POS-tagging of Tunisian Dialect Using Standard Arabic Resources and Tools

Ahmed Hamdi[1] **Alexis Nasr**[1] **Nizar Habash**[2] **Núria Gala**[1]
(1) Laboratoire d'Informatique Fondamentale de Marseille, Aix-Marseille University
(2) New York University University Abu Dhabi
{ahmed.hamdi,alexis.nasr,nuria.gala}@lif.univ-mrs.fr
nizar.habash@nyu.edu

Abstract

Developing natural language processing tools usually requires a large number of resources (lexica, annotated corpora, etc.), which often do not exist for less-resourced languages. One way to overcome the problem of lack of resources is to devote substantial efforts to build new ones from scratch. Another approach is to exploit existing resources of closely related languages. In this paper, we focus on developing a part-of-speech tagger for the Tunisian Arabic dialect (TUN), a low-resource language, by exploiting its closeness to Modern Standard Arabic (MSA), which has many state-of-the-art resources and tools. Our system achieved an accuracy of 89% (∼20% absolute improvement over an MSA tagger baseline).

1 Introduction

The Arabic language is characterized by diglossia (Ferguson, 1959) : two linguistic variants live side by side: a standard written form and a large variety of spoken dialects. While dialects differ from one region to another, the written variety, called Modern Standard Arabic (MSA), is generally the same. MSA, the official language for Arabic countries, is used for written communication as well as in formal spoken communications. Spoken varieties, generally used in informal daily discussions, are increasingly being used for informal written communication on the web. Such unstandardized varieties differ from MSA with respect to phonology, morphology, syntax and the lexicon. Unlike MSA which has an important number of NLP resources and tools, Arabic dialects are less-resourced. In this paper, we focus on the Tunisian Arabic dialect (TUN). It is the spoken language of twelve million speakers living mainly in Tunisia. TUN is the result of interactions and influences of a number of languages including Arabic, Berber and French (Mejri et al., 2009).

In this paper, we focus on the development of a part-of-speech (POS) tagger for TUN. There are two main options when developing such a tool for TUN. The first one is to build a corpus of TUN, which involves recording, transcribing and manually POS tagging. In order to have a state-of-the-art POS tagger one also needs to develop a lexicon. The second option is to *convert* TUN into an approximate form of MSA, that we will call pseudo MSA, and use an existing MSA POS tagger. We intentionally do not use the verb *translate* to describe the process of transforming a TUN text into a pseudo MSA text. The reason being that we are not translating between two natural languages: pseudo MSA is not meant to be read by humans. Its only purpose is to be close enough to MSA so that running it through NLP tools would give good results. The annotation produced is then projected back on the TUN text. More technically, the conversion process focuses on morphological and lexical aspects; it is based on morphological analyzers and generators for TUN and MSA as well as a TUN-MSA dictionaries which are themselves partly automatically produced using the morphological analyzers and generators. Besides producing a POS tagger for TUN, we aim at proposing a general methodology for developing NLP tools for dialects of Arabic.

The rest of the paper is organized as follows: we present, in section 2, phonological, lexical and morphosyntactic variations between TUN and MSA. We then discuss related works and existing POS taggers of Arabic dialects in section 3. Section 4 reviews the tools and resources used

Proceedings of the Second Workshop on Arabic Natural Language Processing, pages 59–68,
Beijing, China, July 26-31, 2015.

in this work. In section 5, we describe in detail our approach to tag TUN texts. Finally, Section 6 presents results evaluating our approach under several conditions.

2 Linguistic variations between MSA and TUN

The TUN dialect differs from MSA on the phonological, lexical, morphological, and syntactic levels. In this work, we focus on the three first levels.

- **phonological and orthographic variations**: TUN has all phonemes that exist in MSA. However, TUN has three extra phonemes /p/, /v/ and /g/. To a lesser extent, variations appear in some common words, that consist in dropping some short vowels[1] on the TUN side. For instance, كتاب *ktAb*[2] "*book*" and كتب *ktb* "*to write*" which exist in both languages but are pronounced differently: /kitAb/, /katab/ in MSA and /ktAb/, /ktib/ in Tunisian dialect. Concerning orthography, unlike MSA, which already has a standard orthography, Tunisian dialect is unstandardized. Zribi et al. (2014) proposes orthographic standards for TUN, following the works of Habash et al. (2012), that aim to establish a common orthographic convention for all Arabic dialects.
- **lexical variations**: from a lexical point of view, the differences between MSA and TUN are significant. They are mainly due to the influence of other languages. Such TUN words still generally follow MSA morphology, sharing the same inflectional and derivational rules. Table 1 gives some examples of words of different origins.
- **morphological variations**: All morphological phenomena that exist in MSA exist also in TUN, but they are sometimes expressed differently. As cliticization is concerned, several MSA prepositions are attached to words on the TUN side. For example, the MSA prepositions على *ςalaý* "*on*" and من *mino* "*from*" become in TUN respectively +ع ς+ and +م m+ proclitics when the word following is definite (marked by the determinant marker +ال Al+). Furthermore, indirect object pronouns are realized as enclitics in TUN verbs and not in MSA. On the other hand, some MSA clitics are detached in TUN. The MSA future particle proclitic +س *sa+* is realized as the autonomous particle باش *bAš* with TUN verbs. As for inflectional morphology, MSA has a richer system than TUN. In fact, MSA nominal case and verbal mood do not exist in TUN. The three MSA number values (singular, dual and plural) are reduced to singular and plural. On TUN side, the masculine and the feminine plural are consolidated. Concerning derivational morphology, TUN words, except loanwords, keep the same principle of word's derivation from a root and a pattern as MSA. The TUN words حجّم *Haj∼im "cap"* and حجّام *Haj∼Am "hair dresser"* are both derived from the root ح ج م *H j m* and the patterns *1a22i3* and *1a22A3* respectively.

