International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. #, No.# (Mon-20..)

Nemlar Corpus Improvement for Arabic Natural Language Processing

Received ## Mon. 20##, Revised ## Mon. 20##, Accepted ## Mon. 20##, Published ## Mon. 20##

Abstract: Most machine learning approaches in Natural Language Processing rely mainly on corpora. Indeed, various applications based on this approaches require prior learning of statistical models, including the Hidden Markov Model for Part Of Speech Tagging. However, this learning resources must meet some criteria to have a well trained model, and thus more accurate results. On the other hand, we find that the Arabic language - despite its vast use on the internet and in social media - has a limited number of linguistic resources for machine learning, especially corpora with morpho- syntactic annotations. Thus, in this article we will treat the Nemlar corpus, one of the richest annotated linguistic corpora for the Arabic language. We will first present the content of this corpus. We will then define some criteria in order to improve its structure and enrich its content. We will also present the different modifications made on the original version, including merging POS tags, separating prefixes and suffixes, creating tags for specific cases, etc. in order to lead to the desired form. Then, we will see the experimentation evaluating the new word recognition rate. At the end, we will talk about the advantages and disadvantages of the resulting version.

Keywords: Corpus, Nemlar, Part Of Speech Tagging, Arabic language.

1. Introduction

Like a human being, the machine needs an information resource in order to learn. This resource can vary depending on the task that we want to teach to the machine. For NLP tasks, we then have a variety of training corpora for various languages. For the Arabic language, as one of the languages widely spoken in the world and used on the internet [1], we find a diversity of corpora. The simplest form: is raw corpora, containing raw text without annotations or other specific information, they are collected from various resources and generally have a large size, such as Tashkeela [2], King Saud University Corpus of Classical Arabic (KSUCCA) [3], and Open Source Arabic Corpora (OSAC) [4]. These corpora can be used for Text Classification, Text Summarization, Text Generation or for unsupervised NLP tasks. There are also Multilingual corpora such as [5] and [6], which are used for automatic translation between two or more languages. But most NLP tasks remain based on annotated corpora. This annotation can concern specific information for particular tasks such as Named Entity Recognition, like JRC-Names [7], or sentiment analysis, etc. But for more complicated linguistic tasks, we need linguistic corpora containing more detailed information concerning the syntactic and morphological analysis of words. Among this corpora, we find the Penn Arabic Tree Bank [8], The Quranic Arabic Corpus [9], KALIMAT [10] as well as the Nemlar corpus which represents the subject of this article.

However, the Arabic corpora, compared to the european ones, are still limited in size, coverage and availability, and even more so when we talk about special and annotated corpora. For this reason, we try in this work to develop and enrich one of the richest annotated linguistic corpora for the Arabic language: the Nemlar corpus, in order to optimize its use in stochastic NLP process, especially Arabic Part Of Speech tagging.

We will proceed, first, by a small presentation of the Nemlar corpus, its composition elements and its writing syntax.

A. *Corpus Presentation*

The NEMLAR (Network for Euro-Mediterranean Language Resources) project was launched between 2003 and 2005 to open a way to collaborate efforts in order to develop the resources of the Arabic language in the Mediterranean region. The project was supported by the European Union under the unco-MED program and had 14 partners from various countries [11].

The files are named as follows:

Domain label_Corpus label_order number_details[1].txt

For example: ScientificPress_Raw_03.txt and BroadcastNews_WithArabicPOS_Tags_01.txt.

It is an Arabic written corpus, annotated by RDI Egypt for NEMLAR Consortium [12]. It contains about 500,000 words from 13 various areas.

It consists of four types of corpora that we will describe more in section 1.2 (See Table 1).

Each type is represented by a folder containing 489 text files, belonging to 13 different domains (See Table 2)

TABLE I. NEMLAR CORPUS CONTENT

Corpus type	Corpus label
Raw text	*Raw*
Fully vowelized text	*WithArabicDiacritization*
Text with Arabic lexical analysis	*WithArabicLexicalAnalysis*
Text with Arabic POS-tags	*WithArabicPOS_Tags*

TABLE II. DOMAINS AND STATISTICS OF NEMLAR FILES

Domain	Domain label	Nbr. of files	Nbr.of words
Dictionary entries explanation	*ArabicDictionaries*	12	52,000
Arabic literature	*ArabicLiterature*	24	30,000
Text taken from Broadcast News (for TTS speakers DB LR)	*BroadcastNews*	4	5,500
Business	*Business*	10	20,000
General news	*GeneralNews*	159	100,000
Interviews	*Interviews*	18	56,000
Islamic text (Preaching and others)	*Islamic*	12	29,000
Legal domain text	*Legal*	10	21,000
Phrases of common words (for TTS speakers DB LR)	*Phrases OfCommonWords*	6	8,500
political debate	*PoliticalDebate*	22	30,000
political news	*PoliticalNews*	63	48,000
Scientific press	*ScientificPress*	51	50,000
Sports press	*SportsPress*	98	50,000
Total size:		**489**	**500,000**

B. *Contents*

We will try to give a clearer view of the corpus contents by browsing its various types.

1) *Raw Corpus*

For the first corpus, it contains only raw Arabic text with diacritics. For example:

[1] The ***details*** field describes more details and concerns the naming of the files belonging only to these three areas: *ArabicDictionaries*, *ArabicLiterature* and *PhrasesOfCommonWords*.

