ALP: An Arabic Linguistic Pipeline

Abed Alhakim Freihat, Gábor Bella, Mourad Abbas, Hamdy Mubarak, and Fausto Giunchiglia

Abstract This paper presents ALP, an entirely new linguistic pipeline for natural language processing of text in Modern Standard Arabic. In contrary to the conventional pipeline architecture, we solve common NLP operations of word segmentation, POS tagging, and named entity recognition as a single sequence labeling task. Based on this single component, we also introduce a new lemmatizer tool that combines machine-learning-based and dictionary-based approaches, the latter providing increased accuracy, robustness, and flexibility to the former. In addition, we present a base phrase chunking tool which is an essential tool in many NLP operations. The presented pipeline configuration results in a faster operation and is able to provide a solution to the challenges of processing Modern Standard Arabic, such as the rich morphology, agglutinative aspects, and lexical ambiguity due to the absence of short vowels.

1 Introduction

Natural language understanding tasks, such as information retrieval [1], word sense disambiguation [2, 3], question answering [4], or semantic search [5], are usually built on top of a set of basic NLP preprocessing operations. These operations are supposed to bring text to a more canonical form with dictionary words (lemmas) and named entities clearly identified. The precise solutions applied depend greatly on the language; however, state-of-the-art approaches typically involve a pipeline

A. A. Freihat (✉) · G. Bella · F. Giunchiglia
University of Trento, Trento, Italy
e-mail: abed.freihat@unitn.it; Gabor.Bella@unitn.it; fausto.giunchiglia@unitn.it

M. Abbas
High Council of Arabic Language, Algiers, Algeria

H. Mubarak
Hamad Bin Khalifa University, Ar-Rayyan, Qatar
e-mail: hmubarak@hbku.edu.qa

M. Abbas (ed.), *Analysis and Application of Natural Language and Speech Processing*, Signals and Communication Technology,
https://doi.org/10.1007/978-3-031-11035-1_4

of components, such as a part-of-speech tagger, a morphological analyzer, a lemmatizer, and a named entity recognizer (NER). Compared to English, both lemmatization and NER are harder for Arabic text: for the former because of the inflectional complexity and ambiguity inherent to written language and for the latter mainly because Arabic does not mark named entities by capitalization.

There has been extensive research on each of the tasks mentioned above. In the case of Arabic POS tagging, the approaches are typically based on statistical classifiers such as SVM [6, 7], sometimes combined with rule-based methods [8] or with a morphological analyzer [9–11]. The idea of POS tagging applied to unsegmented words has been investigated in [10] and in [12].

For NER, several solutions and tools have been reported. They can be classified as rule-based systems such as the approach presented in [13], machine-learning-based ones such as [14, 15], and hybrid systems such as [16]. The correlation between NER and POS tagging is illustrated in [17].

For Arabic lemmatization, while several approaches were proposed, few tools are actually available. Existing tools typically combine multiple techniques to achieve efficient lemmatization. The *Alkhalil* lemmatizer [18] first applies morpho-syntactic analysis to the input sentence in order to generate all potential word surface forms. Then, among these, only one form is selected per word using a technique based on hidden Markov models. The accuracy of the tool is reported to be about 94%. Another lemmatizer is MADAMIRA [19] which relies on preliminary morphological analysis on the input word that outputs a list of possible analyses. As a second step, it predicts the correct lemma using language models. The accuracy of the tool is 96.6%. The FARASA lemmatizer [20] uses a dictionary of words and their diacritizations ordered according to their number of occurrences. The accuracy reported for FARASA is 97.32%. Besides these tools, there are other proposed approaches: for example, [21] proposes a pattern-based approach, while [22] and [23] present rule-based solutions.

For Arabic chunking tools, the only available research on phrase chunking is the work done by Mona Diab who introduced a chunking tool as a component of MADAMIRA. The adopted approach and details about the tools are described in [24]. The reported accuracy of the tools is 96.33%. In terms of overall NLP pipeline architecture, most existing solutions perform the aforementioned tasks as a cascade of several processing steps. For example, POS tagging in FARASA [20, 25] and in MADAMIRA supposes that word segmentation has been done as a previous step. Segmentation, in turn, relies on further preprocessing tasks such as morphological analysis in MADAMIRA.

Likewise, NER and lemmatization are often implemented as separate downstream tasks that rely on the results of POS tagging, base phrase chunking, and morphological analysis. In several Arabic pipelines in the literature [7], however, upstream tasks such as POS tagging are implemented in a coarse-grained manner, which amounts to delegating the resolution of certain cases of ambiguity to downstream components. For example, by using single VERB and PART tags, the POS tagger in [6] avoids challenging ambiguities in Arabic verbs and particles, respectively. Consequently, an additional downstream component is needed for

morphological disambiguation, e.g., to find out whether تعرف is an imperative (تَعَرَّفْ/*recognize*), past (تَعَرَّفَ/*recognized*), or present tense verb (تَعْرِفُ/*you know or she knows*); whether the noun سحب is singular (in which case it means *withdrawal*) or plural (meaning *clouds*); or whether أن is an accusative (أَنَّ) or a subordinate particle (أَنْ).

Good-quality Arabic annotated corpora for machine learning are few and far between. The *Penn Arabic Treebank* [26] is a non-free, half-a-million-word annotated corpus destined for POS tagging and syntactic parsing, upon which a large number of research results are based. The KALIMAT corpus,[1] while freely available, is a silver standard corpus on which a POS tagging accuracy of 96% was reported [27].

In this paper, we present an entirely new linguistic pipeline for natural language processing of text in Modern Standard Arabic. Our goal was to provide an open-source tool that simultaneously maximizes accuracy, speed of execution, as well as the resolution of difficult cases of ambiguity within the Arabic text. This way, NLP tasks downstream of ALP are also expected to work in a more accurate and robust manner as they need to deal with less amount of ambiguity.

One of the general design principles we used to achieve these goals was to reduce the number of individual NLP components within the pipeline. Thus, ALP consists of just two components: the first one is a *preprocessor* that performs word segmentation, POS tagging, and named entity recognition as a single processing task, without any other auxiliary processing tool [28]. The second component uses these results to perform lemmatization [29].

In an effort to improve accuracy with respect to state-of-the-art tools, we decided to implement a solution that is independent from implicit NLP design choices embedded in the annotations of existing corpora. Thus, we hand-annotated an over two-million-token corpus that forms the basis of ALP. The pipeline can be tested online[2] and is freely available for research upon request.

The rest of the paper is organized as follows: Sect. 2 presents the main cases of lexical and morphological ambiguity in Arabic that ALP was designed to tackle. Section 3 introduces the general architecture of ALP and its components. Section 4 provides details and the rationale behind the tag sets we used for annotation. Section 5 presents the annotation methods we used for the two-million-token corpus. Section 6 provides evaluation results on ALP, and Sect. 7 reflects on future work.

[1] https://sourceforge.net/projects/kalimat/.

[2] http://www.arabicnlp.pro/alp/.

2 Ambiguity in Arabic

To illustrate the challenging cases that low-level NLP tasks such as word segmentation or lemmatization typically need to solve, in the following, we list some common examples of lexical and morphological ambiguity in Arabic.

2.1 Ambiguity in Word Segmentation

Certain words can be segmented into morphemes in more than one valid way. In such cases, the correct segmentation can only be determined in context. In Table 1, we list some common examples of ambiguity that occur at the segmentation level.

2.2 Ambiguity in POS Tagging

While correct segmentation decreases the ambiguity in Arabic text, polysemy and the lack of short vowels result in morphemes having multiple meanings with distinct parts of speech. In Table 2, we show some examples of this kind.