MSA	TUN	gloss	origin
تين tiyn	كرموس karmuws	fig	Berber
ولّاعة wal∼Aςaħ	بريكيّة briykiy∼aħ	lighter	French
مكتب بريد maktab bariyd	بوسطة buwSTaħ	post office	Italian
أسود Âaswad	أكحل ÂakHil	black	Arabic
باخرة bAxiraħ	بابور bAbuwr	boat	Turkish

Table 1: Examples of lexical variations between TUN and MSA

3 Related work

Processing Arabic dialects

Most studies concerning Arabic dialects focus on Egyptian, Levantine and Iraqi. Some efforts have been done to create dialectal resources such as Al-Sabbagh and Girju (2010) who built an Egyptian/MSA lexicon exploiting available data from the web. Other researchers focused on building parallel corpora between Arabic dialects, MSA and English (Zbib et al., 2012; Bouamor et al., 2014; Harrat et al., 2014). Habash et al. (2008) and Elfardy and Diab (2012) proposed some standard guidelines for the annotation of Arabic dialects. Other efforts focused in dialect identification (Habash et al., 2008; Elfardy and Diab, 2013; Zaidan and Callison-Burch, 2014) and

[1]In Arabic orthography, short vowels are represented with optional diacritics which makes the language ambiguous.

[2]Arabic orthographic transliteration is presented in the Habash-Soudi-Buckwalter HSB scheme (Habash et al., 2007).

machine translation (Sawaf, 2010; Salloum and Habash, 2011; Sajjad et al., 2013). Concerning morphosyntactic analysis, Al-Sabbagh and Girju (2012) implemented a POS tagger of Egyptian trained on data extracted from the web. Chiang et al. (2006) developed lexicons and morphological rules to build Levantine treebanks from MSA resources in order to parse Levantine dialect.

POS tagging of one language using another language

There have been several attempts to build POS taggers for one language using resources and tools of other languages. The idea consists in transforming the source language for which more resources are available into a target language (Yarowsky et al., 2001), using, for instance, parallel corpora. The source side is tagged using an available tagger, the annotations are then projected on the target. Subsequently, a new tagger is trained on the target side. In the same way, (Das and Petrov, 2011) used a graph-based projection algorithm to project tags across eight European languages. Following this work, (Duong et al., 2013) showed that focusing on selected informative training sentences from the parallel corpus and employing self-training achieve equivalent performance. All these studies concerned unrelated languages.

This approach is more effective when the source and the target languages are closely related. Many researchers exploit this fact to create resources and tools for under-resourced languages using other related well-resourced languages. Duong et al. (2013), for example used the approach based on parallel corpora to build a POS tagger for some European languages. Some efforts looked into dictionaries extracted from Wikitionary instead of parallel corpora (Li et al., 2012) and others combined both resources (Täckström et al., 2013). Other approaches propose to adapt existing taggers of a more-resourced close related languages for miss-resourced languages. Feldman et al. (2006) built taggers for Czech and Catalan starting from existing Russian and Spanish taggers respectively. They trained the taggers on the source language and then adapt its parameter files on the target language by means of a list of cognate word pairs. Similarly, Bernhard et al. (2013) adapted a German tagger to Alsatian. Vergez-Couret (2013) showed that building POS taggers for less-resourced language using annotated corpora for a more-resourced related language is possible by translating only the most frequent words from the source side to the target side. In their experiments, they built two bilingual Occitan/French and Occitan/Castillan lexica of about 300 entries. After translating the most frequent words, existing French and Castillan taggers have been run on Occitan texts.

POS tagging of Arabic dialects

Concerning POS tagging of Arabic dialects, few efforts focused on creating resources for such dialects. (Al-Sabbagh and Girju, 2012) built an Egyptian POS tagger trained on manually annotated corpus of $400K$ tokens extracted from written Arabic social networking. They report an accuracy of 94% in tokenization and 88% in POS tagging. Similarly, Mohamed et al. (2012) annotated a small corpus to train an Egyptian tokenizer. Their system's performance reaches 91%. Some other efforts used existing tools of related languages as starting material to build POS taggers for dialects. The first system proposed by Duh and Kirchhoff (2005), built a Levantine and Egyptian POS tagger using raw text corpora and an existing MSA analyzer. Their POS accuracy achieves 71%. Similarly, Habash et al. (2013) and Pasha et al. (2014) developed an Egyptian morphological analyzer using two systems for Arabic morphology processing: MADA (Habash and Rambow, 2005; Roth et al., 2008) and AMIRA (Diab et al., 2013), they report 92.4% of POS accuracy on Egyptian Arabic.

