coloring, that not only render the proofreading of automatically produced factorizations accurate and efficient, but also turn this tool into an invaluable Arabic NLP demonstrative, tutorial, and evaluation means.

[1] One common reason of this absence is the lack of the necessary, but not sufficient, real world's common sense knowledge. Collecting, filtering, integrating, and coding such knowledge is a gigantic project that – like the Genome project - may need a multinational effort to conduct.

2. Arabic Factorization Models

2.1. Arabic Morphological Analysis: [2], [7]

One basic idea behind our Arabic morphological factorization model is the canonical structure of *any* Arabic word ***w*** that can simply be formulated as a quadruple;

$$w \rightarrow \underline{q} = (t : p, r, f, s) \quad (1)$$

… where *p* is *prefix* code, *r* is *root* code, *f* is *pattern* (or *form*) code, and *s* is *suffix* code. With *t* is word type code whose possible values ∈ {*Regular Derivative*, *Irregular Derivative*, *Fixed*, *Arabized*}, the morphemes that constitute *all* the quadruples uniquely representing *any* generable Arabic word are classified in the 9 categories illustrated by fig.1 below:

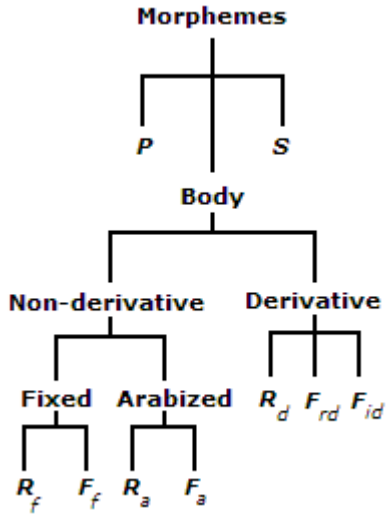

Figure 1. The 9 categories of Arabic morphemes.

Exclusive to Arabic transliterated foreign words - see sec. 2.3 - a dynamic coverage ratio ≥ 99.8% over the standard Arabic language in large is realized by our morphological factorization model relying only on a compact lexicon of about 7,800 morphemes whose distribution over the 9 categories of Arabic morphemes is shown by table 1 below:

Kind of morpheme	*Symbol*	*No. of morphemes*
Prefixes	P	260
Derivative roots	R_d	4600
Regular derivative patterns	F_{rd}	1000
Irregularly derived words	F_{id}	300
Roots of Arabic fixed words	R_f	250
Arabic fixed words	F_f	300
Roots of Arabized words	R_a	240
Arabized words	F_a	290
Suffixes	S	550

Table 1; No. of Arabic morphemes in each category.

The third powerful idea behind our model is the *agent-driven* style in which each morpheme in our lexicon is coded. Collected and filtered from dozens of classical Arabic linguistics sources [22], [24], [25]; all the context-free orthographic, phonological, and morphological features of each morpheme are systematically encoded using very concise feature-sets. Besides this encoding, each morpheme declares its predefined morphological *properties* and *actions*. The compatibility between the properties of whatever two morphemes can be tested, and if a positive test result is obtained, the morphemes' actions defining the mutual orthographic, phonological, and morphological effects between the two morphemes upon their combination are executed. The secret here again is the comprehensiveness and conciseness of these properties and actions. [7]

Given our Arabic lexicon coded in such a way, our morphological analysis is then realized efficiently through a successive constraining methodology. [7] Illustrative sample outputs of such a process are given by table 2 below:

Sample word	*Word type*	*Prefix & prefix code*	*Root & root code*	*Pattern & pattern code*	*Suffix & suffix code*
فَمَا	*Fixed*	فَ 2	اَلَّذِي 87	مَا 48	ـ 0
تَتَنَاوَله	*Regular Derivative*	تـ 86	ن و ل 4077	تَفَاعَلَ 176	ـه 8
اَلْكِتَابَات	*Regular Derivative*	الـ 9	ك ت ب 3354	فِعَال 684	ـات 27
اَلْعِلْمِيَّة	*Regular Derivative*	الـ 9	ع ل م 2754	فِعْل 842	ـيَّة 28
مِنْ	*Fixed*	ـ 0	مِنْ 63	مِنْ 118	ـ 0
مَوَاضِيع	*Regular Derivative*	ـ 0	و ض ع 4339	مَفَاعِيل 93	ـ 0
مُتَّخَذة	*Irregular Derivative*	ـ 0	أ خ ذ 39	مُتَّخَذ 13	ـة 26

Table 2; Exemplar Arabic morphological analyses.

2.2. Arabic PoS-tagging: [2], [4]

Part-of-Speech (PoS) tagging is a fundamental linguistic analysis process where PoS-tags that convey the basic context-free syntactic features of input text words are extracted.

Instead of the infeasible task of scanning the *morpho-syntactic* features of each possible Arabic word in order to compose our Arabic PoS-tags set, we had only to scan the *morpho-syntactic* features of the 7,800 morphemes in our aforementioned compact lexicon, which have then been digested through several iterations of decimation into a non-redundant compact Arabic PoS-tags set composed of only 62 tags.