يتردد الآن في الكثير من وسائل الأعلام أخبار حول مؤتمرات المناخ والتدهور البيئي وارتفاع درجة حرارة الأرض والعديد من المصطلحات، التي تدل جميعها على أن المشكلة البيئية تتفاقم

2) Fully Vowelized Corpus

It contains the same texts but with the addition of some special characters describing pronunciation details (See Table 3).

TABLE III. SPECIAL CHARACTERS ADDED IN THE FULLY VOWELIZED CORPUS

Special character	Signific-ation	Description	Example
@	حرف علة مدِّي	Comes after avowel. («@ا»= *A* ; «@ي»= *I* ; «@و»= *O*)	فُلَا@نٌ
^	ألف مدية محذوفة (خنجرية)	*AlifMaddia* « *A* » pronounced and unwritten	ذ^لِكَ (ذالِكَ)
×	حرف غير منطوق	Letter written and non-pronounced	فِي× ا×ل×شَّيْءِ (فِـشَّيْءِ)
~	ألف مقصورة	*AlifMaqsura* « *A* » written as:ى	إِلَى~

The previous example will be in this corpus as follows:

يَتَرَدَّدُ ا×لْآنَ فِي× ا×لْكَثِي@رِ مِنْ وَسَا@ئِلِ ا×لْإِعْلَا@مِ أَخْبَا@رٌ حَوْلَ مُؤْتَمَرَا@تِ ا×لْمُنَا@خِ وَا×ل×تَّدَهْوُرِ ا×لْبِي@ئِيِّ وَا×رْتِفَا@عِ دَرَجَةِ حَرَا@رَةِ ا×لْأَرْضِ وَا×لْعَدِي@دِ مِنَ ا×لْمُصْطَلَحَا@تِ، اَلَّتِي@ تَدُلُّ جَمِي@عُهَا@ عَلَى~ أَنَّ اَ×لْمُشْكِلَةَ ا×لْبِي@ئِيَّةَ تَتَفَا@قَمُ

3) Text with Arabic lexical analysis

In this corpus we find the previous texts with additional information about the lexical analysis, including word type, prefix, root, pattern and suffix. Each word is represented as following (from right to left):

{SId (S),PtId (Pt),RId (R),PId (P):TId (T);W}

Where: **W**: vowelized word, **T**: Type, **TId**: Type Id, **P**: Prefix, **PId**: Prefix Id, **R**: Root, **RId**: Root Id, **Pt**: Pattern, **PtId**: Pattern Id, **S**: Suffix and **Sid**: Suffix Id.

Example:

{مُؤْتَمَرَا@تِ;(مصرَّفةٌ منتظِمة) 1:() 0,(أمر) 140,(مُفْتَعَل) 285,(ـا@ت) 27} {اَلْمُنَا@خِ;(مصرَّفةٌ منتظِمة) 1:(الـ) 9,(نوخ) 4066,(مُفَع@ل) 560,() 0} {وَا×ل×تَّدَهْوُرِ;(مصرَّفةٌ منتظِمة) 1:(وَا×لـ) 10,(دهور) 1401,(تَفَعْلُل) 22,() 0}

4) Text with Arabic POS-tags

In this version is added to the Fully vowelized texts information about the morphosyntactic analysis, that is called POS Tagging. For each word is associated a vector of POS tags composed by the tags of the prefixes that it contains (or Nullprefix), followed by the stem tags then the suffixes tags (or Nullsuffix) as following: **{(W)T1 T2 T3…}**

Where: **W**: the vowelized word and **T1**, **T2**, **T3**…: POS tags vector.

Example:

{(مُؤْتَمَرَا@تِ)NullPrefix Noun ObjNoun Plural Femin } {(اَلْمُنَا@خِ)Definit Noun ObjNoun NullSuffix } {(وَا×ل×تَّدَهْوُرِ)Conj Definit Noun NounInfinit Intransitive NullSuffix }

(See Appendix 3 for used tags and their meanings.)

After having an idea about the Nemlar corpus and its components, we will explain the improvements made on the corpus as well as the steps followed for the conception of its new version.

2. CORPUS IMPROVING FOR POS TAGGING

We move on to the main sections where we will describe the new structure and detail its design stages, the added elements, the encountered problems and the proposed solutions.

A. Corpus requirements for POS Tagging

Before talking about the conception, we will define some specifications on the learning resource, which are required generally in NLP processes and especially in POS Tagging learning process and which will represent the important points on which the work will be based:

a) First, the corpus form must be optimal to minimize execution time (no additional operations).

b) Each token[2], in the corpus must be tagged.

c) Each token must have just one tag.

d) Optimize the word recognition rate by maximizing word segmentation.

These points will be more explained in further sections.

1) Corpus format (in response to the requirement a)

First, to simplify the learning process from the corpus (given its large size: 489 files), it will be better to avoid

[2] To avoid the confusion between the grammatical and technical meaning of “word”, we used instead “token” to denote the entity having a tag in a tagged corpus (can be a word or part of a word). While “word” is used to denote the original corpus -unsegmented- entities (containing prefixes and suffixes).

training the statistical model directly from the text files. In fact, it may have redaction errors that cause shifts in reading and analyzing the files, which, consequently, will impact on the model operation. We thought then to adopt a tabular format by creating a csv file gathering the content of the all files, as in Table 4.