Tunisian morphology processing

Processing Tunisian morphology has not been the object of many studies. Zribi et al. (2013) adapted an existing MSA morphological analyzer to handle TUN. In order to build such a tool, they used a TUN-MSA lexicon to add specific TUN roots and patterns. Their system achieved an F-measure performance of 88% in morphological analysis. In a similar setting, Boujelbane et al. (2014) used the same lexicon to transform a MSA training corpus to create a large TUN corpus. This resource was used to train a POS tagger. POS tagging of TUN transcribed texts using this tagger and achieved an accuracy of 78.5%.

Our approach is close to Boujelbane et al. (2014): we built a POS tagger for a less-resourced variant of a language using a system trained on an annotated close related language. Our approach differs from their mostly on the morphological processing: we perform a deeper morphological analysis, which allows us to generate a lemmatized version of the MSA text. We will show that performing the POS tagging at this level yields better results.

4 Tools and resources

In this section, we describe the various resources and tools we used in our experiments. We first describe MAGEAD, a morphological analyzer/generator. Then, we detail three lexica that relate MSA and TUN lemmas.

4.1 Morphological analysis and generation of Arabic and its dialect

MAGEAD is a morphological analyzer and generator for the Arabic language family (MSA and Arabic dialects). It processes Arabic verbs (Habash and Rambow, 2006; Habash et al., 2005) and Arabic nouns (Altantawy et al., 2010).

MAGEAD relates a deep representation of a word with its surface form through a sequence of transformations. It can be used bidirectionally, to generate, as well as to analyze, surface forms. At a deep representation level, MAGEAD represents a word as a root, a pattern and a set of feature-value pairs. The features are translated to abstract morphemes which are then ordered, and expressed as concrete morphemes. Finally, morphological and phonological rewrite rules are applied. To describe the different processes made by MAGEAD, we use the surface form واضطرّوا *waAiDTar∼uwA* "*and they were obliged*" as our example. The MAGEAD lexeme and features representation of this word form is as follows:

(1) root:Drr mbc:verb-VIII cnj:w per:3 gen:m num:prl asp:p vox:a

The lexeme is defined as the root *Drr* and a morphological behavior class (MBC) *verb-VIII*. The MBC maps sets of linguistic feature-value pairs to sets of abstract morphemes (AMs). In our example, the MBC verb-III maps asp:p and vox:a to the AM [PAT_PV:VIII][VOC_PV:VIII-act]. The feature value cnj:w is simply mapped to the AM [CNJ:W] while the features values per:3 gen:m num:prl asp:p is mapped to the AM [SUBJ_SUFF:3MP]. AMs are then ordered. At this point our example is represented as:

(2) [CNJ:W] + [ROOT:Drr] [PAT_PV:VIII] [VOC_PV:VIII-act] + [SUBJ_SUFF:3MP]

Note that the root, pattern, and vocalism are not ordered with respect to each other, they are simply juxtaposed. The '+' sign indicates the ordering of affixational morphemes. AMs are then mapped to CMs, which are concatenated in the specified order. Our example becomes:

(3) wa + Drr,V1tV2V3,iaa + uwA

Simple interdigitation of root, pattern and vocalism then yields the form (4) wa+iDtarar+uwA. At this point MAGEAD applies (if they exist) rules of the following type:

- Morphophonemic/phonological rules map the morphemic representation to the phonological and orthographic representations. In our example, two rules are applied. First, the gemination[3] rule, which allows to delete the vowel between the second and the third radical if it is followed by a suffix starting with a vowel. Then, a phonological rule that transforms the /t/ of the pattern *i1ta2a3* to /T/.[4] We get, at this step: /wa+iDTar∼+uwA/.
- Orthographic rules rewrite the orthographic representation. Using standard MSA diacritized orthography, our example becomes واضطرّوا *waAiDTar∼uwA*.

MAGEAD follows (Kiraz, 2000) in using a multi-tape representation. It extends the analysis of Kiraz by introducing a fifth tier. The five tiers are the following :

- Tier 1: pattern and affixational morphemes
- Tier 2: root
- Tier 3: vocalism
- Tier 4: phonological representation
- Tier 5: orthographic representation

In the generation direction, tiers 1 through 3 are input tiers. Tier 4 is an output tier, and an input tier for the orthographic representation.

MAGEAD handles Arabic nouns in the same way. Specific CMs, AMs and morpheme order are defined for nouns. The MBC hierarchy specifies relevant morphosyntactic features such as rationality. The MBC class name indicates the vocalized patterns according to the number and the gender values. Many nominal rules are similar to those presented for verbs. Others are specific, reflecting

[3] A geminate root is a root in which the second and the third radical are identical.