Completeness, atomicity, and insurability of the scanned morpho-syntactic features were the main criteria adhered to during that process. [2], [4]

While many of the resulting Arabic PoS-tags may have corresponding ones in other languages; e.g. English, few do not have such counterparts and may be specific to the Arabic language.

Each morpheme in our Arabic morphological knowledge base is then PoS labeled as exemplified by the sample Arabic PoS labels of few morphemes shown in table 3 below:

Morpheme	*Type & Code*	*Arabic PoS tags vector label*
الـ	*P* *9*	**[Definitive]** [ال التعريف]
سَيـ	*P* 125	**[Future, Present, Active]** [استقبال، مضارع، مبني للمعلوم]
مُفَاعِل	F_{rd} 482	**[Noun, Subjective Noun]** [اسم، اسم فاعل]
اسْتِفْعَال	F_{rd} 67	**[Noun, Noun Infinitive]** [اسم، مصدر]
مَلَائِك	F_{id} 29	**[Noun, No SARF, Plural]** [اسم، ممنوع من الصرف، جمع]
هُوَ	F_f 8	**[Noun, Masculine, Single, Subjective Pronoun]** [اسم، مذكر، مفرد، ضمير رفع]
ذُو	F_f 39	**[Noun, Masculine, Single, Adjunct, MARFOU']** [اسم، مذكر، مفرد، مضاف، مرفوع]
ـات	*S* *27*	**[Feminine, Plural]** [مؤنث، جمع]
ـونَهُمْ	*S* *427*	**[Present, MARFOU', Subjective Pronoun, Objective Pronoun]** [مضارع، مرفوع، ضمير رفع، ضمير نصب]
ـيَّتَانِ	*S* *195*	**[Relative Adjective, Feminine, Binary, Non Adjunct, MARFOU']** [نسب، مؤنث، مثنى، غير مضاف، مرفوع]

Table 3. PoS labels of sample Arabic morphemes.

Three points here must be considered:

I- For any quadruple - see formula 1 above - the Arabic PoS tagging of the body – i.e. stem - is retrieved from the PoS label of the pattern; (*t: f*), while the prefix; *p* and suffix; *s* give the Arabic PoS-tagging of affixes. So, the root morphemes of all kinds do not participate to Arabic PoS-tagging, and are hence not PoS labeled.

II- Due to the atomicity of the tags in the Arabic PoS-tags, and to the compound nature of Arabic

morphemes in general, the PoS labels of Arabic morphemes are vectors of PoS-tags.

III- Only ensured Arabic PoS-tags are included in the Arabic PoS labeling of morphemes; i.e. when an Arabic PoS tag is not a certain - even if it is highly probable – feature of some morpheme, it is not included in its Arabic PoS label.

Finally, the Arabic PoS-tagging runtime process is implemented in the following steps: [2], [4]

I- The sequence of raw Arabic words be PoS-tagged are morphologically analyzed then combinatorially disambiguated; see sec. 3. This results in a disambiguated quadruples sequence where each raw word is substituted by either one quadruple, or a mark of transliterated string.

II- For the prefix, pattern, and suffix morphemes of each quadruple in the sequence, the Arabic PoS labels; *APoS(p), APoS(t: f),* and *APoS(s)* are retrieved.

III- The Arabic PoS-tags vector of each full word in the sequence is then composed as:

$$APoS(w)=Concat(APoS(p), APoS(t: f), APoS(s)) \quad (2)$$

where *Concat* is a function that concatenates the PoS sub-vectors of the constituting morphemes after eliminating any mutual redundancy among their tags.

The Arabic PoS-tags vectors resulting after the application of our model on the sample words of table 2 above are shown in table 4 below.

Sample word	***Arabic PoS tags vector***
فَمَا	[Conjunction, Noun, Relative Pronoun, Null Suffix] [عطف، اسم، اسم موصول، لا لاحقة]
تَتَنَاوَله	[Present, Active, Verb ,Objective Pronoun] [مضارع، مبني للمعلوم، فعل، ضمير نصب]
اَلْكِتَابَات	[Definitive, Noun, Plural, Feminine] [ال التعريف، اسم، جمع، مؤنَّث]
اَلْعِلْمِيَّة	[Definitive, Noun, Relative Adjective, Feminine, Single] [ال التعريف، اسم، نسب، مؤنَّث، مفرد]
مِنْ	[Null Prefix, Preposition, Null Suffix] [لا سابقة، حرف، لا لاحقة]
مَوَاضِيع	[Null Prefix, Noun, No SARF, Plural, Null Suffix] [لا سابقة، اسم، ممنوع من الصرف، جمع، لا لاحقة]
مُتَّخَذة	[Null Prefix, Noun, Objective Noun, Feminine, Single] [لا سابقة، اسم، اسم مفعول، مؤنَّث، مفرَد]

Table 4. PoS-tags vectors of sample Arabic words.