TABLE IV. FIRST CSV FORMAT OF THE CORPUS

Tokens	*Tag Vectors*			
Token$_1$	Tag$_1$(Token$_1$)	Tag$_2$(Token$_1$)	···	Tag$_n$(Token$_1$)
Token$_2$	Tag$_1$(Token$_2$)	Tag$_2$(Token$_2$)	···	Tag$_n$(Token$_2$)
…	…	…	···	…
Token$_m$	Tag$_1$(Token$_m$)	Tag$_2$(Token$_m$)	···	Tag$_n$(Token$_m$)

The tabular format will provide:

- A safer learning: the resource type provides a clearer and easier view for navigation and detection of errors and shifts.
- A faster learning: processing one csv file instead of several text files.
- An easier and faster handling of the corpus: the use of functionalities provided by spreadsheets and text editors (such as selection options, filtering, searching, browsing, etc.) for manual handling.
- The exploitation of libraries and predefined codes in different programming languages for handling csv (given its wide reputation and use).

2) Add tags for untagged tokens (in response to the requirement b)

We observed that there is a considerable number of untagged tokens (about 13% of the corpus tokens) that are either numbers, punctuation marks, URLs, etc. We thought then to use this information to further enrich the corpus. Therefore, we created for these tokens a set of special tags that are not included in the Nemlar tagset:

punct_mark: for punctuation marks: full stop, comma, brackets, etc.

- *num*: for numbers.
- *other*: for others.

Taking into account the additions described and the various exceptions, a program was developed to create this first corpus version in csv format that will be called *synt_corpus.csv* (See Table 5).

3) Tag interpretation (in response to the requirement c)

Generally, each word is composed of prefix(es), stem and suffix(es). And as we have seen, the corpus gathers the tags of stems and affixes (prefixes and suffixes). So that the tagger gives each part of a word its own tag, it must recognize -at the learning step- each part and its independent tag. The problem is that in the Nemlar tagged corpus words are not segmented: a tag sequence is given to the entire set: prefix(es), stem and suffix(es). Moreover, the stems often have more than one tag (See for example Table 6).

We then consider 4 types of tags (See Appendix 3 for the meaning of each tag[3]):

Prefix_tag = *NullPrefix, Conj, Confirm,Interrog, Definit, Present, Future*

Suffix_tag = *NullSuffix, ObjPossPro, PossessPro, RelAdj, Femin, Masc, Single, Binary,Plural, Adjunct, NonAdjunct, MANSS_MAGR, MAGR, SubjPro, ObjPro,*

TABLE V. SAMPLE OF *SYNT_CORPUS*

Tokens	Tokens	Tokens	Tokens	Tokens	Tokens	Tokens	Tokens
IslamicTopics_07.txt	387	وَيُعْرِضَ	Conj	Present	Active	Verb	NullSuffix
IslamicTopics_07.txt	387	عَنْ	NullPrefix	Prepos	NullSuffix		
IslamicTopics_07.txt	387	اَلْقَوْمِ	Definit	Noun	NullSuffix		
IslamicTopics_07.txt	387	~حَتّى	NullPrefix	Prepos	ParticleNAASSIB	NullSuffix	

TABLE VI. TAGGING OF THE WORD وَنَحْتَرِمُهُ IN THE ORIGINAL CORPUS

Word	Tag sequence	Prefix tags		Stem tags		Suffix tag
وَنَحْتَرِمُهُ	Conj Present Active Verb ObjPro	Conj (و)	Present (نَ)	Active (احترم)	Verb (احترم)	ObjPro (ـهُ)

[3] We mention that the tags *SOW* and *Padding* are listed in the tag list of the documentation provided with Nemlar corpus but do not exist in the corpus, while the tag *CondNotJAAZIMA* exists in the corpus and does not appear in the list.

MANS_MAJZ

Stem_tag = *MARF, MANSS, Noun, NounInfinit, NounInfinitLike, SubjNoun, ExaggAdj, ObjNoun, TimeLocNoun,NoSARF, Active, Passive, Imperative, Verb, Intransitive, MAJZ, Past, PresImperat, Prepos, Interj, PrepPronComp, RelPro, DemoPro, InterrogArticle, JAAZIMA, CondJAAZIMA, CondNot, JAAZIMA, LAA, LAATA, Except, NoSyntaEffect, DZARF, ParticleNAASIKH, VerbNAASIKH, ParticleNAASSIB, MASSDARIYYA, CondNotJAAZIMA*

And the Special tags that are added for untagged tokens (except the tag Translit given to transliterated words):

Special_tag= *start, num, punct_mark, other, Translit*

So, returning to the previous example, we can see that the stem احترم has two tags: *Active* and *Verb*. Hence if we create a simple tagset based on the separated tags we will have a tag conflict, because the stem must have only one tag (requirement c).

We have so concatenated such tags (associated to the same entity) in one tag by adding "+".

In the previous example, the stem will have this tag: *Active + Verb*.

Consequently, the new tagset will consist of separated and concatenated tags (which will increase the tagset length). So that, for a parsed input word, each part (prefix(es), stem and suffix(es)) will have its own tag.

4) Reformulation of content (in response to the requirement d)

a) Word division–segmentation

To tag each part of the word separately, the words must be divided and tagged separately since the learning stage (in the tagged corpus), while the corpus contains un-segmented words. Let's take another example: {(وَا×لْأَبْجَدِيَّةُ)Conj Definit Noun NounInfinit RelAdj Femin Single We have in this example 7 tags attributed to the whole word. But referring to have one tag per token, we should have a form as in Table 7.