[4] The /t/ of the pattern *i1ta2a3* is converted to /T/ when the first root radical corresponds to /D/, /T/ or /Ď/.

the differences between Arabic nominal and verbal morphology.

We adapted MAGEAD to process TUN. Changes concerned only the representation of linguistic knowledge, leaving the processing engine unchanged. We modified the MBC hierarchy, in order to process TUN patterns and vocalisms. The AM ordering has been modified and new AMs have been added. The mapping from AMs to CMs and the definition of rules, which are variant-specific, have been written by a linguistically trained native speaker.

We also modified a number of morphophonemic rules in the TUN implementation. We briefly describe three changes. First, in MSA, the gemination rule deletes the vowel between the second and the third radical if it is followed by a suffix starting with a vowel: e.g., compare مددت *madad+tu* '*I extended*' with مدّت *mad∼+at*'*she extended*' (NOT *madad+at*). In TUN, however, a long vowel is inserted before consonant-initial suffixes following geminate verbs: مدّيت *mad∼+iy+t* "*I extended*" and مدّت *mad∼+it* "*she extended*". Second, unlike MSA, the first root radical in TUN becomes a long vowel in the imperfective aspect when it corresponds to ء ' (*hamza/glottal stop*) (يأكل *yÂkl* becomes ياكل *yAkl* '*he/it eats*'). Finally, TUN verbs whose root ends with ء ', behave the same way as verbs whose final root radical ي *y* in the perfective aspect. For example, roots of TUN verbs بدينا *bdiynA* "*we started*" and رمينا *rmiynA* "*we threw*" are respectively ب د ء *bd'* and ر م ي *rmy*. For more details, see (Hamdi et al., 2013).

4.2 Lexica

Due to the lexical differences between MSA and TUN, the conversion process cannot be limited to morphological transformations and requires some lexical transformations. We used three lexica to map from TUN to MSA: a lexicon of verbs, a lexicon of deverbal nouns and a lexicon of particles.

4.2.1 Lexicon of verbs

The verbal lexicon consists of pairs of the form (P_{MSA}, P_{TUN}) where P_{MSA} and P_{TUN} are themselves pairs made of a root and a pattern. Its development was based on the Penn Arabic Tree Bank (*PATB*) (Maamouri et al., 2004) which contains $29,911$ verb tokens. Each token was then analyzed to extract its root and its pattern. Each lemma was translated, in context, to TUN by a Tunisian native speaker. Since the lemma is the result of combining a root and a pattern, the TUN pair (root, pattern) can be deduced. This process allowed us to define about 100 new roots for TUN. The lexicon contains $1,638$ entries. The TUN side contains 920 distinct pairs and the MSA side $1,478$ distinct pairs. This difference shows that MSA is lexically richer than TUN. On average, a TUN lemma corresponds to almost two MSA lemmas. For instance, the TUN verb مشى *mšaý* matches with MSA verbs ذهب *ðahab* '*to go*' and مشى *mašaý* '*to walk*'. The maximum ambiguity is 16 in the TUN $\rightarrow$ MSA direction and 4 in the opposite direction.

4.2.2 Lexicon of deverbal nouns

This lexicon is automatically built using the lexicon of verbs. In fact, many deverbal nouns can be derived from verbs such as participles, infinitive forms, adjectives, nouns of time and place ... The deverbal noun is produced by combining a root and a deverbal pattern. The deverbal patterns are derived from verbal patterns. Each pair (root, pattern) of the verbal lexica generates many deverbal entries by combining the root with all deverbal patterns that share the same meaning on both sides. This method overgenerates and can produce wrong pairs. In order to face this problem, we filtered the MSA part using the MSA large-scale lexicon SAMA (Graff et al., 2009). At the end of the process, a lexicon made of $33,271$ entries is created (Hamdi et al., 2014).

4.2.3 Lexicon of particles

Arabic particles cover many categories: conjunctions, prepositions, clitics ... Our lexicon, made of about 200 pairs (MSA particle, TUN particle), includes all of them. The MSA particles are extracted from the PATB and then translated to TUN (Boujelbane et al., 2013). In its current version, the lexicon matches 262 Tunisian particles to 143 MSA particles.

5 Architecture and experiments

Our system consists of three step: conversion, disambiguation and POS tagging.

The TUN input sentence $t_1\ t_2\ t_3\ \ldots t_n$, is converted to a MSA lattice. The lattice is then disambiguated to produce a pseudo MSA target sentence $m_1\ m_2\ m_3\ \ldots m_n$. Next, a MSA tagger assign to each target word its POS tag. The disambiguation step is optional, the MSA lattice can be sent directly to the POS tagger which tags the lattice and produces the most likely tag sequence.