Table 1 Ambiguity examples at the segmentation level

Ambiguity	Example
Nouns vs conjunction+pronoun	وهن/*weakness* vs وهن/*and they (feminine)*
Noun vs conjunction+verb	وحل/*mud* vs وحل/*and (he) solved*
Noun vs conjunction+noun	وصفة/*receipt* vs وصفة/*and (a) character*
Noun vs singular noun+pronoun	كتابي/*two books (in genitive)* vs كتابي/*my book*
Noun vs preposition+noun	لسعة/*sting* vs لسعة/*for capacity*
Proper noun vs preposition+noun	بعقوبة/*a city in Iraq* vs بعقوبة/*with punishment*
Proper noun vs conjunction+noun	وهران/*a city in Algeria* vs وهران/*and two cats*
Proper noun vs definite article+noun	الباب/*a city in Syria* vs الباب/*the door*
Noun vs interrogative particle+negation particle	ألم/*pain* vs ألم/*did I not*
Adjective vs noun+pronoun	جانبي/*lateral* vs جانبي/*my side*
Adjective vs preposition+noun	بحرية/*nautical* vs بحرية/*to freedom*
Verb vs conjunction+pronoun	فهم/*(he) understood* vs فهم/*and they (masculine)*
Verb vs conjunction+verb	وفر/*saved* vs وفر/*and (he) escaped*
Verb vs verb+pronoun	علمنا/*we knew* vs علمنا/*(he) taught us*
Verb vs interrogative particle+verb	أتذكر/*(I) remember* vs أتذكر/*do (you) remember*

Table 2 Ambiguity examples at the POS tagging level

Ambiguity	Example
Verb vs noun	حمل/*carried* vs حمل/*carrying*
Verb vs comparative	أثقل/*overburdened* vs أثقل/*heavier*
Verb vs adjective	سهل/*facilitate* vs سهل/*easy*
Verb/noun vs particle	لم/*gathered* vs لم/*not*
Verb vs number	تسع/*expanded* vs تسع/*nine*
Verb vs proper noun	طلعت/*rose* vs طلعت/*Talat*
Noun vs number	سبع/*lion* vs سبع/*seven*
Noun vs proper noun	إحسان/*philanthropy* vs إحسان/*Ehsan*
Adjective vs noun	نوعية/*qualitative* vs نوعية/*quality*
Adjective vs proper noun	جميل/*nice* vs جميل/*Jamil*
Interrogative particle vs relative pronoun	According to their position in the sentence
Particle ambiguity in أن	أنْ/*subordinating* vs أنَّ/*accusative*
Particle ambiguity in إن	إنْ/*conditional* vs إنَّ/*accusative*
Particle ambiguity in لم	لمْ/*negation* vs لمَ/*interrogative*
Particle ambiguity in ما	ما/*negation* vs ما/*interrogative*

Even with correct segmentation and POS tagging, challenging cases of ambiguity still remain on the level of fine-grained POS tags, mostly due to MSA words overwhelmingly being written without diacritics. In the following, we list some examples of ambiguity with which we deal on the fine-grained level.

2.2.1 Verb Ambiguities: Passive vs Active Voice

Many verbs in Arabic have the same form in the active or passive voice cases. Verbs like سجّل/*reported or has been reported* can be only through the context disambiguated.

2.2.2 Verb Ambiguities: Past vs Present Tense

The same verb word form that denotes a verb in first-person singular present denotes (another) verb in third-person singular masculine past. Consider, for example, the verb أجمل which can be أُجْمِلُ/*(I) illustrate* can also be أَجْمَلَ/*(he) illustrated.*

A third-person singular feminine present verb form denotes (another) verb in third-person singular masculine past. Consider, for example, the verb تحمل which can be تَحْمِل/*(she) carries* can be also تَحَمَّلَ/*(he) sustained*.

2.2.3 Verb Ambiguities: Imperative

The imperative verb form (second person singular masculine) can be read as a past tense verb (third person singular masculine). For example, the verb تعرف which may be an imperative verb (تَعَرَّفْ/*recognize*) or a past tense verb (تَعَرَّفَ/*(he) recognized*).

The imperative verb form (second person plural masculine) can be read as a past tense verb (third person plural masculine). It can be also a present tense verb (third person plural masculine). For example, the verb تعرفوا which may be an imperative verb (تَعَرَّفُوا/*recognize*), a past tense verb (تَعَرَّفُوا/*(they) recognized*), or a present tense verb in cases like كَيْ تَعْرِفُوا/*so that (you) know*.

The imperative verb form (second person singular feminine) can be read as a present tense verb (second person singular feminine), after some particles. For example, the same form تعرفي can be an imperative verb (second person singular feminine like in (تَعَرَّفِي/*recognize*) or a present tense verb (second person singular feminine) after subordination particles such as in the case (كَيْ تَعْرِفِي/*so that you know*).

2.2.4 Noun Ambiguities: Singular vs Plural

In Arabic, there are several word forms that denote (different) singular and plural nouns. For example, the word سحب denotes the singular noun سَحْب/*dragging* and the plural noun سُحُب/*clouds*.

2.2.5 Noun Ambiguities: Dual vs Singular

The ا accusative case ending in Arabic leads to dual singular ambiguity. For example, the word form كتابا may be read as singular noun كِتَابًا/*one book* or dual كِتَابَا/*two books* (in genitive dual cases such as كِتَابَا العُلُوم).

2.2.6 Noun Ambiguities: Dual vs Plural

Dual form nouns and masculine plural noun in general are ambiguous. For example, the word مؤمنين can be read as مؤمِنَيْن/*dual form* or as مؤمِنِيْن/masculine plural form.

2.2.7 Noun Ambiguities: Feminine vs Masculine Singular

There are cases in which the same word form denotes singular but with different gender. For example, the word قدم can be feminine قَدَم/*foot* or masculine قِدَم/*antiquity*.

2.3 Ambiguity in Named Entity Recognition

Besides the ambiguity cases that we have presented in the previous section, we present below two examples of ambiguity related to NER, referring the reader to [30] for a more detailed treatise on the matter.

2.3.1 Inherent Ambiguity in Named Entities

It is possible for a word or a sequence of words to denote named entities that belong to different classes. For example, واشنطن denotes both a person and location. It is also frequent that organizations and establishments are named after person names. For example, جامعة الملك عبد الله للعلوم التقنية/King Abdullah University of Science and Technology.

2.3.2 Ellipses

Ellipses (omitting parts of nominal phrases and entity names) contribute to the high ambiguity of natural languages. Considering the lack of orthographic features in Arabic, ellipses increase the ambiguity. For example, a text about البحر الأبيض المتوسط/*The Mediterranean Sea* mentions it explicitly at the beginning of the text. After that, it may omit البحر الأبيض/the White Sea and refers to it by المتوسط/*the Mediterranean*. This word is used mostly as an adjective (which means *the average*), and there is no orthographic triggers that may disambiguate the entity from the adjective token.

2.4 Ambiguity in Lemmatization

Lexical ambiguity is pervasive in conventional written Arabic due to the absence of short vowels. For example, the past tense verb صرت could be vocalized as صِرْتُ with the corresponding lemma صَارَ/*become* or as صَرَّ with the corresponding lemma صَرَّ/*grit*. Nouns can also be ambiguous: the word سبل can be read as سُبُ/*ways* or سَبَل/ears.

As choosing the correct lemma is ultimately a word sense disambiguation problem, such cases put considerable stress on the quality of lemmatization. Tools that are capable of outputting multiple solutions in an order of preference are in this sense more robust as they potentially allow the disambiguation problem to be delayed to subsequent syntactic or semantic processing steps.

2.5 Ambiguity in Phrase Chunking

Ambiguity in phrase chunking is related to ambiguity at POS tagging and named entity recognition ambiguities. For example, the two tokens محمود الذهب could be read in two different ways. The first can be read as one nominal phrase محمود/*Mahmud* الذهب/*Althahab* (family name) in a sentence like رأيت محمود الذهب/*I saw Mahmud Althahab*. They can also be read as two separate nominal phrases محمود/*Mahmud* الذهب/*the gold* as in the following sentence: باع محمود الذهب/*Mahmud sold the gold*.

We have also the named entity boundary ambiguity problem. For example, the two nouns محمد/*Mohammed* and محمود/*Mahmud* can constitute two different noun sequences. While the sentence رأى محمد محمود/*Mohammed saw Mahmud* contains two nominal phrases, the sentence غادر محمد محمود//*Mohammed Mahmud left* contains only one. We believe that solving this kind of ambiguity needs extending the verb tags with the verb *transitivity*/*intransitive* information which is planned as a future work.

3 Pipeline Architecture

The pipeline specific to our method is shown in Fig. 1 and is composed of the following main steps:

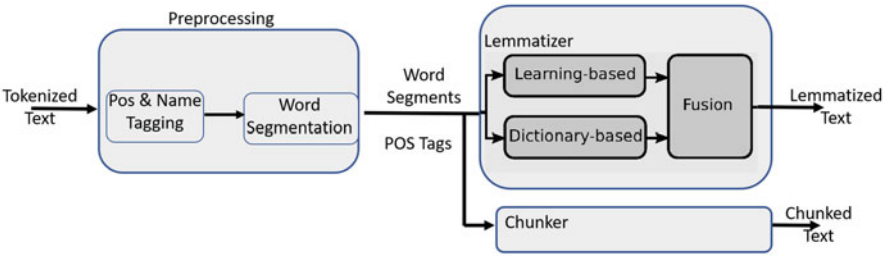

Fig. 1 The high-level NLP pipeline architecture for lemmatization and chunking

1. *Prepossessing:* taking white-space tokenized Arabic text in input, we pre-annotate the text through the following operations:
   (a) *POS and name tagging:* tokens are annotated by a machine-learning-based sequence labeler that outputs POS, named entity, and word segment tags.
   (b) *Word segmentation:* using the POS output, cliticized words are segmented into a proclitic, a base word, and an enclitic, making the subsequent lemmatization step simpler.
2. *Lemmatization:* the segmented and pre-annotated text is fed into the following lemmatizer components:
   (a) *Dictionary-based lemmatizer:* words are lemmatized through dictionary lookup.
   (b) *Machine-learning-based lemmatizer:* words are lemmatized by a trained machine learning lemmatizer.
   (c) *Fusion:* the outputs of the two lemmatizers are combined into a single output.
3. *Chunking:* the input of the chunker is similar to the input of the lemmatizer. The output is a list of base chunks.