TABLE VII. DESIRABLE FORM OF THE CORPUS

Tokens	Tags
وَ	*Conj*
ال	*Definit*
أبجد	*Noun+NounInfinit+Femin* (created tag)
يَّ	*RelAdj*
ة	*Single*

Therefore, we will need for that to parse each word in the corpus and define its parts.

b) Delimitation of prefixes and suffixes

We chose to avoid the use of automatic parsers, because it may introduce analysis errors and, in addition, requires that the parser had the same Nemlar syntax (for example Al Khalil analyzer does not consider the prefix "ا" in "اضربه"). For this, we thought first time to do statistics on prefix and suffix tags and there original words in order to parse the words according to an affix-tag list (e.g. Confirm: (ل) Conj (و ف بل أو ثم أم لكن أو) defines (ال) ...) but it turns out that it is quite difficult to browse wholes cases to extract tags (e.g. "Binary" is assigned to 1134 words and "Adjunct" is assigned to 810 and take various forms). So we sought in the other types of the corpus to see if they contain a manual parsing of words.

We then found that the lexically analyzed corpus contains information about affixes, as following:

{ (فَعْلَل),1 (أبجد),10 (وَا×لـ):1 (مصرَّفةٌ منتظِمة);وَا×لْأَبْجَدِيَّةُ 28 (ـيَّة),26}

So we can define and extract prefix and suffix parts of each word in the corpus. To do that, we must have the same format as the *synt_corpus* to facilitate and ensure the correspondence between words in the two corpora.

3. INSERTING LEXICAL INFORMATION

A. Creation of the lexical corpus lex_corpus

We created the *lex_corpus* as we have already done with the first corpus, but taking into account the new syntax of Nemlar lexically analyzed corpus. A csv file was created as in Table 8.

B. Mixing the lexical and syntactic corpora

To ensure the correspondence between the lexical and syntactic information for each word, we tried to create a corpus mixing the two previous corpora. However, we encountered some problems.

1) Shift problem

We have, in total, 577,054 words (lines) in *lex_corpus* and 576,445 words in the *synt_corpus* (after including special tags) which makes a shift of 600 words. This has obliged us to find an identification way of each word, which can be common to the two corpora. We then adopted the two variables: file name and line number.

2) Ignored differences

There are some cases where we found a slight difference between the two corpora which can cause errors in the correspondence and whose negligence will not have a great impact on the final corpus.

3) Empty lines

We have removed the empty lines because we found differences in the number of line breaks between the two corpora.

4) Punctuation marks

We found also sometimes a punctuation mark appears in a location of a corpus and does not appear in the corresponding location in the other corpus;

For example, in *lex_corpus*:

File	Line	word
ArabicLiterature_07....txt	11	حَيَا@تِهِ
ArabicLiterature_07....txt	11	وَلَا@
ArabicLiterature_07....txt	11	يَزَالُ

In *syn_corpus*:

File	Line	word
ArabicLiterature_07....txt	11	حَيَا@تِهِ
ArabicLiterature_07....txt	11	,
ArabicLiterature_07....txt	11	وَلَا@
ArabicLiterature_07....txt	11	يَزَالُ

5) Words manually changed

Then it remained a few cases that we dealt with manually:

Lex_corpus	***Synt_corpus***
قُوَى	~قُوَى
تَكْرَارِ	تِكْرَا@ر
كَوِيَّتَيْهِ	كُوَيْتِيَّةٍ
@فَلَا	@فَلَا (with space)
وَاتْ	تْ ; ا ; وَ (separated)
اَلَا@سُوجِي	سُوجِي ; ال ;ا (separated)

C. Mixing result

As result of mixing the two corpora, we have a larger corpus gathering both syntactic and lexical information. It is thus possible to determine for each word its prefix and suffix parts, and so, we are close to the envisaged form, previously described.

A sample of the mixed corpus is presented in Table 9.

D. Affixes separation

At this stage we can treat concatenated prefixes and suffixes as done with stems: gather tags of each type by "+". For example the word وَالْأَبْجَدِيَّةُ will be parsed and tagged as follows:

وَالـ	*Conj+Definit*
أبجد	*Noun+NounInfinit*
ـيَّة	*RelAdj+Femin+Single*

TABLE VIII. SAMPLE OF *LEX_CORPUS*

File	Line	Word	Type	T Id	P	P Id	R	R Id	Pt	Pt Id	S	S Id
ArabicDict…txt	3	وَا×لْيَا@بِسُ	مصرَّفةٌ منتظِمة	1	وَا× لـ	10	يبس	4419	فَا@عِل	805		0
ArabicDict...txt	3	وَهُوَ	جامِدة	3	وَ	1	هُوَ	75	هُوَ	8		0
ArabicDict...txt	3	اَلْمَرْعَى~	مصرَّفةٌ منتظِمة	1	الـ	9	رعي	1613	مَفْعَل~	797		0
ArabicDict...txt	3	كَذَ^لِكَ	جامِدة	3	كَـ	17	ذَا@	31	ذَ^لِ	76	ـكَ	3