Taking as an example the TUN sentence تجبر باش يقعد *tijbar bAš yuqςud* '*he was obliged to stay*', which correspond to the sequence of POS tags *verb-pass*[5] - *part* - *verb*. This sentence translates into MSA as اضطرّ إلى البقاء *AiðTar∼a Ǎilaý AlbaqA'*. Our system produces for this sentence, after conversion and disambiguation, the sentence اضطرّ سوف يجلس *AuðTur∼a sawfa yajlisu* '*he was obliged will sit-down*' which receives the correct POS tags sequence *verb-pass* - *part* - *verb*, although the MSA translation is suboptimal. In the remainder of this section, we describe in detail each step of the whole process.

5.1 Conversion

The process of converting a source TUN word form to a target MSA form proceeds in three main steps: morphological analysis using MAGEAD for the source language, lexical transfer and morphological generation of target MSA forms. Figure 1 describes the process that allows to switch from a TUN source input to a MSA target output.

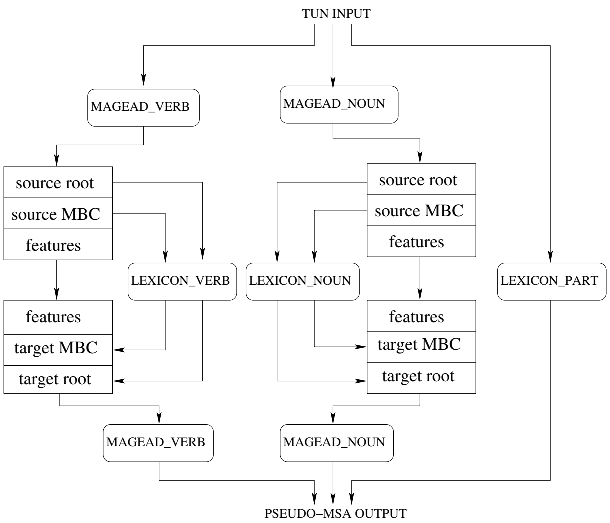

Figure 1: TUN-to-MSA conversion

Each TUN source word is processed by MAGEAD to produce several analyses; each of them is compound of a root, a pattern and a set of feature-value pairs. The root and the pattern are translated to a MSA root and pattern by a lexicon lookup. MAGEAD finally uses the target root and pattern and the feature-value pairs to generate a target MSA word.

This process was evaluated on $1,500$ tokens of TUN verbal forms that were identified and translated in context to MSA by Tunisian native speakers. Table 2 gives the accuracy and the ambiguity resulting from the translation. The recall indicates the proportion of cases where the correct target form was produced while the ambiguity indicates the number of target forms produced on average for an input.

recall		ambiguity	
tokens	types	tokens	types
76.43%	74.52%	26.82	25.57

Table 2: Recall and ambiguity on translation of TUN verbs to MSA

In order to extend the coverage of the lexica, we introduced a back-off process. When a pair (root, MBC) is missing in the noun or the verb lexicon, the root and MBC are translated separately, using a root lexicon and an MBC correspondence table. The root lexicon is made of pairs (r_{MSA}, r_{TUN}), where r_{MSA} is a MSA root and r_{TUN} is a TUN root. The root lexicon contains $1,329$ entries. The MBC correspondence tables indicates, for a TUN MBC, the most frequent corresponding MBCs on the MSA side. In cases of lexicon look-up failure, the MSA target word is produced by combining the target root lexicon and the target pattern. Table 3 gives the accuracy and the ambiguity resulting of the back-off process.

recall		ambiguity	
tokens	types	tokens	types
79.71%	78.94%	29.16	28.44

Table 3: Recall and ambiguity on translation of TUN verbs to MSA with back-off

Table 3 shows that this back-off mechanism reaches a reasonable recall but the price to pay is a high ambiguity. More details are given in (Hamdi et al., 2013).

5.2 Disambiguation

The conversion process contains two sources of ambiguity: the morphological analysis can create multiple outputs and the lexica may propose for a TUN input many MSA outputs. Each word in the TUN sentence is translated into a set of MSA words producing a lattice. The disambiguation can

[5] verb in the passive form

be performed by the POS tagger, as we will see below or it can be done independently, using a language model. We have trained a 1-gram and a 3-gram language models on a two million word MSA corpus. This corpus is itself made of two corpora. The first one is a written corpus, it is a collection of reports of the French press agency (AFP). The second one is a spoken corpus, it is a collection of political debates transcriptions. The trigram model is used to give the first best path while the unigram allowed to filter and score the lattice.

Three different inputs can be handled by the POS tagger: an unscored lattice derived from the conversion, a scored lattice produced by the disambiguation based on the unigram language model and the first best path generated by the 3-gram language model.

5.3 Pos-Tagging

The taggers used in this work are based on Hidden Markov Models (HMM). We have chosen this type of model mainly for their ability to take word lattices as input in a straightforward way. The tagger itself is a weighted finite state transducer and the tagging process is performed by a composition operation of the word lattice and the tagger, followed by a best path operation. When the tagger is fed with a lattice produced by the conversion step (containing potentially several MSA forms for a TUN form), the tagger actually does more than POS tagging, it also selects a sequence of words from the word lattice.