2.3. Arabic Phonetic Transcription:

The spelling of a given Arabic word alone is not sufficient for a direct mapping into its exact phonetic transcription. Extra marks – called *diacritics* - need be accompanying each letter to indicate how the major sound of this letter is *vowelized* upon the utterance of the text it is involved in. As most modern standard Arabic (MSA) writers scarcely add such marks, the inference of these diacritics need be made through Arabic NLP.

Our internal orthographic model in this regard had first to augment the typical set of Arabic diacritics from 8 to 12 marks in order to fully describe the phonetic transcription of any Arabic text. The phonetic transcription coded in whatever convention; IPA, SAMPA … etc. can then be trivially obtained via a one-to-one mapping from the fully diacritized Arabic text. [2]

The diacritization of Arabic words is obtained through two components; morphological, and syntactic.

The first, and most important of the two, is obtained as a byproduct of the Arabic morphological analysis; see table 2 above. As presented on sec. 2.1 above; prefixes, patterns, and suffixes in our Arabic lexicon encode their full phonological features along with the possible mutual phonological effects upon their allowable combinations.

Only about 65% of the Arabic words have one diacritic that is dependent on their syntactic role through the parsing tree of their surrounding text. The diacritization of the rest is morphology-dependent in full. As those syntax-dependent diacritics may always be defaulted to produce acceptable pronunciation of MSA text [2], the syntactic diacritization component of Arabic words is regarded as a complementary enhancement. While *Fassieh*® allows annotators to directly add syntactic diacritics whenever missing; see fig. 6 in sec. 4 below, those diacritics can also be inferred statistically with the Arabic PoS-tags as main input features. [4]

Occurring at as a high frequency as 7.5%, the diacritization of non-Arabic words written in Arabic orthography; i.e. *transliterated* words, raises another tough challenge as they are apparently not governed by Arabic morphology. Moreover, building a look-up table of such words is a messy solution, not only due to the trouble collecting them and the multiple scripts given for the same word by different writers, but also as the phenomenon of their emergence is highly time-variant. One more complication is the tendency of MSA writers to append affixes to Arabic transliterated words. [2], [5]

Seeking for a solution that may survive all these complexities at a rational cost of training and update, we statistically infer the diacritics of such words based on long m-grams stochastic models of letters and diacritics; see sec. 3. [2], [5] Without any linguistic constraints, statistical inference alone would perform poorly due to the underlying enormous *perplexity*. For this reason, the NLP layer of phonology preceding that of morphology on the NLP ladder had to be modeled. A years-long investigation of classical Arabic phonology [23] resulted in a comprehensive formalization of the intra-word Arabic

phonology as a simple compact grammar shown in fig. 2 below. This solution is also applicable on the same problem with any language provided a formalization of its intra-word phonology.

In order to produce either connected-speech or isolated-word Arabic phonetic transcription, the Arabic inter-word phonology also is synthetically formalized in a very simple flowchart with only 8 conditional rules. [2]

$\mathbf{W} := \mathbf{y}_{start}[\mathbf{y}_{mid}\#][\mathbf{y}_{end}]$

$\mathbf{y}_{start} := \mathbf{c}_{start}\ \mathbf{f}_{vowel}$
$\mathbf{y}_{mid} := \mathbf{y}_{mid,regular}|\mathbf{y}_{mid,sokoon}|\mathbf{y}_{mid,silent}$
$\mathbf{y}_{end} := \mathbf{y}_{end,sokoon}|\mathbf{y}_{end,silent}|\mathbf{y}_{end,layyina}|\mathbf{y}_{end,tanween}$

$\mathbf{y}_{mid,regular} := \mathbf{c}_{mid}[SHADDA]\mathbf{f}_{vowel}$
$\mathbf{y}_{mid,sokoon} := \mathbf{c}_{mid}\ SOKOON\ \mathbf{c}_{mid}\ \mathbf{f}_{vowel}$
$\mathbf{y}_{mid,silent} := \mathbf{c}_{mid}\ BYPASS$

$\mathbf{y}_{end,sokoon} := (\mathbf{c}_{end}\ SOKOON)|(\mathbf{c}_{mid}\ SOKOON\ \mathbf{c}_{end}\ SOKOON)|(\mathbf{c}_{mid}\ SHADDA\ SOKOON)$
$\mathbf{y}_{end,silent} := \mathbf{c}_{mid}\ (SOKOON|\mathbf{f}_{vowel}|\mathbf{f}_{tanween}|(SHADDA\ \mathbf{f}_{tanween}))\ \mathbf{c}_{end}\ BYPASS$
$\mathbf{y}_{end,layyina} := \mathbf{c}_{mid}[SHADDA]\mathbf{f}_{layyina}$
$\mathbf{y}_{end,tanween} := \mathbf{c}_{end}[SHADDA]\mathbf{f}_{tanween}$

$\mathbf{c}_{start} := (HMZA|BAA|TAA|...|HA|WAW|YAA)|(ALIF|HMZe)$
$\mathbf{c}_{mid} := (\mathbf{c}_{start} - \{ALIF,HMZe\})|(HMZs|HMZy|HMZw)$
$\mathbf{c}_{end} := \mathbf{c}_{mid}|Yend|TAAM$

$\mathbf{f}_{vowel} := (FATEHA[ALIF\ VWL])|(KASRA[YAA\ VWL])|(DHAMMA[WAW\ VWL])$
$\mathbf{f}_{layyina} := FATEHA\ YAA\ YAAL$
$\mathbf{f}_{tanween} := TNWa|TNWo|TNWe$

Figure 2. Inter-word Arabic Phonetic Grammar in BNF, with terminals written in capital italics.