The input of the annotator is expected to be UTF-8-encoded, white-space tokenized but otherwise unannotated text in Modern Standard Arabic. We are also supposing that sentences have been previously split by the usual sentence end markers (“.”, “!”, “?”, “. . . ”) and newlines.

3.1 Preprocessing: POS, NER, and Word Segment Tagging

The first component of ALP is a single *preprocessor* component that tackles three conventionally distinct NLP tasks: part-of-speech tagging, named entity tagging, and word segmentation. The common underlying goal of these preprocessing tasks is to reduce the ambiguity of words by extracting information from their morphology and context. Consequently, our combined tagger uses a machine-learning-based sequence labeling approach.

Performing the three operations in a single step presents several advantages:

- It is faster to execute than running several machine learning models in series.
- It is easier to reuse as part of a natural language understanding application.
- It does not suffer from the problem of cumulative errors that are inherent to solutions that solve the same tasks in series.

3.1.1 POS Tagging

The training corpus of the ALP preprocessor was annotated with fine-grained POS tags that, besides the high-level category, also provide number, gender, tense, and other information (see Sect. 4.1 for details). This detailed output can be effectively used by downstream components—such as the ALP lemmatizer—for solving a large number of cases of lexical ambiguity due to missing vocalization. There remain, however, some cases of word sense disambiguation that POS tagging alone cannot deal with, such as transitive/ditransitive verb ambiguity. For example, verbs such as علم (عَلِمَ/*knew* or عَلَّمَ/*taught*) remain ambiguous according to our current annotation tag set.

3.1.2 Named Entity Recognition

The ALP preprocessor does not mark named entities as nouns or proper nouns; rather, it annotates them directly with named entity tags (see Sect. 4.3 for details). This way the need for a separate NER component is avoided.

Based on the NER tags output by the ALP annotator, identifying the start and end of a named entity is a trivial task. Then through subsequent word segmentation, clitics can be removed from the entity and the canonical name obtained.

3.1.3 Word Segmentation

Word segmentation serves a double goal: to reduce the amount of distinct word forms, resulting in smaller and more robust lemmatizers, and to reduce lexical ambiguity due to multiple possible interpretations. For example, segmentation reduces the number of possible word forms of the lemma قلم from several hundreds of cliticized nouns { قلم, قلمي,بقلمي,وبقلمي, القلم,...} to six forms { قلم,قلمان,قلما,قلمين,قلمي,أقلام} only. On the other hand, word segmentation reduces the lexical ambiguity in cases such as لسعة which may be single word (*sting*) or a cliticized word (*for capacity*).

The actual segmentation of words is executed based on the clitic tags provided by the POS tagger. The input of the method is a word and its corresponding tag. The

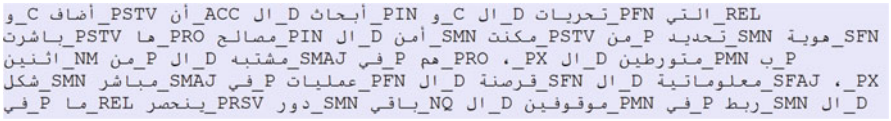

Fig. 2 An example output of the word segmenter

output is a list of tokens that correspond to the PROCLITIC, the BASETAG, and the ENCLITIC tags. Given that clitics are linguistically determined, segmentation becomes a simple string splitting task. An example of the output of a segmentation tool we implemented is shown in Fig. 2.

3.2 Lemmatization

Lemmatization returns the canonical (dictionary) forms of inflected words of a text. As such, it is a frequent upstream processing step before any analysis of lexical semantics (the meaning of words). For morphologically rich languages, lemmatization is usually a complex task, e.g., due to the presence of irregular cases. In Arabic, this includes broken plurals and irregular verbs. State-of-the-art lemmatizers typically apply finite-state transducers and/or lemmatization dictionaries to such cases, such as *AraComLex* [21] or the *OpenNLP lemmatizer*.[3] For regular cases, such as singular nouns and adjectives, regular plural nouns, and regular verbs, lemmatization is reduced to a straightforward task of *normalization* that removes inflectional prefixes and suffixes. For example, the verb form يحملوا is normalized into the lemma حمل. In the case of singular and regular plural nouns, it is sufficient to remove the plural suffixes and case endings. For example, the lemma of the dual noun form ولدين is ولد.

The ALP lemmatizer operates over an input pre-annotated by previous preprocessing steps, taking the segmented and POS-annotated text in input. It is built from the following components:

1. *A machine-learning-based lemmatizer:* words are lemmatized by a supervised classifier.
2. *A dictionary-based lemmatizer:* words are lemmatized through dictionary lookup.
3. *A fusion lemmatizer:* the outputs of the two lemmatizer components above are combined into a single output.

Both the learning-based and the dictionary-based lemmatizers were implemented using the Apache OpenNLP toolkit.[4]

[3] https://opennlp.apache.org/docs/1.8.4/manual/opennlp.html.

[4] http://opennlp.apache.org/docs/1.9.0/manual/opennlp.html#tools.cli.lemmatizer.

3.2.1 Learning-Based Lemmatizer

The principal component of our lemmatization approach is a machine-learning-based classifier. It takes word segments and their corresponding POS tag in input, also taking context (words and tags) into account. The learning-based approach is justified by the inherent ambiguity of diacritic-free Arabic words whose meanings are typically deduced, by humans and machines alike, from context. While the preliminary POS tagging resolves a great deal of ambiguity, some cases still remain such as the verb form يكون which may represent the verb كان or the verb كون.

3.2.2 Dictionary-Based Lemmatizer

A downside of learning-based lemmatization is that more rare and exceptional cases, such as أسنة (*spears*), may not be covered by its training corpus, which leads to lemmatization mistakes. The addition of new cases requires the re-training of the classifier. Another inconvenience is that classifiers—such as OpenNLP that we used—typically commit on a single output result, which may or may not be correct. In case of such ambiguity, from the full set of possible lemmas, further NLP processing steps may be able to provide correct results based on, e.g., syntactic or semantic analysis. In order to support these cases, we complement the learning-based lemmatizer by a dictionary-based one. The dictionary lemmatizer can be run independently, but we also provide a simple fusion method that combines the results of the two lemmatizers as described below.

3.2.3 Fusion Lemmatizer

While the learning-based lemmatizer outputs for each word a single candidate lemma, from the dictionary, multiple solutions could be retrieved even for a single part of speech (e.g., the verb form يسلم can be a verb form of the verb سلم *gave up* or أسلم *converted to Islam*). The goal of the simple fusion component is to produce a final result from these solutions. The final output is a list of one or more lemmas in a decreasing order of confidence.

The idea underlying the fusion method is that we *usually* trust the dictionary to be capable of providing a correct solution space (a small set of possible lemmas), while we *usually* trust the classifier to return the most likely lemma from the previous set. However, in the case of out-of-corpus words, the classifier may return incorrect results extrapolated from similar examples, such as returning the lemma تهمن for the word form يتهمن. Thus, whenever a lemma is returned by the classifier that is not included in the dictionary, it will still be included as a solution but with a lower confidence.

Accordingly, our simple fusion method is as follows. We take as input the results output by the two lemmatizers, namely, $L_{\mathrm{DIC}} = \{l_1, \ldots, l_n\}$ for the dictionary-based one and $L_{\mathrm{CL}} = \{l\}$ for the classifier-based one, and output L_{F}, the fusion result. We start by comparing the results of the two lemmatizers:

1. If $|L_{\mathrm{DIC}}| = 1$ and $l_1 = l$, i.e., the outputs are identical, then the solution is trivial, and we return either output and we are done: $L_{\mathrm{F}} = \{l\}$.
2. Otherwise, two further cases are distinguished:
   (a) If $l \in L_{\mathrm{DIC}}$, that is, the dictionary contains the classification output, then we prioritize the result of the classifier by making it first (i.e., the preferred lemma): $L_{\mathrm{F}} = \{l, l_1, \ldots, l_n\}$.
   (b) Otherwise, we add the classifier result as the last element: $L_{\mathrm{F}} = \{l_1, \ldots, l_n, l\}$.