We built six taggers that differ in the order of the HMM they are based on (bigram or trigram) as well as in the nature of the observables of the HMM: forms, lemmas and *lmms*. The latter is the undiacritized form of a lemma. There are two main reasons for using lemmas and *lmms* based taggers: first, the translation task is more accurate and gives less ambiguity for lemmas and *lmms* than for forms. Second, the POS tagging achieves better results on lemmas and *lmms* than on forms, as shown in Table 4.

The taggers are trained on the Penn Arabic Treebank (PATB) Part 3 (Maamouri et al., 2004) in the representation of the Columbia Arabic Treebank (CATIB) (Habash and Roth, 2009). The corpus is made from $24K$ MSA sentences compound of $330K$ tokens and $30K$ types. The CATIB POS tagset consists of six tags only: nominal, proper noun, verb, verb-pass, particle and punctuation.

Table 4 gives the results of POS tagging of a MSA corpus using our different HMM taggers. These results are comparable to state-of-the-art MSA POS tagging systems: Habash and Roth (2009) report a higher result using the MADA system (Habash and Rambow, 2005). However, we cannot use the MADA system because it does not support POS tagging over a lattice, which we need for TUN POS tagging. It should be noted that the results in the table are for forms (real task), but also for gold lemmas and *lmms*. We present the lemma and *lmm* results only for comparative reasons as the starting point is artificial, and the performance numbers should be seen as upper bounds.

	forms	gold lemmas	gold *lmms*
bigram	94.52	97.61	96.84
trigram	94.72	97.63	96.94

Table 4: Accuracy of POS tagging of MSA corpus

The results in the table suggest that using the trigram HMM is slightly better than the bigram HMM models. For the rest of this paper, we will report only using the trigram model.

6 Evaluation

In order to evaluate our method, we used a transcribed and annotated corpus of 805 sentences containing $10,746$ tokens and $2,455$ types. These sentences were obtained from several sources: TV series and political debates, a transcribed theater play (Dhouib, 2007) and a transcribed corpus made of conversations between a customer and a railways officer. This selection aims to include different TUN spoken varieties. After transcribing, we have assigned to each token its lemma, *lmm* and POS tag using the same conventions as the corpus used to train the tagger.

Our baseline experiment consists of running the MSA POS tagger directly on TUN texts without any processing. This baseline will allow us to measure the contribution of converting TUN to pseudo MSA prior to POS tagging with the MSA tagger. The accuracy of tagging and the number of out-of-vocabulary words are given in Table 5. The lemmas and *lmms* used for the experiment are gold lemmas and *lmms*, presented again for comparative reasons. Our official baseline is with forms.

Table 5 shows that the baseline is very low, around 69%. The result on lemmas is even worse.

	forms	gold lemmas	gold *lmms*
accuracy	69.04%	67.41%	71.41%
OOVs	2891 26.90%	4766 44.35%	2705 25.17%

Table 5: Baseline Accuracy of POS tagging TUN using MSA POS tagger

This is not unexpected since the TUN lemma space is different from the MSA lemma space, which the tagger is trained on. Lemmas are completely diacritized and diacritics on lemmas are different on MSA and on TUN. For instance, the TUN undiacritized form يكتب *yktb* "*he writes*" exists in MSA side but its lemma *ktib* "*to write*" is different from the MSA one *katab*. Results are a bit higher on *lmms*, which do not contain diactritics. It is also interesting to note that the number of OOVs on *lmms* is still high, showing that lexica of MSA and TUN are quite different.

For our main experiment we convert TUN texts to pseudo MSA before POS tagging. The conversion step produces three lattices (forms, lemmas, *lmms*). The form lattice is disambiguated by the language models providing a scored lattice and the first best path. We ran the POS tagging of pseudo-MSA forms in three modes: on the best form path, on the scored lattice and the unscored lattice produced by the conversion. The final output is the sequence of POS tags for the words in the original sentence. Results are shown in Table 6.

	best path	scored lattice	unscored lattice
accuracy	77.2%	80.3%	82.5%
OOVs	16.9%	15.3%	13.5%

Table 6: Accuracy of POS tagging of pseudo MSA

Results show that the conversion decreases the number of OOVs and subsequently the POS-tagging accuracy of forms increases (comparing with Table 5). Disambiguation based on the POS tagger gives better accuracy (∼82.5% on forms) than the language model (77.2%).

Our convertion process allows to produce, MSA lemmas and *lmms* rather then forms by leaving the morphological generation of MSA forms. The POS tagger was ran thus on the lattices of lemmas and *lmms*. In Table 7, we give results of POS tagging such inputs. We give again results on forms to compare these final results with the basline results (Table 5).

	forms	*predicted* lemmas	*predicted* *lmms*
accuracy	82.5%	86.9%	**89.1%**
OOVs	13.5%	6.2%	4.9%

Table 7: Accuracy of POS tagging of pseudo MSA lemmas and *lmms*

As shown in Table 7, POS tagging of lemmas and *lmms* outperforms POS tagging of forms. Our best accuracy, with *lmms*, jumps to 89.1%: a 20% absolute increase of the baseline of using the MSA POS tagger directly on the TUN sentences. An error analysis of the first 100 errors shows that 34 of them are due to bad conversion and 49 to bad disambiguation. Only, 17 of the errors came from POS tagging.