2.4. Arabic Lexical Semantics Analysis: [1]

The objective of this analysis is simply to map any legitimate Arabic morphological quadruple into the appropriate element of a closed set of *semantic fields*; i.e. *word senses*.

Building a knowledge base relating the two sides of mapping is a huge task, and it'd better not to start from scratch. Neatly based on the theory of semantic fields [15], [26], [29], we have surveyed the best sources treasuring such Arabic lexical semantic knowledge base in its raw format [27], [28], [30]. After iterations of filtering, refining, patching, and homogenizing this material, we were essentially left with a relational database (RDB) relating about 2,000 semantic fields as their primary keys with about 40,000 Arabic words.

To maximize the coverage of that RDB over the generable Arabic words, these 40,000 full-form words are first encoded in the form of what we call *PoS-constrained Arabic lexical compounds*. Each of these compounds is composed of the underlying morphemes that are flexible to be fully or partially matched against the morphemes composing a given Arabic word. A morpheme code is explicitly mentioned only *if* its exact existence in the lexical compound is necessary to imply the semantic field(s) tied to this lexical compound. If the existence of *any* morpheme containing a certain PoS-tag is only necessary to imply those semantic field(s), the code of this PoS-tag with a negative sign is mentioned in place of that morpheme. A *don't-care* code (assigned -1000) in some place signifies that the morpheme at that place is semantically neutral. [1]

Next, the resulting *forward* RDB is inverted using SQL operation to produce an *inverse* RDB whose primary key is a PoS-constrained lexical compound provoking the possible semantic fields it may belong to.

As a final touch on the inverse RDB, a special *back-off* row is inserted per each distinct root in the inverse RDB in order to further attenuate the runtime retrieval miss ratio of input words. The lexical compound of a back-off row mentions only the root morpheme explicitly, and all the other morphemes (prefix, pattern, and suffix) as *don't care*. If an input word matches none of the explicitly registered derivatives of some root in the inverse RDB, the corresponding back-off row is resorted to. The recalled semantic fields of such a row are the union of the recalled semantic fields of all the registered derivatives of its root in the inverse RDB.

A sample fragment of this inverse lexical semantic RDB is shown by table 5 on the next page.

3. Statistical Disambiguation

The challenging ambiguity of the outputs of Arabic language factorizations may be exemplified by table 6 below showing 12 possible morphological analyses of one simple Arabic raw word.

Diacritized word	*Type*	*Prefix &* *prefix code*	*Root &* *root code*	*Pattern &* *pattern code*	*Suffix &* *suffix code*
بَطِين	*Regular* *Derivative*	– **0**	ب ط ن **352**	فَعِيل **673**	– **0**
بُطَيْن	*Regular* *Derivative*	– **0**	ب ط ن **352**	فُعَيْل **789**	– **0**
بَطَّيْنِ	*Regular* *Derivative*	– **0**	ب ط ط **348**	فَلَّ **820**	ـــيْنِ **80**
بِطِين	*Regular* *Derivative*	بِـــ **15**	ط ي ن **2563**	فِـــعـــل **847**	– **0**
بِطِّين	*Regular* *Derivative*	– **0**	ب ط ن **352**	فِعِّيل **673**	– **0**
بُطَيِّن	*Regular* *Derivative*	– **0**	ب ط ن **352**	فُعَيِّل **788**	– **0**
بِطَيْن	*Regular* *Derivative*	بِـــ **15**	ط ي ن **2563**	فَعْل **819**	– **0**
بُطَّيْنِ	*Regular* *Derivative*	– **0**	ب ط ط **348**	فُلَّ **834**	ـــيْنِ **80**
بِطَّيْنِ	*Regular* *Derivative*	– **0**	ب ط ط **348**	فِلَّ **843**	ـــيْنِ **80**
بِطِيَن	*Regular* *Derivative*	بِـــ **15**	ط ي ن **2563**	فِعِل **850**	– **0**
بِطُيَّن	*Regular* *Derivative*	بِـــ **15**	ط ي ن **2563**	فُعَّل **739**	– **0**
بِطَيِّن	*Regular* *Derivative*	بِـــ **15**	ط ي ن **2563**	فَعَّل **675**	– **0**

Table 6. The multiplicity of the possible morphological analyses of a sample input Arabic word: (**بطين**)