3.3 Base Chunker

Base chunker splits sentences into groups of base chunks. These chunks in turn form sentence phrases or may be parts of sentence phrases. It operates over an input pre-annotated by previous preprocessing steps, taking the segmented and POS-annotated text in input. The output of the component is a list of chunks. These chunks may be one of the following base phrases:

1. Nominal phrases: Base nominal phrases are the longest possible sequence of adjacent words that constitute a phrase which does not contain coordinations or relative clauses. The length of base nominal phrases may range from one token to ten tokens or more. An example for such long phrases is عضو الرابطة الألمانية العامة لاضطرابات النوم المزمنة هارتموت رينتمايستر/ *member of the German General Association of Chronic Sleep Disorders Hartmut Rintmäster*.
2. Verbal phrases: Verbal phrases are phrases that contain a verb, two verbs, and an optional nominal phrase, prepositional phrase, or adverbial phrase complement. The verb part of the phrase may be also preceded by a particle such as لا in the following example: لا يزالون يعدون الترتيبات/*are still making arrangements*. Notice that the boundary of verbal phrases is implicit in case of the object complement. This means only the verbal part of the phrase will be annotated as a verbal phrase. The complement part will keep its original annotation.
3. Predicative adjective phrases: A predicate adjective is a predicate that follows a nominal phrase. It may range from one to several adjectives such as the phrases مناسب/*appropriate* and غير مناسب/*inappropriate*.
4. Prepositional phrases: Prepositional phrases are phrases that contain a preposition followed by a nominal phrase such as في المدسة/*at school*.

5. Adverbial phrases: Adverbial phrases are modifiers that follow an adjective like جدا in جميل جدا or a verb such as أسبوعيا in يسافر أسبوعيا/*travels weekly*. In the case of verbs, it is not necessary that the adverb follows the verb directly. It is possible to find intermediate phrases between the verb and its modifiers. Consider, for example, the phrase يسافر إلى فرنسا أسبوعيا/*travels to France weekly*.

4 Annotation Schema

This section presents the tag set used for the preprocessing component of the pipeline and the design choices behind it.

We annotated a single large training corpus with complex and fine-grained tags that encode information with respect to part of speech, word segments, and named entities. On a high level, the tag set is composed as follows:

```
<TAG>       ::= <PREFIX> <BASETAG> <POSTFIX>
<BASETAG>   ::= <POSTAG> | <NERTAG>
<PREFIX>    ::= <PREFIX> | <PROCLITIC> "+" | ""
<POSTFIX>   ::= <POSTFIX> | "+" <ENCLITIC> | ""
```

A tag is thus composed of a mandatory base tag and of zero or more (i.e., optional) proclitics and enclitics concatenated with the "+" sign indicating word segments. A base tag, in turn, is either a POS tag or a NER tag, but not both (in other words, we do not annotate named entities by part of speech). For example, the full tag of the word وبكتابه is a noun tag preceded by two proclitic tags (conjunction and preposition) and followed by a pronoun enclitic tag. The choice of our base and clitic tags was inspired by the coarse-grained tags used in MADA 2.32 [7] and 3.0 [19], as well as by the more fine-grained tags used in the Qur'an Corpus [31]. For compatibility with other NLP tools, mapping our tags to MADA 2.32, MADA 3.0, and Stanford [32] or any other tag sets is straightforward.

In Fig. 3, we provide an example of a complete annotated sentence.

```
وجد_PSTV مدير_SMN الشركة_D+SFN من_P خلال_LC الاطلاع_D+SMN على_P
السيرة_D+SFN الذاتية_D+SFAJ للشاب_P+D+SMN أنه_ACC+PRO متفوق_SMAJ
أكاديميا_AV بشكل_P+SMN كامل_SMAJ منذ_P أن_SUB كان_PSTV في_P
الثانوية_D+SFN العامة_D+SFAJ وحتى_C+P التخرج_D+SMN من_P
الجامعة_D+SFN ,_PX لم_NEG يخفق_PRSV أبدا_AV !_PX
```

Fig. 3 An example of annotated sentence

4.1 Annotation of POS Tags

The POS tag set consists of 58 tags classified into 5 main categories:

```
<POSTAG> ::= <NOUN> | <ADJECTIVE> | <VERB> | <ADVERB>
           | <PREPOSITION> | <PARTICLE>
```

Nouns The noun class has 13 tags as shown in Table 4. The first nine tags are fine-grained annotations of common nouns that we classify according to their number (singular, dual, or plural) and gender (masculine, feminine, or irregular). We use the feature *irregular* to annotate the irregular plural nouns. As it is the case in MADA, we consider quantifiers, numbers, and foreign words as special noun classes. Following the Qur'an corpus, we consider pronouns, demonstrative pronouns, and relative pronouns as special noun classes (Table 3).

Adjectives The adjective class has nine tags as described in Table 4. Similar to nouns, the first seven tags are fine-grained annotations of adjectives that we classify according to their number and gender. As it is the case in MADA, we consider comparative adjectives and numerical adjectives as special adjective classes.

Verbs The verb class contains five tags as described in Table 4. The first four tags are fine-grained annotations of verbs that we classify according to their passive marking (active or passive) and tense (past or present). Annotating future tense in Arabic is explained in the particle class. For imperative verbs, we use the tag *IMPV*.

Adverbs It is not clear in the modern Arabic linguistics community whether *adverb* belongs to the Arabic part of speech system or not. In this study, we follow FARASA and MADA in considering adverbs as a category of the Arabic part of speech system, where we consider adverbs as *predicate* modifiers that we classify in three classes as shown in Table 4.

Prepositions and Particles This class contains 28 preposition and particle tags from the Qur'an corpus tag set that we list in Table 5.

Table 3 Clitic tags

No. of clitics	No. of tags	Examples
0	78	*SMN* كتاب
1	163	*C+PRSV* ويفعع
2	105	*C+PIN+PRO* وأصدقاؤه
3	12	*C+FUT+PRSV+PRO* وسيكتبه

Table 4 Noun, adjective, verb, and adverb tags

Tag	Explanation	Example
SMN	Singular masculine noun	رجل , جبل ,كتاب
SFN	Singular feminine noun	بنت , قرية , جريدة
DMN	Dual masculine noun	كتابي , كتابا , كتابين , كتابان
DFN	Dual feminine noun	قريتي , قريتا , قريتين , قريتان
PMN	Plural masculine noun	موظفي ,موظفو , موظفين , موظفون
PFN	Plural feminine noun	كتابات ,حالات , إسهامات , موظفات
PIN	Plural irregular noun	رجال ,بنات , قرى , كتب
FWN	Foreign noun	سيناتور , لابتوب , موبايل , بنسلين
NQ	Quantifiers	أي , بعض , كل , جميع
NM	Numbers	اثنين , اثنان , ١ ,واحد
PRO	Pronouns	هما , هي , ـه ,هو
DM	Demonstrative pronouns	تلك , هؤلاء , هذان , هذا
REL	Relative pronouns	اللواتي , اللذان , التي ,الذي
SMAJ	Singular masculine adjective	سليم , قوي , جميل
SFAJ	Singular feminine adjective	سليمة , قوية , جميلة
DMAJ	Dual masculine adjective	جميلي , جميلا , جميلين , جميلان
DFAJ	Dual feminine adjective	سليمتان , قويتان , جميلتين
PMAJ	Plural masculine adjective	جميلي ,جميلو , جميليون , جميلين
PFAJ	Plural feminine adjective	سليمات , قويات , جميلات
PIAJ	Plural irregular adjective	حمر , أقوياء , أصحاء
AJCMP	Comparative adjectives	أسلم , أقوى , أجمل
AJNM	Ordinal adjectives	ثالث , ثاني ,أول
PRSV	Present verb (active)	يحمل , يسأل , يقول
PSTV	Past verb (active)	حمل , سأل ,قال
PPRSV	Present verb (passive)	يحمل , يسأل , يقال
PPSTV	Past verb (passive)	حمل , سئل , أقيل
IMPV	Imperative verb	احمل , اسأل , قل
T	Temporal adverb	بعد , أحيانا , صباحا
LC	Location adverb	بعد , تحت , فوق
AV	Adverb	تماما , خاصة , يوميا

Table 5 Preposition and particle tags

Tag	Explanation	Example
D	Definite article	ال
C	Conjunctions	ف , أو ,و
P	Prepositions	ل , الى ,من
Q	Interrogative particles	كيف ,هل ,ماذا
COND	Conditional particles	إنْ ,إذا ,لو
NEG	Negation particles	لن ,لا ,لم
ACC	Accusative particles	لعل , لكن ,إن
SUB	Subordinate particles	كي ,أنْ
FUT	Future particles	سوف ,سـ
VOC	Vocative particles	يا ,ـم
ANS	Answer particles	كلا لا ,نعم
EXL	Explanation particles	أما ,أي
EXP	Exceptive particles	عدا ,سوى
EXC	Exclamation particles	يا ,ما
RES	Restriction particle	إلا
CERT	Certainty particle	قد
SUR	Surprise particle	إذن ,إذا
EMPH	Emphatic particle	لـ
PRP	Purpose particle	لـ
RET	Retraction particle	بل
REM	Resumption particles	و ,ف
INTG	Interrogative particle	أ
PREV	Preventive particle	ما
INC	Inceptive particle	ألا
IMPP	Imperative particle	ل
PR	Prohibition particle	لا
ABB	Abbreviation	د دكتور
PX	Punctuation	., :

4.2 Annotation of Word Segments

We represent the morphology of words through complex tags that correspond to their internal structure. As shown above, the structure of a complex tag is

[PROCLITIC+]* BASETAG [+ENCLITIC]*

where BASETAG is one of the base POS tags; ENCLITIC, when present, stands for one or two clitic tags at the end of the word; and PROCLITIC, when present, is the combination of one to three tags out of a set of the proclitic tags at the beginning of the word.