7 Conclusion

In this paper, we proposed, implemented and evaluated an approach to POS tagging of TUN using an MSA tagger. Prior to tagging, the TUN text is converted to pseudo MSA. The conversion process is composed of three steps: morphological analysis of the TUN words, followed by a lexical transfer and a morphological generation of MSA forms. The system achieved an accuracy of 89% (∼20% absolute improvement over an MSA tagger baseline). Experiments showed that the best results were obtained by tagging at the level of lemmas, more precisely, lemmas from which diacritics were removed.

In future work, we aim to complete our processing chain by adding a TUN speech recognition system (since TUN is a primarily spoken language) at the beginning of the chain, and to evaluate our approach in some other NLP tasks such as syntactic parsing. We are also interested in applying these results to other dialects.

Acknowledgments

This paper is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract Nos. HR0011-12-C-0014 and HR0011-12-C-0016. Any opinions, findings and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of DARPA.

References

Rania Al-Sabbagh and Roxana Girju. 2010. Mining the web for the induction of a dialectical Arabic lexicon. In *LREC*.

Rania Al-Sabbagh and Roxana Girju. 2012. A supervised pos tagger for written Arabic social networking corpora. In *Proceedings of KONVENS*, pages 39–52.

Mohamed Altantawy, Nizar Habash, Owen Rambow, and Ibrahim Saleh. 2010. Morphological Analysis and Generation of Arabic Nouns: A Morphemic Functional Approach. In *Proceedings of the seventh International Conference on Language Resources and Evaluation (LREC)*, Valletta, Malta.

Delphine Bernhard, Anne-Laure Ligozat, et al. 2013. Hassle-free pos-tagging for the alsatian dialects. *Non-Standard Data Sources in Corpus Based-Research*.

Houda Bouamor, Nizar Habash, and Kemal Oflazer. 2014. A Multidialectal Parallel Corpus of Arabic.

Rahma Boujelbane, Siwar BenAyed, and Lamia Hadrich Belguith. 2013. Building bilingual lexicon to create dialect Tunisian corpora and adapt language model. *ACL 2013*, page 88.

Rahma Boujelbane, Mariem Mallek, Mariem Ellouze, and Lamia Hadrich Belguith. 2014. Fine-grained pos tagging of spoken Tunisian dialect corpora. In *Natural Language Processing and Information Systems*, pages 59–62. Springer.

David Chiang, Mona T Diab, Nizar Habash, Owen Rambow, and Safiullah Shareef. 2006. Parsing Arabic dialects. In *EACL*.

Dipanjan Das and Slav Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 600–609. Association for Computational Linguistics.

E Dhouib. 2007. El makki w zakiyya. Maison d'Ãl'dition manshuwrat manara, Tunis.

Mona Diab, Nizar Habash, Owen Rambow, and Ryan Roth. 2013. Ldc Arabic treebanks and associated corpora: Data divisions manual. *arXiv preprint arXiv:1309.5652*.

Kevin Duh and Katrin Kirchhoff. 2005. Pos tagging of dialectal Arabic: a minimally supervised approach. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, pages 55–62. Association for Computational Linguistics.

Long Duong, Paul Cook, Steven Bird, and Pavel Pecina. 2013. Simpler unsupervised pos tagging with bilingual projections. In *ACL (2)*, pages 634–639.

Heba Elfardy and Mona T Diab. 2012. Simplified guidelines for the creation of large scale dialectal Arabic annotations. In *LREC*, pages 371–378.

Heba Elfardy and Mona T Diab. 2013. Sentence level dialect identification in Arabic. In *ACL (2)*, pages 456–461.

Anna Feldman, Jirka Hana, and Chris Brew. 2006. A cross-language approach to rapid creation of new morpho-syntactically annotated resources. In *Proceedings of LREC*, pages 549–554.

Charles Albert Ferguson. 1959. Diglossia. *WORD-JOURNAL OF THE INTERNATIONAL LINGUISTIC ASSOCIATION*, 15(2):325–340.

David Graff, Mohamed Maamouri, Basma Bouziri, Sondos Krouna, Seth Kulick, and Tim Buckwalter. 2009. Standard Arabic Morphological Analyzer (SAMA) Version 3.1. Linguistic Data Consortium LDC2009E73.

Nizar Habash and Owen Rambow. 2005. 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*, pages 573–580. Association for Computational Linguistics.

Nizar Habash and Owen Rambow. 2006. Magead: a morphological analyzer and generator for the Arabic dialects. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 681–688. Association for Computational Linguistics.