Lexical compound												*Possible semantic fields*						
String	Q_1					Q_2					…	SF_1	SF_2	SF_3	SF_4	SF_5	SF_6	…
	t	*r*	*f*	*p*	*s*	*t*	*r*	*F*	*P*	*s*								
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
ك ت ب	1	3354	-1000	-1000	-1000	-1000	-1000	-1000	-1000	-1000	…	الاستثمار 1800	العُقود 450	المراسَلة 179	التأليف 466	الكِتابة 1678	الإلزام 590	…
تَكَاتَبَ	1	3354	176	-1000	-1000	-1000	-1000	-1000	-1000	-1000	…	العُقود	-	-	-	-	-	…
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
اِكْتَتَبَ	1	3354	249	-1000	-1000	-1000	-1000	-1000	-1000	-1000	…	الاستثمار	-	-	-	-	-	…
اِكْتَتَبَ فِي	1	3354	249	-1000	-1000	3	42	132	-1000	-1000	…	الاستثمار	-	-	-	-	-	…
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
اِكْتِتَاب	1	3354	280	-1000	-1000	-1000	-1000	-1000	-1000	-1000	…	الاستثمار	-	-	-	-	-	…
كَاتَبَ	1	3354	457	-1000	-1000	-1000	-1000	-1000	-1000	-1000	…	المراسَلة	-	-	-	-	-	…
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
مُكَاتَبَةٌ	1	3354	487	-1000	-1000	-1000	-1000	-1000	-1000	-1000	…	المراسَلة	-	-	-	-	-	…
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
كَتَبَ	1	3354	859	-1000	-1000	-1000	-1000	-1000	-1000	-1000	…	العُقود	المراسَلة	التأليف	الكِتابة	الإلزام	-	…
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
كِتَابٌ	1	3354	648	-1000	-1000	-1000	-1000	-1000	-1000	-1000	…	العُقود	المراسَلة	التأليف	الكِتابة	الإلزام	-	…
كِتَابَةٌ	1	3354	648	-1000	-48	-1000	-1000	-1000	-1000	-1000	…	المراسَلة	التأليف	الكِتابة	-	-	-	…
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Table 5. A sample fragment of the inverse Arabic lexical semantic RDB.

The multiplicity of possible analyses may wildly grow to exceed a hundred of possible analyses for the same test unit. To resolve these linguistic ambiguities, we rely on the well established approach of *maximum a posteriori* (MAP) probability estimation [2], [6], [13], [20] famously formulated by:

$$\hat{\underline{I}} = \arg\max_{\forall \underline{I}} \{P(\underline{I} \mid \underline{O})\} = \arg\max_{\forall \underline{I}} \left\{ \frac{P(\underline{O} \mid \underline{I}) \cdot P(\underline{I})}{P(\underline{O})} \right\} = \arg\max_{\forall \underline{I}} \{P(\underline{O} \mid \underline{I}) \cdot P(\underline{I})\} \quad \text{___ (3)}$$

In other pattern recognition problems like OCR and automatic speech recognition (ASR), the term $P(\underline{O}|\underline{I})$ referred to as the *likelihood* probability, is modeled via probability distributions; e.g. HMM in ASR. [17]

Our aforementioned language factorization models enable us to do better by viewing the formal available structures, in terms of probabilities, as a binary decision; i.e. a decision of whether the observation obeys the formal rules or not. This simplifies formula no. 3 into:

$$\hat{\underline{I}} = \arg\max_{\forall \underline{I} \in \Re} \{P(\underline{I})\} \quad \text{___(4)}$$

… where $\Re$ is the space of factorization model, and $P(\underline{I})$ is the independent probability of the input which is called the statistical language model (SLM). The term $P(\underline{I})$ then expresses the m-grams probability estimated according to the distributions computed from the training corpus.

Faced with the severe *Zipfian* sparseness of m-grams of whatever natural language entities [2], [6], [13], [20], we resort to the *Bayes'-Good_Turing_discount-Back_off* hybrid methodology [14] for both building their discrete distributions from labeled corpora and also for estimating the probability of any given m-gram w_1^m of these entities in runtime.

Using the chain rule for decomposing marginal into conditional probabilities, the term $P(\underline{I})$ may be approximated by:

$$P(\underline{Q}) \cong \prod_{i=1}^{L} P(q_i \mid q_{i-h}^{i-1})$$

… where *h* is maximum affordable m-gram in the SLM. Running a language factorization model on a sequence of raw linguistic units one by one leaves us with cascaded ambiguous columns of possible analyses of each producing the typical trellis search problem. [2], [6] Using a variant of A^*-based algorithm; e.g. beam search, is the best known way for obtaining the most likely sequence of analyses among the exponentially increasing space $\boldsymbol{S}$ of possible sequences (paths) implied by the trellis's topology in light of formula no. 4 by obtaining:

$$\hat{\underline{Q}} = \arg\max_{\mathbf{S}} \left\{ P\left(q_{1,j_1}^{L,j_L}\right) \right\} =$$

$$\arg\max_{\mathbf{S}} \left\{ \prod_{i=1}^{L} P\left(q_{i,j_i} \mid q_{(i-h),j_{(i-h)}}^{(i-1),j_{(i-1)}}\right) \right\} = \quad \text{___ (5)}$$

$$\arg\max_{\mathbf{S}} \left\{ \sum_{i=1}^{L} \log P\left(q_{i,j_i} \mid q_{(i-h),j_{(i-h)}}^{(i-1),j_{(i-1)}}\right) \right\}$$