In our corpus, the number of distinct individual tags (including both simple and complex tags) is 358, as shown in Table 3.

4.3 Annotation of Named Entities

The named entity tags output by the ALP preprocessor are shown below:

```
<NERTAG>    ::= <POSITION> "-" <CLASS>
<POSITION>  ::= "B" | "I"
<CLASS>     ::= "PER" | "LOC" | "ORG" | "EVENT"| "MISC"
```

Following the conventions of CONLL-2003,[5] the NER tags provide both the class of the entity and its boundaries through indicating the positions of the tokens composing it. `B-` stands for *beginning*, i.e., the first token of the entity, while `I-` stands for *internal*, marking subsequent tokens of the same entity.

Our corpus currently distinguishes the most common types of named entities: persons, locations, organizations, events, and others. We did not yet classify entity classes such as date, time, currency, or measurement nor subclasses of organizations (e.g., we do not differentiate between a football team and a university).

Thus, the total number of NER tags is 10 as shown in Table 6; however, as shown earlier, NER tags can be further combined with clitic tags.

Table 6 Named entity tags

Tag	Explanation	Example
B-PER, *I-PER*	Person	نجيب_B-PER محفوظ_I-PER
B-LOC, *I-LOC*	Location	البحر_B-LOC الأبيض_I-LOC المتوسط_I-LOC
B-ORG, *I-ORG*	Organization	حزب_B-ORG الحرية_I-ORG والعدالة_I-ORG
B-EVENT, *I-EVENT*	Event	الحرب_B-EVENT العالمية_I-EVENT الثانية_I-EVENT
B-MISC, *I-MISC*	Misc	درب_B-MISC التبانة_I-MISC

[5] http://www.cnts.ua.ac.be/conll2003/ner/annotation.txt.

4.4 Annotation of Lemmas

The lemmatization dictionary is a text file with each tab-separated row containing a word form, its POS tag, and the corresponding lemma and each column separated by a tab character. In case of ambiguous word forms (i.e., a word form-POS tag pair that has several lemmas), the corresponding lemmas are separated by "#" character: for example, the lemmas of the word form تزور are زار#زور. An example of the dictionary is shown in Fig. 4a.

The format of the annotated corpus for the learning-based lemmatizer is similar to the dictionary. The only difference is that the entries are ordered according to their original position in the sentence in the segmented corpus, in order to retain context. An empty line indicates the end of a sentence. An example of the training corpus is shown in Fig. 4b.

4.5 Annotation of Base Chunks

For example, the phrase الرجل الطويل/*the tall man* in Fig. 5 is a base chunk, while the phrase الرجل الطويل فوق المبنى/the tall man on the building is not.

We use the following BIO annotation schema to annotate the base chunks:

يوم	SMN	يوم
يوما	SMN	يوم
يومان	DMN	يوم
يوما	DMN	يوم
يومين	DMN	يوم
يومي	DMN	يوم
أيام	PIN	يوم
أياما	PIN	يوم

(a)

مواطنين	PMN	مواطن
و	C	و
إقلاق	SMN	إقلاق
ال	D	ال
راحة	SFN	راحة
ال	D	ال
عامة	SFAJ	عام
في	P	في

(b)

Fig. 4 Examples of the contents of (**a**) the lemmatization dictionary and (**b**) the training corpus of the classifier-based lemmatizer

The ancient palace	at	looks	The building	on	The tall man
القصر القديم	إلى	ينظر	المبنى	فوق	الرجل الطويل
I-NP B-NP	B-PP	B-VP	B-NP	B-PP	I-NP B-NP

Fig. 5 An example output of the base chunker

```
<PHRASE>    ::= <B> "-" <CLASS> (<I> "-" <CLASS)*
<B>         ::= "B"
<I>         ::= "I"
<CLASS>     ::= "NP" | "VP" | "PP" | "ADJP"| "ADVP"
```

In the following, we describe the base chunks and their annotations:

- Nominal phrases: The annotation schema for basic nominal phrases is B-NP (I-NP)*. This can be one of the following basic noun phrase categories:
  1. Pronouns: This category refers to separate pronouns such as هو/*he* or attached object pronouns like ه/*him*. Attached possessive pronouns like ه/*his* are genitive constructions as decribed below. Pronouns are annotated using the tag B-NP.
  2. Nouns: These refer to single-word nouns. They can be definite such كتاب/*book* and indefinite like الكتاب/*the book*. We use the tag B-NP if the noun is indefinite and the tag B-NP I-NP otherwise.
  3. Nouns+possessive pronouns: This category refers to nouns followed by attached possessive pronouns such as كتابي/*my book*. Such phrase is annotated as B-NP I-NP.
  4. Nouns+adjective modifiers: This category refers to nouns modified by one or more adjectives. They can be definite nouns such as الكتاب الجديد/*the new book* and كتابي الجديد/*my new book* or indefinite nouns like كتاب جديد/*a new book*. The possible tag sequence here is (B-NP|(B-NP I-NP)) $(\text{I-NP})^+$.
  5. Genitive nouns: Noun+possessive pronouns are special case of this category. It includes also the other forms of genitive constructions. They can be definite nouns such كتاب المدرسة/*The school book* or indefinite nouns like كتاب مدرسة/*school book*. The length of the noun sequence can be of course more than two tokens such as كتاب معلم المدرسة/*the school teacher book*. We use the tag sequence B-NP $(\text{I-NP})^+$ here.
  6. Genitive nouns+adjective modifiers: This category contains possessive constructions modified by one or more adjectives. The phrases may be definite such as كتاب المدرسة الجديد/*the new school book* or indefinite like كتاب مدرسة جديد/*a new school book*. The used tag sequence here is B-NP (I-NP) $(\text{I-NP})^+$.
  7. Proper nouns: This category contains proper nouns such as محمد/*Mohammed* or محمد بن عبد الله/*Mohammed bin Abdullah*. The length of the sequence can be in some cases very long.
  8. Nouns+proper nouns+adjective modifiers B-NP (I-NP) $(\text{I-NP})^+$: This category contains phrases such as خطاب ترامب الهجومي/*Trump's offensive speech*.

- Prepositional phrases: Prepositional phrases begin with a preposition followed by one of the basic nominal phrases. The only used tag here is B-PP.
- Verbal phrases: Verbal phrases consist of two parts—a verbal part that may be followed by one of the basic nominal phrases. The sequence in the verbal part may contain one token which the main verb such as تعلم/*learned*, an auxiliary verb followed by the phrase main verb such as كان يتعلم/*was learning*, negation particle followed by the main verb such as لم يتعلم/*didn't learn*, or a negation particle followed by an auxiliary verb and the main verb such as لم يكن يتعلم/*was not learning*. The possible tag sequences are B-VP (NULL—I-VP—(I-VP I-VP)).
- Predicative adjective phrases: Predicative constructions may be a predicative adjective such as جميل/*beautiful* or a negation particle followed by a predicative adjective like غير جميل/*not beautiful*. The possible tag sequences are B-ADJ (NULL|I-ADJ | (I-ADJ I-ADJ)).
- Adverbial phrases: Adverbial constructions contain an adverb that modifies a verbal or adjectival phrase. Examples for this category are جدا/*very*, أسبوعيا/*weekly*, or شخصيا/*personally*. The used tag for adverbial phrase is B-ADV.

5 Corpus Annotation

This section presents the methods we used to annotate the more than two-million-token corpus that is the fundament of ALP. We have chosen to develop an entirely new corpus instead of relying on existing resources such as the Penn Arabic Treebank [26] in order to be free both from licensing restrictions and from past modeling choices. The main challenge of the annotation process, besides the corpus size, was to cover and address as many cases of ambiguity (discussed in Sect. 2) as possible.