TABLE IX. SAMPLE OF THE MIXED CORPUS

	***Lex_corpus* information**										***Synt_corpus* information**						
Word	***T***	***TId***	***P***	***PId***	***R***	***RId***	***Pt***	***PtId***	***S***	***SId***	***Tag1***	***Tag2***	***Tag3***	***Tag4***	***Tag5***	***Tag6***	***Tag7***
وَالْأَبْجَدِيَّةُ	مصرَّفةٌ منتظِمة	1	وَالـ	10	أبجد	1	فَعْلَل	26	ـيَّة	28	Conj	Definit	Noun	NounInfinit	RelAdj	Femin	Single
اَلْعَرَبِيَّةُ	مصرَّفةٌ منتظِمة	1	الـ	9	عرب	2628	فَعَل	821	ـيَّة	28	Definit	Noun	RelAdj	Femin	Single		
هِيَ	جامِدة	3		0	هُوَ	75	هِيَ	9		0	NullPrefix	Noun	SubjPro	NullSuffix			

But we opted to separate all parts of the word including concatenated affixes. Indeed, as already mentioned in corpus requirements (d), the division of words enriches the corpus because the number of possible combinations of 5 elements (و+ال+أبجد+ي+ة) is greater than the combination of 3 elements (وال+أبجد+ية). And we will see that in graphics later.

Thus, another process will be required: is to separate concatenated prefixes and suffixes:

وَ → وَال (conj) + ال (Definit)

ـيَّ → ـيَّة (RelAdj) + ة (Femin+Single)

1) *Prefixes separation*

For the prefixes, we browsed the prefixes and its tags after removing duplicates cases. Then we created a list of unit prefixes (i.e. consisting of a single prefix) (See appendix 2). Thus for each prefix group, we consult this list and compare the possible combinations prefix/tag to draw the appropriate division[4].

2) *Suffixes separation*

For the suffixes, we found some difficulties: first, even for single suffixes, we found the same problem found in stems: a suffix can have multiple tags, we then thought to proceed by gathering these tags - as was done for stems- by concatenation and addition of "+". But even after removing duplicates and filtering by the suffix id, we found suffixes with multiple tags. For example:

Suffix	Id	Tags
ـة	26	*Femin+Single*
ـة	26	*Plural+Femin+Single*
ـة	26	*Adjunct+Femin+Single*
ـة	26	*Plural+Masc+Femin+Single*
ـة	26	*Masc+Single+Femin*

To deal with these cases, we thought at first to consider as tag: the common part that is repeated in all possible tags. However, we saw that this will depend more on the frequency of occurrence. In fact there are exceptions that come from rare cases, so we will not affect thousands of cases for one or two cases. So, we built a function that calculates the frequency of each suffix (with Id) and creates a new suffix table as follows:

Suffix	Id	Tags	Freq.
ـة	26	*Femin+Single*	51,274
ـة	26	*Plural+Femin+Single*	1,188
ـة	26	*Adjunct+Femin+Single*	88
ـة	26	*Plural+Masc+Femin+Single*	3
ـة	26	*Masc+Single+Femin*	40

Then we filter out less frequent tags to leave only one tag by id [5].

After that, as done for the prefixes, we worked in suffix separation in order to have, in case of multi suffix, each suffix with its own tag. We tried to do this in the same way followed for prefixes, but we found that the combination of suffixes and their tags are very complicated and difficult to extract, which has obliged us to do the suffixes separation manually.

First, we distinguish unit suffixes (containing just one suffix). For the multi suffix, we check the suffixes delimitation by consulting the original word in the corpus to decide where introduce the division character (the space) between the suffixes as well as between the tags, by referring to the list of unit suffixes.

For example, before separation:

Suffix	Tags
ـتَيْهِمْ	*Femin+Binary+MANSS_MAGR+Adjunct+PossessPro*

After separation:

Suffix	Tags
ـتَ يْ هِمْ	*Femin Binary+MANSS_MAGR+Adjunct PossessPro*

For the unit suffixes, there are those that are predefined in the Nemlar corpus (they were mentioned as a single suffix with their ids) and we have added other unit suffixes manually, in case we see that a suffix can be further divided, for example:

Suffix	Id	Tags
ـا@	-	*Binary*
يْ	-	*Binary*
ن	-	*NonAdjunct*

[4] We mention here some rare errors found in Nemlar prefixes:

- There are some past verbs for which is considered the prefix “وأ” while the correct prefix (that we consider) is: “و”. Such as: وَأُضِيفَ, وَأَنْهَى, وَأَخْضَعَ, وَأَنْذَرَ.
- For the word آفاقه it was considered the prefix “أا” which is not correct (this word has no prefix). It was then removed.

[5] Assuming that, in writing Nemlar, it was associated for each set of suffixes a unique id.

In this example, we considered *NonAdjunct* as a unit tag because after browsing all the suffixes we find that this is the case for sub-suffixes نِ and نَ such as: تانِ, ينَ, etc. because, in the Arabic language, the meaning of *NonAdjunct* is that the word is not syntactically adjunct to another and this is marked by the suffix ن (ثبوت النون) for the dual and plural masculine [13].

4. THE RESULTING CORPUS

At this stage we come to the proposed structure of the corpus:

Token	Type	T. Id	Root	R. Id	Pattern	P. Id	Tag
و							*Conj*
ال							*Definit*
أبجد	مصرفة منتظمة	1	أبجد	1	فَعْلَل	26	*Noun+ NounInfinit*
ي							*RelAdj*
ة							*Femin+Single*
ال							*Definit*
عرب	مصرفة منتظمة	1	عرب	26 28	فَعَل	821	*Noun*
ي							*RelAdj*
ة							*Femin+Single*

We left the root and pattern information for the stem[6], and we removed diacritics to use the corpus in learning POS tagging where the tokens to tag are not often vowelized.