An example of the Arabic morphological search trellis is shown in figure 3 below that achieves an error margin below 5% with SLM whose h = 12 and are built from about 2,400,000 morphemes extracted from Arabic text corpora balanced over several domains news, scientific, sport, ... etc. [2], [21] We use similar trellises with different topologies for other tasks; e.g. phonetic disambiguation. [2], [4], [5]

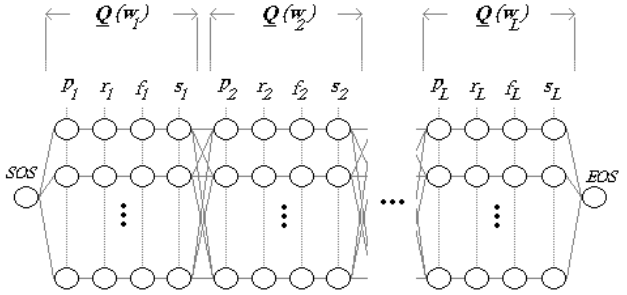

Figure 3. Arabic morphological disambiguation trellis.

4. *Fassieh*®; Arabic Text Annotation Tool

Handling a massive raw text corpus targeted for factorization is a non trivial job for which *Fassieh*® has introduced the notion of a *book project*. A book consists of an arbitrary number of text pages with each in turn may contain an arbitrary bulk of text. A book project upon its creation contains no pages, whence pages may arbitrarily be added, named, re-named, and/or deleted. A page, like any word processing document, is empty upon its creation, whence text may be edited and/or imported from other text files. All the basic text editing functions are afforded with some extra control imposed upon the editing process in order to preserve the integrity of the page text represented internally in a word-based format rather than the simpler character-based format as in typical text editors.

The following three on/off switchable options [2] are significant examples among the ones that may be applied either on each page individually or on all the book pages:

I- *Text normalizer:* This facility first recognizes Arabic text blocks and separates them from non-Arabic ones. If enabled, it also parses and converts the non-verbal expressions; e.g. numerals, famous acronyms, salutations, time and date formats … etc. into verbal Arabic text. This parsing and synthesis take place according to a formal grammar.

II- *Misspelling handler:* If enabled, this facility broadens the possibilities of morphological analyses proposed for a given raw Arabic word to account for the famous misspellings commonly committed while casual writing. For neatly written Arabic scripts, this option is best switched off to avoid unnecessarily broadening of the morphological ambiguity.

III- *Word-level phonetic concatenator:* If enabled, this option tunes the produced Arabic diacritization to consider the mutual inter-word phonetic effects corresponding to the phonetic transcription of connected speech.

A command to run automatic Arabic morphological analysis associated with PoS tagging and diacritization over either a whole open book, only one page, or even a marked part of a page may be issued any time with current options and parameters setting assumed. Only Arabic text segments as recognized by the aforementioned Arabic text normalizer are factorized. The statistical disambiguation in this automatic mode is performed combinatorially as explained in section 3 above. These Arabic text factorizations together with the disambiguation process runs on a today's high-end PC at an avg. speed around 100 words per second with RAM (main & cash) memory acting as the critical speed determinant.

With the progress of working on a book project, each word in the book may have one of several possible states; *raw*, *automatically factorized*, *incorrect/transliterated*, *proof-read* … etc. *Fassieh*® designates a specific color to each of these states, and words are then rendered in these colors resulting in a visual illustration of the text status of any page. For an extra convenience at monitoring the project status, page-level as well as book-level statistics of words in each possible state are also provided.

Arabic annotators may browse the text in any page in order to proofread its factorization. For this purpose, all the possible morphological factorizations of the currently browsed word are displayed in a *grid* in the bottom of the page, as exemplified in fig. 4 on the next page, where each column of the grid details the elements of one possible analysis. These possibilities are ranked according to their statistical likelihoods in a descending order with a special marking on any previous automatic or manual selection. This ranking makes the proofreading quite an efficient process as the annotator need only to choose a factorization other than the proposed most likely one – displayed in the most right column – or the automatically selected one – if automatic disambiguation has been run - at a low frequency close to the small statistical disambiguation error margin below 5%.

To assist annotators examining the analyses while the proofreading esp. when apparently similar ones are proposed in the grid, auxiliary linguistic tools like Arabic morpheme dictionaries, and phonological character coloring are provided. By clicking any cell in the column detailing a proposed morphological analysis, the dictionary entry of the clicked morpheme is displayed in the dictionary area; see upper left corner of fig. 4. Each character of a pattern morpheme; (*t: f*) is also displayed in a color matching its phonological status.

Figure 4: A screen capture of *Fassieh*® while proofreading.

For the proposed morphological analysis in fig. 4:
Fig, 5 (Left) shows its PoS-tags vector, and Fig. 6 (Right) shows its possible semantic labels.

Figure 7: Adding a syntactic diacritic to the selected morphological analysis in fig. 4.

Moreover, the PoS-tags vector is mnemonically displayed by clicking on the upper cell of the proposed solution – see fig. 5 on the previous page.