The corpus was assembled from documents and text segments from a varied set of online resources in Modern Standard Arabic, such as medical consultancy web pages, Wikipedia, news portals, online novels, and social media, as shown in Table 7. The diversity of sources serves the purpose of increasing the robustness of the model with respect to changes in domain, genre, style, and orthography. For consistency within the corpus and with the type of texts targeted by our annotator, we removed all short vowels from the input corpora.

The current corpus consists of more than 130k annotated sentences with more than 2 millions Arabic words and 200k punctuation marks (Table 8).

Table 7 Resources used in the training corpus

Resource	Proportion
Aljazeera online	30%
Arabic Wikipedia	20%
Novels	15%
Al-quds Al-Arabi newspaper	10%
Altibbi medical corpus	10%
IslamWeb	5%
Social networks (Facebook, Twitter)	5%
Other resources	5%

Table 8 Domains covered in the training corpus

No. of documents	No. of tokens	Domain	Description
Archaeology	61	19745	Archaeological and fossil science
Articles	285	261264	Articles from news portals
Business	73	49937	(Online) Business and trading
Economy	159	61,954	Regional and international economy
Encyclopedia locations	140	129553	Cities, villages, and countries
Encyclopedia persons	131	90601	Famous persons
Encyclopedia politics	53	52080	Political organizations and events
Environment	110	41506	Environment-related documents
Food recipes	11	4511	Food and cooking recipes
Literature	130	264354	Novels, short stories, proverbs
Medical	339	159,452	Human health-related documents
Medical answers	1	297608	Medical question answers from medical portals
Miscellaneous	24	2190	Uncategorized documents
Miscellaneous news	985	276323	None political or military nature news
Nature	62	25044	Nature-related documents
News	1013	373884	Political and war news
Quran	1	2915	Verses from Quran
Science	196	73650	Science-related documents
Space	84	41629	Space-related documents
Sport	30	9398	Sport-related documents
Technology	77	24588	Technology-related documents
Theology	10	19576	Islamic theology articles

5.1 POS and Name Annotation Method

The annotation was performed semi-automatically in two major steps:

1. Annotation of a corpus large enough to train an initial machine learning model.

2. Iterative extension of the corpus. We add new sets of sentences tagged by the current model after manual correction. Then, we retrain the model after each iteration.

Step 1 was an iterative process. It was bootstrapped using a 200-sentence gold standard *seed corpus* that was fully hand-annotated. The goal of each iteration was to add a new set of 100 new sentences to the seed corpus, until about 15k sentences were tagged. Each iteration consisted of the following steps:

1.a For each word in the untagged corpus that was already tagged in the seed corpus, simply copy the tag from the tagged word (this of course can lead to wrong tagging as the process does not take the context into account; we fix such mistakes later).
1.b Find the 100 sentences with the highest number of tags obtained through replacement in the previous step.
1.c Manually verify, correct, and complete the annotation of the sentences extracted in step 1.b.
1.d Add the annotated and verified sentences to the seed corpus and repeat.

At the end of step 1, many rounds of manual verification were performed on the annotated corpus.

In step 2, we extended the corpus in an iterative manner:

2.a Train an initial machine learning model from the annotated corpus.
2.b Use this model to label a new set of 100 sentences.
2.c Verify and correct the new annotations obtained.
2.d Add the newly annotated sentences to the annotated corpus and repeat.

For training the machine learning model, we used the POS tagger component of the widely known OpenNLP tool with default features and parameters. The annotation work was accomplished by two linguists, the annotator and a consultant who was beside the design of the tag set active in corrections and consultations especially in the first phase.

5.2 Lemma Annotation Method

In the following, we describe the method we used to build the lemmatization dictionary and the annotations for the learning-based lemmatizer. Our starting point was the POS corpus that we extended with lemma annotations.

5.2.1 Dictionary Lemmatizer

The dictionary was generated through the following steps:

Table 9 Plural classes

Class	Possible word forms	Example
SMN_PMN	SMN,SFN,DMN, DFN, PFN, PMN	مؤمن:مؤمن,مؤمنة,مؤمنان,مؤمنتان,مؤمنات,مؤمنون
SMN_PFN	SMN,DMN,PFN	إجراء:إجراء,إجراءان,إجراءات
SFN_PFN	SFN,DFN,PFN	كتابة:كتابة,كتابتين,كتابات
SMN_PIN	SMN,DMN,PIN	علم:علم,علمان,أعلام
SFN_PIN	SFN,DFN,PIN	أسرة:أسرة,أسرتان,أسر
FWN_PFN	FWN,DMN,PFN	تلفزيون:تلفزيون,تلفزونان,تلفزيونات

1. **Segmentation.** The corpus was segmented as explained in the previous section. The result of this step was generating a segmented corpus that contains more than 3.1 million segmented tokens.
2. **POS tag-based classification.** In this step, we classified the word forms according to their POS tag.
3. **Inherent feminine and adjectival feminine classification.** In this step, we classified the feminine nouns into inherent feminine and adjectival feminine nouns. For example, the noun أسرة_SFN/*family* is inherent feminine, while the noun أسيرة_SFN/*prisoner* is adjectival. This differentiation is important because the lemma of adjectival nouns is the masculine singular form of the noun أسير, while there is no masculine singular lemma for أسيرة.
4. **Plural type classification.** In this step, we classified the singular and dual nouns (after extracting their singular forms) according to their plural type into six classes as shown in Table 9. This classification enables us to build the possible *word number-gender* forms of a given lemma automatically. For example, the class SMN_PMN has six different possible *number-gender* forms. On the other hand, using the feminine classification lists in the previous step enabled us to differentiate between the SMN_PFN and SFN_PFN. In the class SMN_PFN, the lemma of a singular feminine noun (SFN) is the singular masculine noun (SMN). In the class SFN_PFN, on the other hand, the lemma is the singular feminine noun itself. The adjectives were classified into three classes. The first class is similar to the class SMN_PMN which allows six different word forms. The second class contains a seventh possible form which is the broken plural adjective form. The third class contains PIAJ as a single possible plural form. For example, ماهر belongs to this class since it has two possible plural forms ماهرون and مهرة.
5. **Lemma extraction.** This step is semi-automatic as follows:
   (a) **Manual.** Assigning the lemmas to broken plural nouns and adjectives was performed manually.

(b) **Automatic.** Based on the morphological features in the tags, it was possible to extract lemmas for singular, dual, masculine plurals, feminine plurals adjectives, and nouns. We also used rules to extract the verb lemmas such as removing the affixes وا.

6. **Lemma enrichment.** Using the lemmas from the previous step, we have enriched the corpus with new verbs, adjectives, and nouns. For example, if the lemma of a plural noun or adjective was missing, we added it to the noun and adjective lemma lists.
7. **Dictionary generation.** The files produced so far are as follows:

   - **Noun files.** Three files for masculine, feminine, and foreign nouns. The lemmas in these files were classified according to Table 9. There is a fourth file that contains quantifiers, pronouns, adverbs, etc.
   - **Adjectives.** Three files for adjectives, comparatives, and ordinal adjectives. The lemmas in the adjective file are classified according to Table 9.
   - **Verbs.** One file that contains all extracted verb lemmas.

   Using these files, the dictionary was generated as follows:

   (a) **Noun and adjective generation.** According to the plural class, the noun and adjective forms were generated. The ا case ending and changing ة to ت were also considered in this step.
   (b) **Verb generation.** For each verb in the verb lemma list, we automatically generated the verb conjugations in present, past, and imperative cases. We considered also accusative (فعل منصوب) and asserted verbs (فعل مجزوم).
   (c) **Dictionary building.** Using the results from the previous step, we built the dictionary as shown in Fig. 4b, where the lemmas of ambiguous surface forms were concatenated into a single string using the # separator.

5.2.2 Machine Learning Lemmatizer

We used the segmented corpus from the previous section to build the lemmatization corpus in the two steps below:

1. **Lemma assignation.** We used a dictionary lemmatizer to assign the word forms to their corresponding lemmas. In case of prepositions, particles, and numbers, the lemma of the word form was obtained through simple normalization. The lemmas of named entities were the named entities themselves. If a word form was ambiguous, all its possible lemmas were assigned.
2. **Validation.** We disambiguated the lemmas of the ambiguous word forms manually.

The size of the generated corpus is 3,229,403 lines. The unique word forms after discarding the digits are 59,049 as shown in Table 10.

Table 10 Distribution of lemmas and unique word forms by part of speech in the corpus of the learning-based lemmatizer

POS	No. of lemmas	No. of word forms
Noun	18,165	26,337
Adjective	6369	13,703
Verb	4258	19,009
Named entity	20,407	20,407
Particle	605	649

In a final step, we added all generated word forms and their corresponding lemmas from the dictionary described in the previous section to the corpus. This increased the size of the corpus to 3,890,737 lines.