A. *Delimitation of lines, sentences and words*

The definition of the beginnings and ends of lines, sentences and words, represents additional information contained in the original Nemlar texts. This additional information must be also contained in the new corpus. Since this information appears in text format, we have chosen some special characters to make it apparent in the table format. So, we defined the special tokens <l> and </l> (tagged *line_start* and *line_end*) to mark the beginning and the end of a line. For sentences, we defined punctuation marks determining the end of a sentence (meaning termination) in order to include the special tokens </s> tagged *sentence_end* and <s> tagged *sentence_start* (to mark the end of the current sentence and the beginning of the next sentence). For words, we found that after the parsing, we have to gather parts of the same word in a single set, so we added the special tokens: <w> tagged *word_start* and </w> tagged *word_end* to delimit the words that were part of the same word before the separation.

The previous example in the new corpus will be as in Table 8.

TABLE X. SAMPLE OF THE FINAL RESULTING CORPUS

Token	Type	T. Id	Root	R. Id	Pattern	P.Id	Tag
<l>							*line_start*
<s>							*sentence_start*
<w>							*word_start*
و							*Conj*
ال							*Definit*
أبجد	مصرفة منتظمة	1	أبجد	1	فَعْلَل	26	*Noun+NounInfinit*
ي							*RelAdj*
ة							*Femin+Single*
</w>							*word_end*
ال							*Definit*
…							…
</s>							*end*

B. *Corpus size and word detection rate*

After the separation of affixes, the size of the lexicon is considerably increased. Consequently, word detection rate will also increase. To view this increase according to the corpus domains, we have conducted an experiment on 40 sentences from different domains. For every sentence, each word was manually parsed. We subsequently partitioned the corpus according to the 13 domains and executed on each partition a program that checks the existence of each token. We collected the experiment results which are displayed on the graphs of figure 2 and figure 3.

Here are some general statistics:

The original corpus length	500,000
New corpus length (number of tokens) without special characters (<s>, <w>, etc.)	902,864
Total new corpus length (including the special characters)	2,003,250
Input sentence number	40
Input words number (for calculating the original corpus detection rate)	1,220
Input tokens number -after manual parsing- (for calculating the new corpus detection rate)	2,380

[6] Because the division operation cuts the stem from the word, and often gives ambiguous stems, e.g. the stem صفـfrom the word الصفة.

We can see that when the number of words increases, the margin between the sizes becomes more important. It can be explained by the fact that each time a word is added, the parts of words become more important, because often one single word can generate several parts (prefix(es), stem and suffix(es)).

However, in some domain, this proportion is not evident. We can explain this by the fact that in some domains, especially *ScientificPress* and *SportsPress*, we often encounter transliterated words (such as: اَلتِّكْنُولُوجْيَا, اَلْفِيزْيَاءْ, etc. in *ScientificPress* and مِيلَانُو, رُونَالْدُو, etc. in *SportsPress*) and such words cannot be parsed, thus we will not have a significant margin.

C. Benefits and characteristics of the new corpus

We can mention here some benefits and differences that we can find in the new corpus, in comparison to the original one:

- An appropriate format providing a clear and easy readability of data: the user can read raw text (without tags) just by traversing the column of tokens from top to bottom. While in the original version, tokens are mixed with tags and other information.
- Easy and flexible data manipulation: as already mentioned, the user can enjoy a variety of features provided by spreadsheets and text editors that facilitate data browsing, searching and editing. For example, to extract the tagset from the new corpus, user can simply eliminate redundancies in the tag column and copy the result in a new table; which will take much time if we want to program it for the old corpus version.
- All the data are grouped in a single file, so the user can access all the information (lexical and syntactic), while in the original corpus the data is dispersed in 978 files (489 files in both lexical and syntactic corpora).
- Optimal use of the corpus data by adding special tags (*num*, *punct_mark* and *other*), while in the original corpus, the tokens having these tags did not make sense.
- Separation of prefixes and suffixes, and their tags, which gives more semantic content. Indeed, this separation, gives more information than assigning a sequence of tags to the set of word's parts.
- Reducing the time execution of corpus analysis: we will not traverse multiple files and we will not need to do text processing required to extract the data from this files.
- Ability to integrate the data into a various database formats.