For each automatically or manually selected morphological factorization, a set of possible semantic labels; i.e. word senses are proposed so that the annotator may select the most contextually proper one as suggested by fig. 6 on the previous page.

If a selected analysis may take a syntactic diacritic, only the possible diacritics are autonomously highlighted in an array of all the Arabic diacritics – see fig. 7 above - so that the annotator may optionally add it to the resulting phonetic transcription of the analyzed word.

Fassieh® then allows the export of any performed language factorization in output text files. Each of these files corresponds to one factorization of one book page. The produced analysis in each exported file is aligned with the raw text in its corresponding page. [21]

This section may be best concluded by a brief history of this tool. RDI www.RDI-eg.com started the development of the aforementioned Arabic factorization models since 1996 and is going on till the moment. [1], [2], [4], [5], [6], [7] The development of *Fassieh*® started during 1999 to produce the first version in late 2000 and the most recent one, called version 4, released on mid. 2008. The daily productivity of the trained annotator performing proofreading with an error margin ≤ 0.1% has moved up from about 1,000 words on version 1 of *Fassieh*® to about 3,000 words on its latest version.

Fassieh® has a considerable work history not only inside RDI where it produces the supervised factorized training corpora for our aforementioned stochastic modeling besides serving many other Arabic speech, multimedia, and IR applications, but also with other external parties. Software producers of those parties made use of its output in their Arabic text-oriented products, while the academic ones deployed it as a tutorial means in their NLP labs. Moreover, it has been successfully deployed in multinational R&D projects for the production of written Arabic language resources. [16], [21]

5. Conclusion

This paper has reviewed our fundamental Arabic language factorization models, namely; morphological analysis, phonetic transcription, Part-of-Speech tagging, and lexical semantics analysis. Coverage, compactness, and completeness have been shown to be main virtues of these models. The ambiguity problem due to the current lack of sound high-level language processing layers is also addressed and the statistical approach to its resolution has been discussed. Experimental disambiguation error margins realized by our stochastic modeling over the Arabic text factorized into its basic entities are presented and shown to be within the acceptable boundaries of industrial HLT applications.

One precious lesson we have learnt through all that work is to hybridize language factorization models with statistical methods as the best recipe for producing sound NLP systems.

The paper then introduced our text annotation tool *Fassieh*® that enables the production of large Arabic text corpora factorized according to the aforementioned models in a fully automatic mode with a narrow error margin, and in same time it allows supervised proof-reading of these factorizations for certain error intolerant tasks. A multitude of auxiliary linguistic tools (e.g. morpheme dictionaries, limited-context statistical ranking ... etc.) and illustrative GUI tools (e.g. character/word status coloring) are provided not only to make proof-reading as accurate and efficient as possible, but also to render this tool into an invaluable Arabic NLP demonstrative, tutorial and evaluation means.

6. References[2]

I. References in English:

[1] M. Attia, M. Rashwan, A. Ragheb, M. Al-Badrashiny, H. Al-Basoumy, S. Abdou, *A Compact Arabic Lexical Semantics Language Resource Based on the Theory of Semantic Fields*, Lecture Notes on Computer Science (LNCS): Advances in Natural Language Processing, Springer-Verlag Berlin Heidelberg; www.SpringerOnline.com, LNCS/LNAI; Vol. No. 5221, Aug. 2008.

[2] M. Attia, *Theory and Implementation of a Large-Scale Arabic Phonetic Transcriptor, and Applications*, PhD thesis, Dept. of Electronics and Electrical Communications, Faculty of Engineering, Cairo University, 2005.

[3] M. Attia, *Arabic Orthography vs. Arabic OCR*, Multilingual Computing & Technology magazine, USA, Dec. 2004.

[2] Ref.'s no. [1], [2], [3], [4], [5], [6], [7], [16], and [21] are freely downloadable under the section of *Papers on Natural Language Processing* at http://www.RDI-eg.com/RDI/technologies/papers.htm

[4] M. Attia, M. Rashwan, *A Large-Scale Arabic PoS Tagger Based on a Compact Arabic PoS Tags-Set, and Application on the Statistical Inference of Syntactic Diacritics of Arabic Text Words*, Proceedings of the Arabic Language Technologies and Resources Int'l Conference; NEMLAR, Cairo 2004.

[5] M. Attia, M. Rashwan, G. Khallaaf, *A Formalism of Arabic Phonetic Grammar and Application on the Automatic Arabic Phonetic Transcription of Transliterated Words*, Proceedings of the Arabic Language Technologies and Resources Int'l Conference; NEMLAR, Cairo 2004.

[6] M. Attia, M. Rashwan, G. Khallaaf, *On Stochastic Models, Statistical Disambiguation, and Applications on Arabic NLP Problems*, The Proceedings of the 3rd Conference on Language Engineering; CLE'2002, by the Egyptian Society of Language Engineering (ESoLE); www.ESoLE.org.

[7] M. Attia, *A Large-Scale Computational Processor of the Arabic Morphology, and Applications*, M.Sc. thesis, Dept. of Computer Engineering, Faculty of Engineering, Cairo University, 2000.