5.3 Base Chunking Annotation Method

We used the segmented corpus from the previous section to build the chunking corpus by using the BIO tags semi-automatically. The manual part of this process was identifying the POS tag sequences that constitute a phrase chunk. The total number of the identified sequences was 4298. Most of these sequences were nominal phrase sequences, where the number of this group was 4250 sequence. In Table 11, we give examples for the identified noun sequences. The complete identified POS tag sequences can be found on the project page on ResearchGate.[6] In Table 12, we show the identified verb sequences. Table 13 contains predicative adjective sequences. The prepositional and the adverbial groups contained one sequence only which is the preposition or the adverb.

Using the identified sequences, we have built the corpus automatically. The size of the current chunking corpus is more than three million tokens. Of course, we do not claim that the identified sequences are exhaustive. There may be other non-detected sequences in our corpus or other sequences that our corpus does not cover.

6 Evaluation

In this section, we present evaluation results, both on individual NLP tasks and overall results pertaining to the ALP pipeline as a whole (the latter included within the lemmatizer results).

[6] https://www.researchgate.net/project/ALP-Arabic-Linguistic-Tool.

Table 11 Noun phrases Sequences

Length	No. of sequences	Sequences sample	Sample phrase
1	21	D+SMN	البيت
2	351	D+SFN D+SFAJ	الراحة الجسدية
3	1119	SMN PFN D+DMN	تعزيز علاقات البلدين
4	1534	AJCMP PFN SMN D+PFN	أهم مقومات نجاح الشركات
5	861	PFN SMN PFN D+PIN D+PIAJ	احتمالات ارتفاع مستويات الديون المتعثرة
6	263	SMN PIN SFN D+SMN D+SFAJ D+SFAJ	تنفيذ بنود خطة العمل الشاملة المشتركة
7	68	SFN SMN PFN SMN D+PFN D+PFAJ D+PFAJ	ورشة عمل مهارات إعداد الموازنات التشغيلية المالية
8	14	SMN NQ PIN SMN SFN SMN D+PIN D+PIAJ	اتباع بعض طرق تسهيل عملية بلع الأقراص الدوائية

Table 12 Verbal phrase sequences

Length	No. of sequences	Sequence sample	Sample phrase
1	5	PSTV	أرجع
2	14	PRSV PSTV	تكون أأارت
3	7	NEG PSTV PPRSV	ما زالت ترتكب

Table 13 Predicative adjective phrases Sequences

Length	No. of sequences	Sequence sample	Sample phrase
1	9	DFAJ	صامتين
2	9	NEG SFAJ	غير ضارة
3	7	SMAJ SMAJ SMAJ	بني رمادي مشقق

6.1 Evaluation of POS Tagging

To evaluate the performance of the proposed solution, we trained a machine learning model on the annotated corpus using the *OpenNLP maximum entropy POS tagger* with default features and *cutoff* = 3. We did not apply any preliminary normalization to the evaluation corpus. The evaluation corpus was taken from two sources: the *Aljazeera* news portal and the *Altibbi* medical consultancy web portal. The text contained 9990 tokens (9075 words and 915 punctuations). Manual validation of the evaluation results was performed. The per-task accuracy figures are shown in Table 14.

Table 14 Evaluation results on the POS tagging and word segmentation tasks

Error type	Number of errors	Accuracy
Segmentation	25	99.7%
Coarse-grained POS	131	98.7%
Fine-grained POS	206	97.9%

The *segmentation* error type refers to words that were not segmented correctly. The *coarse-grained POS* error type refers to words that were correctly segmented but the base POS tag was wrong. This also includes incorrect named entity POS tags. Finally, the *fine-grained POS* error type means that the word segmentation and the coarse-grained POS tag were correct but the fine-grained information within the tag was incorrect in one of the following ways:

- For nouns and adjectives: number/gender error
- For verbs: tense error or passive/active voice error

In some cases, the tag included more than one type of error. For example, the ومضرة_SFN tag (instead of C+SFAJ) includes both segmentation and POS tagging errors and therefore was counted twice.

The evaluation data and the process to replicate the evaluation tests are available online.[7]

6.2 Evaluation of NER

We evaluated the named entity recognition component separately. Our evaluation corpus contained 674 named entity tags that denote 297 named entities (e.g., روبرت_B-PER واتسون_I-PER is one named entity that contains two named entity tags). The total number of true positives (correctly detected and classified named entities) was 265 (89.2% precision). The number of false negatives (assigning a non-named entity tag, partial tagging, named entity boundary error, or a wrong named entity class applied) was 32 and the number of the false positives 15 (94.6% recall). F1-Measure = 91.8%. In Table 15, we provide some examples of these errors.

6.3 Evaluation of Lemmatization and Base Chunking

For evaluating the lemmatizers, we used a corpus of a 46,018-token text, retrieved and assembled from several news portals (such as Aljazeera news portal[8] and *Al-*

[7] http://www.arabicnlp.pro/alp/eval.zip.

[8] http://www.aljazeera.net/.

Table 15 Examples of NER mistakes

Error type	Example
Non-NER tag	وففوريدا_C+SMN instead of C+B-LOC
Partially tagged	وفيك_SMN سيجبيسما_I-PER instead of وفيك_C+B-PER سيجبيسما_I-PER
Boundary error	شركة_SFN سيبريان_B-ORG االروسية_I-ORG instead of شركة_SFN سيبريان_B-ORG االروسية_D+SFAJ
Wrong classification	غرين_B-LOC (in ودعا غرين) instead of غرين_B-PER
False positive	نظم_P+B-ORG االآيكوووجية_I-ORG instead of نظم_P+D+PIN االآيكوووجية_D+PIAJ

quds Al-Arabi newspaper[9]). We excluded from the evaluation the categories of tokens that cannot be lemmatized: 5853 punctuation tokens, 3829 tokens tagged as named entities, 482 digit tokens, and 10 malformed tokens (i.e., containing typos, such as أوروبيةبالموافق instead of أوروبية بالموافق). Thus, the number of tokens considered was 35,844. In order to have a clear idea of the efficiency of the lemmatization pipeline, we evaluated it in a fine-grained manner, manually classifying the mistakes according to the component involved. This allowed us to compute a comprehensive accuracy for the entire pipeline as well as evaluate individual components. The evaluation data files are available online.[10]

The fine-grained evaluation is summed up in Table 16.[11] *Nonexistent lemma* stands for cases where the POS tag and the segmentation were correct, yet the classifier gave a wrong, non-linguistic result. *Wrong disambiguation* means that the lemmatizer chose an existing but incorrect lemma for an ambiguous word form.

The accuracy measures reported in Table 17 were computed based on the results in Table 16. On these, we make the following observations. The performance of preprocessing (98.6%) represents an upper bound for the entire lemmatization pipeline. In this perspective, the near-perfect results of the classifier (99.5% when evaluated in isolation, 98.1% on the entire pipeline) are remarkable. We cross-checked these results using the built-in cross-validation feature of OpenNLP and obtained similar results (99.7%). The dictionary-based lemmatizer reached a somewhat lower yet still very decent result (96.6% in isolation, 95.2% on the entire pipeline), due to the 1207 OOV word forms. The fusion of the two lemmatizers, finally, improved slightly on the classifier: of the 170 mistakes made by the classifier, 120 could be correctly lemmatized using the dictionary. Thus, the fusion method

[9] http://www.alquds.co.uk/.

[10] http://www.arabicnlp.pro/alp/lemmatizationEval.zip.

[11] While, after tagging and segmentation, the number of (segmented) tokens rose to 62,694, we computed our evaluation results based on the number of unsegmented tokens.

Table 16 Types of mistakes committed by the learning-based lemmatizer and their proportions

Type of mistake	Occurrences	Example
POS tag (coarse-grained) mistakes	199	تبريج_SMN instead of تبريج_PRSV
Morphological tag (fine-grained) mistakes	201	كروم_SMN instead of كروم_PIN
Segmentation tag mistakes	103	أخذتنا_PSTV instead of أخذت_PSTV and نا_PRO
Classifier mistakes: nonexistent lemma	158	جدا instead of وجد for يجدوا_PRSV
Classifier mistakes: wrong disambiguation	12	كان instead of كون for يكونوا_PRSV
Dictionary mistakes: missing word form	1207	كاشطة,دواعم,قرفصاء
Fusion mistakes	50	نتن ,عواام ,دواعم

Table 17 Accuracy values computed for various components of the lemmatization pipeline

Component	Evaluation method	Accuracy
Preprocessing	All mistakes (POS, morphological, segmentation)	98.6%
Classifier-based lemmatizer	In isolation	99.5%
Classifier-based lemmatizer	In isolation, built-in OpenNLP cross-validation	99.7%
Classifier-based lemmatizer	Entire pipeline	98.1%
Dictionary-based lemmatizer	In isolation	96.6%
Dictionary-based lemmatizer	Entire pipeline	95.2%
Fusion lemmatizer	Entire pipeline	98.4%

reached a full-pipeline result of 98.4%, only a tiny bit worse than the performance of preprocessing itself. We have used cross-validation method to evaluate the chunking corpus. The evaluation result was 98.6%.