D. Weaknesses

- Big size of the file, which requests an application

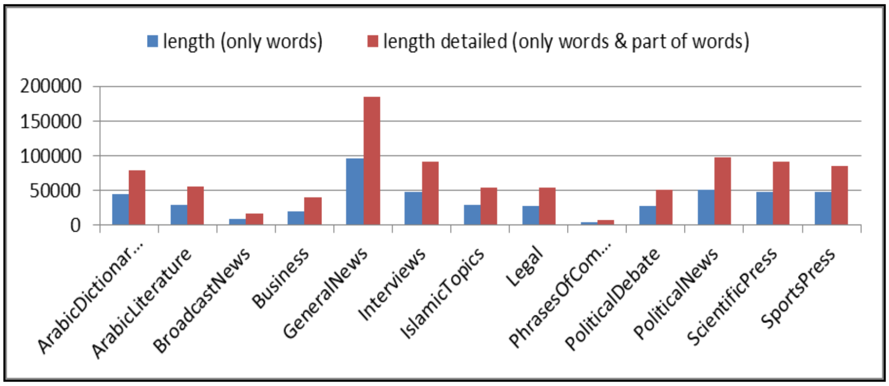

Figure 1. Change of the lexicon size before and after separation of the affixes

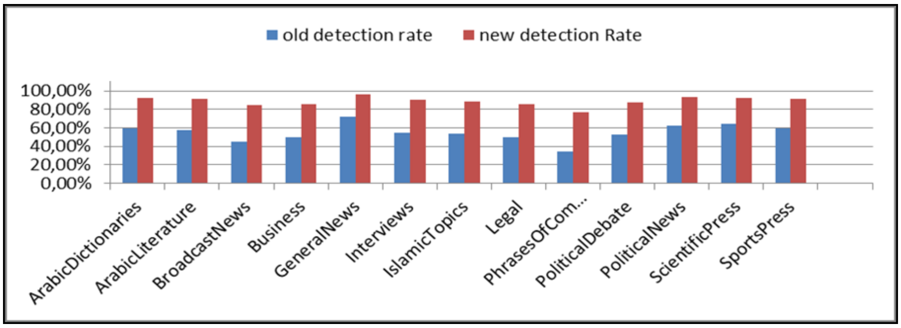

Figure 2. Change of word detection rate before and after separation of the affixes

memory and can cause transportation difficulties.

- Loss of some information when mixing the two corpora (lexical and syntactic) to ensure the correspondence.
- Repetition of the names of files. In fact, we tried to keep the file titles for the reason of text classification (each title of a file represents a classification of its texts). For this, the title is repeated in all the tokens belonging to the same file. However, we can replace these titles by Ids to reduce the file size.
- Big tagset: the tagset size becomes greater, due to the introduction of special tags, in addition to the new tags generated after concatenation of tags, which will make the processes heaviest.

5. Conclusion

We can conclude that we have led to a final version of the corpus that responds to the defined requirements, where every token is tagged, and each word part (prefix, suffix or stem) has its own and unique tag in which we tried as much as possible to keep the original corpus information.

We have seen the advantages of using the table format where several provided utilities can be used to organize and handle the corpus contents, in particular with large data. We have also seen the contribution of parsing words in increasing the corpus size which reached 80.57% (without counting the added special characters and untagged tokens). Consequently, we have also an increase in word detection rates that we saw on graphics.

To keep all the tagging information, we were forced to gather the tags of the same element as a single tag, which generated a large tagset. But we can reduce this tagset by replacing less frequent tags by the closest frequent tags. For example, replace all tags starting with *Active+Verb* (for example, *Active+Verb+MANSS*) by the frequent tag *Active+Verb*. By doing so, the complexity of learning and tagging will be remarkably reduced (since the model matrices' sizes will be also reduced).

We have so reached an advanced and enriched version of the Nemlar corpus that will be easier to handle for researchers in the field of natural language processing.

We can also take into consideration the followed steps and the designed architecture in designing new resources or optimizing other existing ones.

References

[1] M. Diab, N. Habash, and I. Zitouni, “NLP For arabic and related languages,” Traitement Automatique des Langues, vol. 58, no. 3, pp. 9-13, 2017.

[2] T. Zerrouki, and A. Balla, “Tashkeela: Novel corpus of Arabic vocalized texts, data for auto-diacritization systems,” Data in brief, vol. 11, pp. 147-151, 2017.

[3] M. Alrabiah, A. Al-Salman, and E. S. Atwell, “The design and construction of the 50 million words KSUCCA,” In Proceedings of WACL’2 Second Workshop on Arabic Corpus Linguistics, The University of Leeds, pp. 5-8, 2013.

[4] M. K. Saad, and W. Ashour, “Osac: Open source arabic corpora,” In 6th ArchEng Int. Symposiums, EEECS, Vol. 10, p. 55, 2010.

[5] D. Samy, and A. González-Ledesma, “Pragmatic Annotation of Discourse Markers in a Multilingual Parallel Corpus (Arabic-Spanish-English),” In LREC, May.

[6] A. Rafalovitch, and R. Dale, “United nations general assembly resolutions: A six-language parallel corpus,” In Proceedings of Machine Translation Summit XII: Posters, 2009.

[7] R. Steinberger, B. Pouliquen, M. Kabadjov, and E. Van der Goot, “JRC-Names: A freely available, highly multilingual named entity resource,” arXiv preprint arXiv:1309.6162, 2013.

[8] M. Maamouri, A. Bies, T. Buckwalter, and W. Mekki, “The penn arabic treebank: Building a large-scale annotated arabic corpus,” In NEMLAR conference on Arabic language resources and tools, vol. 27, pp. 466-467, 2004.

[9] K. Dukes, and N. Habash, “Morphological Annotation of Quranic Arabic,” In Lrec, pp. 2530-2536, 2010.

[10] M. El-Haj, and R. Koulali, “KALIMAT a multipurpose Arabic Corpus,” Culture, vol. 2, p. 1-359, 2013.

[11] B. Maegaard, “The NEMLAR project on Arabic language resources,” In Proceedings of the 9th EAMT Workshop: Broadening horizons of machine translation and its applications, pp. 124-128, 2004.

[12] M. Yaseen, M. Attia, B. Maegaard, K. Choukri, N. Paulsson, S. Haamid, ... and A. Ragheb, “Building Annotated Written and Spoken Arabic LRs in NEMLAR Project,” In LREC, pp. 533-538, 2006.

[13] M. Badreddine, “شرح ابن الناظم على ألفية ابن مالك (Explanation of Alfiat Ibn Malik by the son of its author),” DAR al-KOTOB al-ILMIYAH, Beirut, Lebanon, 1st ed., 2000.