[8] V. Cavalli-Sforza, A. Soudi, T. Mitamura, *Arabic Morphology Generation Using a Concatenative Strategy*, ACM International Conference Proceeding Series; Proceedings of the first conference on North American chapter of the Association for Computational Linguistics (ACL), 2000.

[9] J. Dichy, M. Hassoun, *The DINAR.1 (DIctionnaire INformatisé de l'ARabe, version 1) Arabic Lexical Resource, an outline of contents and methodology*, The ELRA news letter, April-June 2005, Vol. 10 n.2, France.

[10] M. Diab, *The Feasibility of Bootstrapping an Arabic Word Net Leveraging Parallel Corpora and an English Word Net*, Proceedings of the Arabic Language Technologies and Resources Int'l Conference; NEMLAR, Cairo 2004.

[11] B. J. Grosz, K. S. Jones, B. L. Webber, *Readings in Natural Language Processing*, Morgan Kauffman publishers, 1986.

[12] M. Hearst, *Untangling Text Data Mining*, Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL), 1999; http://www.sims.Berkeley.edu/~hearst/papers/acl99/acl99-tdm.html

[13] D. Jurafsky, J. H. Martin, *Speech and Language Processing; an Introduction to Natural Language Processing, Computational Linguistics, and Speech Processing*, Prentice Hall, 2000.

[14] S. M. Katz, *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.

[15] A. Lehrer, *Semantic Fields and Lexical Structures*, Amsterdam-London, 1974.

[16] B. Maegaard, M. Attia, K. Choukri, S. Krauwer, C. Mokbel, M. Yaseen, *MEDAR – Collaboration Between European and Mediterranean Arabic Partners to Support the Development of Language Technology for Arabic*, LREC2008 conference http://www.lrec-conf.org/lrec2008, Marrakech-Morocco, May 2008.

[17] M. A. A. Rashwan, M. W. T. Fakhr, M. Attia, M. El-Mahallawy, *Arabic OCR System Analogous to HMM-Based ASR Systems; Implementation and Evaluation*, Journal of Engineering and Applied Science, Cairo University, www.Journal.eng.CU.edu.eg, Vol. 54 No. 6, pp. 653-672, Dec. 2007.

[18] A. Ratenaparkhi, *Maximum Entropy Models for Natural Language Ambiguity Resolutions*, PhD thesis in Computer and Information Science, Pennsylvania University, 1998.

[19] E. Riloff, R. Jones, *Learning Dictionaries for Information Extraction Using Multi-level Boot-Strapping*, Proceedings of AAAI-99.

[20] H. Schütze, C. D. Manning, *Foundations of Statistical Natural Language Processing*, the MIT Press, 2000.

[21] M. Yaseen, et al., *Building Annotated Written and Spoken Arabic LR's in NEMLAR Project*, LREC2006 conference http://www.lrec-conf.org/lrec2006, Genoa-Italy, May 2006.

II. References in Arabic:

[22] (Al-Waseett Dictionary, 1985) المُعْجَمُ الوَسيطُ، مَجْمَعُ اللُّغَةِ العَرَبِيَّةِ بالقاهِرة، الطَبْعةُ الثَّالِثَةُ، 1985م.

[23] (S. H. Al-Aany, 1983) فونولوجيا العربية، سَلْمَان حَسَن العانِي، ترجمة ياسِر المَلَّاح، دار النَّادي الأَدَبيِّ بجِدَّة–المَمْلَكة العَرَبيَّة السَّعوديَّة، 1983م.

[24] (A. Arragehy, 1993) التَّطبيقُ الصَّرْفِيُّ، عَبْدُهُ الرَّاجِحيّ، دارُ المَعْرِفةِ الجامِعِيَّةِ، الإسْكَنْدريَّةُ، 1993م.

[25] (S. Fayyaadh, 1990) الحُـقولُ الدِّلَالِيَّةُ الصَّرْفِيَّةُ لِلأَفْعالِ العَرَبيَّةِ، سُلَيْمان فَيَّاض، دارُ المَرِّيخِ بالرِّياضِ، 1990م.

[26] (A. H. Gabal, 1997) في علم الدلالة، عبد الكريم حسن جبل، دار المعرفة الجامعية، الإسكندرية، 1997م.

[27] (H. Ghaleb, 2003) كنز اللغة العربية، حنا غالب، لبنان ناشرون، 2003م.

[28] (M. I. Siny et al., 1993) المَكْنَز العربي المعاصر، محمود إسماعيل صيني–وآخرون، مكتبة لبنان، بيروت، الطبعة الأولى، 1993م.

[29] (A. M. Umar, 1998) عِلْمُ الدِّلَالةِ، أحمد مختار عمر، عالَمُ الكُتُب، الطَّبْعَةُ الخَامِسةُ، 1998م.

[30] [A. M. Umar et al., 2002–] المَكْنَزُ الكَبيرُ، أحمد مختار عمر وآخَرُونَ، دارُ نَشْرِ "سُطُور" المملكة العربية السعودية، الطَّبْعَة الأُولَى، 2002م.