7 Conclusion and Future Work

This paper presented ALP, an Arabic linguistic pipeline that solves low-level Arabic NLP tasks: POS tagging, word segmentation, named entity recognition, and lemmatization. All of these tasks were performed using tools and resources derived from a new 2.2-million-word corpus hand-annotated by the authors. Due to the size of the corpus but also the annotation schemas and the overall pipeline design, ALP manages to disambiguate a large proportion of cases of lexical ambiguity and perform the tasks above with high accuracy. This increases the potential of downstream language understanding tasks, some of which we are planning to include in ALP in the future.

In particular, we are working on new Arabic components such as a vocalizer, a phrase chunker, a dependency parser, or a multiword expression detector.

The trained models and corresponding tools are free for research purposes upon request. We are also planning to release the annotated corpus itself in the near future.

We are also planning further improvements on the existing components, as detailed below:

Fine Tuning While the tool reached very good results with default OpenNLP features, we believe that they can still be improved by customizing the classifier and the features or using another machine or deep learning algorithm such as CRF and BiLSTM.

Noun Classification In the current tag set, we do not differentiate between gerunds (المصدر) and other noun classes. For example, the noun قلب/*heart* is tagged the same as the gerund قلب/*overthrow*.

Named Entity Classification The classification of named entities in our corpus is still incomplete and coarse-grained. For example, العباسيين, الشعب الألماني, and اللغة العربية are not classified as named entities. We plan to introduce new classes such as dates and currencies, as well as a finer-grained classification of organizations.

Chunker Coverage Extension We are planning to extend the chunker to detect sentence phrases without restriction. This will include detecting coordinated nominal phrases, verbal phrases, and relative clauses. We plan also to do more research on the relation between adverbs and other phrases and find a way to connect the modifying adverb to its modified phrase. On the other hand, we will also work on extending the verb POS tags to differentiate between transitive and intransitive verbs and study the effect of this new verb classification on proper name boundary disambiguation.

Other Tools and Corpora We plan to use the same corpus and tag set to produce annotations for other NLP tasks such as co-reference resolution and parsing.

References

1. Balakrishnan, V., Ethel, L.: Stemming and lemmatization: a comparison of retrieval performances. Lect. Notes Soft. Eng. **2**, 262–267 (2014)
2. Navigli, R.: Word sense disambiguation: a survey. ACM Comput. Surv. **41**, 10:1–10:69 (2009). http://doi.acm.org/10.1145/1459352.1459355
3. Bella, G., Zamboni, A., Giunchiglia, F.: Domain-based sense disambiguation in multilingual structured data. In: The Diversity Workshop at the European Conference on Artificial Intelligence (ECAI) (2016)
4. Freihat, A., Qwaider, M., Giunchiglia, F.: Using grice maxims in ranking community question answers. In: Proceedings of the Tenth International Conference on Information, Process, and Knowledge Management, EKNOW 2018, Rome, March 25–29, pp. 38–43 (2018)

5. Giunchiglia, F., Kharkevich, U., Zaihrayeu, I.: Concept search. In: The Semantic Web: Research and Applications, pp. 429–444 (2009)
6. Darwish, K., Mubarak, H., Abdelali, A., Eldesouki, M.: Arabic POS tagging: Don't abandon feature engineering just yet. In: Proceedings of the Third Arabic Natural Language Processing Workshop, pp. 130–137 (2017)
7. Diab, M.: Second generation AMIRA tools for Arabic processing: Fast and robust tokenization, POS tagging, and base phrase chunking. In: 2nd International Conference on Arabic Language Resources and Tools, vol. 110 pp. 285–288 (2009)
8. Khoja, S.: APT: Arabic part-of-speech tagger. In: Proceedings of the Student Workshop at NAACL, pp. 20–25 (2001)
9. Aldarmaki, H., Diab, M.: Robust part-of-speech tagging of Arabic text. In: Proceedings of the Second Workshop on Arabic Natural Language Processing, pp. 173–182 (2015)
10. Habash, N., Rambow, O.: Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, pp. 573–580 (2005)
11. Sawalha, M., Atwell, E.: Fine-grain morphological analyzer and part-of-speech tagger for Arabic text. In: Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC'10), pp. 1258–1265 (2010)
12. Mohamed, E., Kübler, S.: Is Arabic part of speech tagging feasible without word segmentation? In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 705–708 (2010)
13. Shaalan, K., Raza, H.: Arabic named entity recognition from diverse text types. In: Proceedings of the 6th International Conference on Advances in Natural Language Processing, pp. 440–451 (2008)
14. Althobaiti, M., Kruschwitz, U., Poesio, M.: A semi-supervised learning approach to Arabic named entity recognition. In: Recent Advances in Natural Language Processing, RANLP 2013, 9–11 September, Hissar, Bulgaria, pp. 32–40 (2013). http://aclweb.org/anthology/R/R13/R13-1005.pdf
15. Darwish, K.: Named entity recognition using cross-lingual resources: Arabic as an example. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol. 1 pp. 1558–1567 (2013)
16. Abdallah, S., Shaalan, K., Shoaib, M.: Integrating rule-based system with classification for Arabic named entity recognition. In: Computational Linguistics and Intelligent Text Processing - 13th International Conference, CICLing 2012, New Delhi, March 11–17, 2012, Proceedings, Part I, pp. 311–322 (2012)
17. AlGahtani, S.: Arabic Named Entity Recognition: A Corpus-Based Study, Ph.D. Thesis. University of Manchester (2011)
18. Boudchiche, M., Mazroui, A., Ould Abdallahi Ould Bebah, M., Lakhouaja, A., Boudlal, A.: AlKhalil Morpho Sys 2: A robust Arabic morpho-syntactic analyzer. J. King Saud Univ. Comput. Inf. Sci.. **29**, 141–146 (2017). https://doi.org/10.1016/j.jksuci.2016.05.002
19. Pasha, A., Al-Badrashiny, M., Diab, M., El Kholy, A., Eskander, R., Habash, N., Pooleery, M., Rambow, O., Roth, R.: MADAMIRA: a fast, comprehensive tool for morphological analysis and disambiguation of Arabic. LREC. **14**, 1094–1101 (2014)
20. Abdelali, A., Darwish, K., Durrani, N., Mubarak, H.: Farasa: A fast and furious segmenter for arabic. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, pp. 11–16 (2016)
21. Attia, M., Zirikly, A., Diab, M.: The power of language music: Arabic lemmatization through patterns. In: Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon, CogALex@COLING 2016, Osaka, December 12, 2016, pp. 40–50 (2016). https://aclanthology.info/papers/W16-5306/w16-5306
22. Al-Shammari, E., Lin, J.: A novel Arabic lemmatization algorithm. In: Proceedings of the Second Workshop on Analytics for Noisy Unstructured Text Data, pp. 113–118 (2008). http://doi.acm.org/10.1145/1390749.1390767

23. El-Shishtawy, T., El-Ghannam, F.: An accurate Arabic root-based lemmatizer for information retrieval purposes. CoRR **abs/1203.3584** (2012). http://arxiv.org/abs/1203.3584
24. Diab, M.: Improved Arabic base phrase chunking with a new enriched POS tag set. In: Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources, pp. 89–96 (2007). https://www.aclweb.org/anthology/W07-0812
25. Darwish, K., Mubarak, H.: Farasa: A new fast and accurate Arabic word segmenter. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16) (2016)
26. Maamouri, M., Bies, A., Buckwalter, T., Mekki, W.: 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)
27. El-Haj, M., Koulali, R.: KALIMAT a multipurpose Arabic corpus. In: Second Workshop on Arabic Corpus Linguistics (WACL-2), pp. 22–25 (2013)
28. Freihat, A., Bella, G., Mubarak, H., Giunchiglia, F.: A single-model approach for Arabic segmentation, POS tagging, and named entity recognition. In: 2018 2nd International Conference on Natural Language and Speech Processing (ICNLSP), pp. 1–8 (2018)
29. Freihat, A., Abbas, M., Bella, G., Giunchiglia, F.: Towards an optimal solution to lemmatization in Arabic. In: Proceedings of the 4th International Conference on Arabic Computational Linguistics (ACLing 2018), pp. 1–9 (2018)
30. Shaalan, K.: A survey of Arabic named entity recognition and classification. Comput. Linguist. **40**, 469–510 (2014)
31. Dukes, K., Habash, N.: Morphological annotation of quranic Arabic. In: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10) (2010)
32. Toutanova, K., Klein, D., Manning, C., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1, pp. 173–180 (2003). https://doi.org/10.3115/1073445.1073478