APPENDICES

APPENDIX I. UNIQUE (INDIVISIBLE) PREFIXES

Prefix	Tag
ا	*Imperative*
أ	*Present*
أ	*Interrog*
أ	*Present*
أ	*Present*
ال	*Definit*
بِ	*Prepos*
ت	*Present*
تُ	*Present*
سَ	*Future*
فَ	*Conj*
فَ	*ParticleNAASSIB*
كَ	*Prepos*
لَ	*Confirm*
لِ	*Prepos*
لِ	*ParticleNAASSIB*
نـ	*Present*
نُـ	*Present*
وَ	*Conj*
وَ	*Prepos*
يـ	*Present*
يُـ	*Present*

APPENDIX II. UNIQUE SUFFIXES (MANUALLY DEFINED)

Suffix	Tag
ـا@	*Binary*
يْ	*Binary+MANSS_MAGR*
يْ	*Binary+MANSS_MAGR+Adjunct*
ن	*NonAdjunct*
ـكِ	*ObjPro*
ـكِ	*PossPro*
ت	*Femin+Single*
ـتِ	*Femin*
ـتِ	*Femin+Single*
ـكُمْ	*ObjPro*
ـكُمْ	*PossPro*
ـكُمَا@	*ObjPro*
ـكُمَا@	*PossPro*
ـكُنَّ	*ObjPro*
ـكُنَّ	*PossPro*
ـنَّ	*AffirmNoon*
ـنَا@	*ObjPro*
ـنَا@	*PossPro*
كَ	*ObjPro*
كَ	*PossPro*
ـه	*ObjPro*
ـه	*PossPro*
ـهَا@	*ObjPro*
ـهَا@	*PossPro*
ـهمْ	*ObjPro*
ـهمْ	*PossPro*
ـهمَا@	*ObjPro*

ـهمَا@	*PossPro*
ـهنَّ	*ObjPro*
ـهنَّ	*PossPro*
ـو@	*Plural+Masc*
نَ	*NonAdjunct*
ـو	*SubjPro*
ـو	*MANS_MAJZ+SubjPro*
هُنَّ	*ObjPro*
هُ	*ObjPro*
هُمَا@	*ObjPro*
ـي@	*Plural+Masc+MANSS_MAGR*
نَا@	*PossPro*
هِ	*PossPro*
هِمْ	*PossPro*
ا@تِ	*Plural+Femin*
نِ	*NonAdjunct*
تَ	*Femin*
ـي	*SubjPro*

APPENDIX III. THE NEMLAR TAGSET

Category	Mnemonic	Meaning in English	Meaning in Arabic
Start of word marker	*SOW*	Start-Of-Word marker	بِدايةُ كَلِمة
Pad-ding string	*Padding*	Padding string	حَشْو
Features of noun and verb prefixes	*NullPrefix*	Null prefix	لا سابِقَ
	Conj	Conjunctive	عَطْف
	Confirm	Confirmation by Laam	لامُ التَّوكيد
	Interrog	Interrogation by Hamza	هَمْزةُ الاستفهام
Features of noun and verb suffixes	*NullSuffix*	Null suffix	لا لاحِقَ
	ObjPossPro	Object or possession pronoun	ضَميرُ نَصْبٍ أو جَرٍّ
Verb and noun syntactic cases	*MARF*	1st Arabic syntactic case	مرفوع
	MANSS	2nd Arabic syntactic case	منصوب
Features of noun-only prefixes	*Definit*	Definitive article	"ال" التَّعريف
Features of noun-only stems	*Noun*	Nominal	اِسْم
	NounInfinit	Nouns made of infinitives	مَصْدَر
	NounInfinitLike	"NounInfinit" like	اِسْمُ مَصْدَر
	SubjNoun	Subject noun	اِسْمُ فاعِلٍ
	ExaggAdj	Exaggeration adjective	صِيغةُ مُبالَغة
	ObjNoun	Object noun	اِسْمُ مفعولٍ
	TimeLocNoun	Noun of time or location	اِسْمُ زَمَانٍ أوْ مَكَانٍ
	NoSARF	An Arabic feature of a specific class of nouns	ممنوعٌ مِنَ الصَّرْفِ
Features of noun-only suffixes	*PossessPro*	Possessive pronoun	ضَميرُ جَرٍّ
	RelAdj	Relative adjectives maker	نَسَب
	Femin	Feminine	تأنيث
	Masc	Masculine	مذكّر
	Single	Singular	مُفْرَد
	Binary	Binary	مثنّى

	Plural	Plural	جَمْع
	Adjunct	Adjunct	مُضَاف
	NonAdjunct	Non Adjunct	غَيْرُ مُضَاف
	MANSS_MAGR	2nd or 3rd Arabic syntactic case	منصوبٌ أو مجرور
	MAGR	3rd Arabic syntactic case	مجرور
Features of verb-only prefixes	*Present*	Present tense	مُضارِع
	Future	Future tense	استقبال
Features of verb-only stems	*Active*	Active sound	مَبْنِيٌّ للمعلوم (للفاعل)
	Passive	Passive sound	مَبْنِيٌّ للمجهول (للمفعول)
	Imperative	Imperative	أمْر
	Verb	Verb	فِعْل
	Intransitive	Intransitive verb	لازِم
	MAJZ	4th Arabic syntactic case	مجزوم
	Past	Past tense	ماضٍ
	PresImperat	Present tense, or imperative	مُضارِعٌ أو أمْر