Nizar Habash and Ryan M Roth. 2009. Catib: The columbia Arabic treebank. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 221–224. Association for Computational Linguistics.

Nizar Habash, Owen Rambow, and George Kiraz. 2005. Morphological analysis and generation for Arabic dialects. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, pages 17–24. Association for Computational Linguistics.

Nizar Habash, Abdelhadi Soudi, and Tim Buckwalter. 2007. On Arabic Transliteration. In A. van den Bosch and A. Soudi, editors, *Arabic Computational Morphology: Knowledge-based and Empirical Methods*. Springer.

Nizar Habash, Owen Rambow, Mona Diab, and Reem Kanjawi-Faraj. 2008. Guidelines for annotation of Arabic dialectness. In *Proceedings of the LREC Workshop on HLT & NLP within the Arabic world*, pages 49–53.

Nizar Habash, Mona T Diab, and Owen Rambow. 2012. Conventional orthography for dialectal Arabic. In *LREC*, pages 711–718.

Nizar Habash, Ryan Roth, Owen Rambow, Ramy Eskander, and Nadi Tomeh. 2013. Morphological analysis and disambiguation for dialectal Arabic. In *HLT-NAACL*, pages 426–432. Citeseer.

Ahmed Hamdi, Rahma Boujelbane, Nizar Habash, and Alexis Nasr. 2013. The effects of factorizing root and pattern mapping in bidirectional Tunisian - standard Arabic machine translation. In *MT Summit, Nice*.

Ahmed Hamdi, Núria Gala, and Alexis Nasr. 2014. Automatically building a Tunisian lexicon for deverbal nouns. *COLING 2014*, page 95.

S Harrat, K Meftouh, M Abbas, and K Smaili. 2014. Building resources for algerian Arabic dialects. *Corpus (sentences)*, 4000(6415):2415.

George Anton Kiraz. 2000. Multitiered nonlinear morphology using multitape finite automata: a case study on Syriac and Arabic. *Computational Linguistics*, 26(1):77–105, March.

Shen Li, Joao V Graça, and Ben Taskar. 2012. Wiki-ly supervised part-of-speech tagging. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1389–1398. Association for Computational Linguistics.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. 2004. The penn Arabic treebank: Building a large-scale annotated Arabic corpus. In *NEMLAR conference on Arabic language resources and tools*, pages 102–109.

S. Mejri, S. Mosbah, and I. Sfar. 2009. Pluringuisme et diglossie en tunisie. *Synergies Tunisie n 1*, pages 53–74.

Emad Mohamed, Behrang Mohit, and Kemal Oflazer. 2012. Annotating and learning morphological segmentation of egyptian colloquial Arabic. In *LREC*, pages 873–877.

Arfath Pasha, Mohamed Al-Badrashiny, Ahmed El Kholy, Ramy Eskander, Mona Diab, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. 2014. Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic. In *In Proceedings of the 9th International Conference on Language Resources and Evaluation, Reykjavik, Iceland*.

Ryan Roth, Owen Rambow, Nizar Habash, Mona Diab, and Cynthia Rudin. 2008. Arabic morphological tagging, diacritization, and lemmatization using lexeme models and feature ranking. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 117–120. Association for Computational Linguistics.

Hassan Sajjad, Kareem Darwish, and Yonatan Belinkov. 2013. Translating dialectal Arabic to english. In *ACL (2)*, pages 1–6.

Wael Salloum and Nizar Habash. 2011. Dialectal to standard Arabic paraphrasing to improve Arabic-english statistical machine translation. In *Proceedings of the First Workshop on Algorithms and Resources for Modelling of Dialects and Language Varieties*, pages 10–21. Association for Computational Linguistics.

Hassan Sawaf. 2010. Arabic dialect handling in hybrid machine translation. In *Proceedings of the 9th Conference of the Association for Machine Translation in the Americas*.

Oscar Täckström, Dipanjan Das, Slav Petrov, Ryan McDonald, and Joakim Nivre. 2013. Token and type constraints for cross-lingual part-of-speech tagging. *Transactions of the Association for Computational Linguistics*, 1:1–12.

Marianne Vergez-Couret. 2013. Tagging occitan using french and castillan tree tagger. In *Proceedings of 6th Language & Technology Conference*.

David Yarowsky, Grace Ngai, and Richard Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the first international conference on Human language technology research*, pages 1–8. Association for Computational Linguistics.

Omar F Zaidan and Chris Callison-Burch. 2014. Arabic dialect identification. *Computational Linguistics*, 40(1):171–202.

Rabih Zbib, Erika Malchiodi, Jacob Devlin, David Stallard, Spyros Matsoukas, Richard Schwartz, John Makhoul, Omar F Zaidan, and Chris Callison-Burch. 2012. Machine translation of Arabic dialects. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 49–59. Association for Computational Linguistics.

Inès Zribi, Mariem Ellouze Khemakhem, and Lamia Hadrich Belguith. 2013. Morphological analysis of Tunisian dialect. In *Proceeding of International Joint Conference on Natural Language Processing (IJCNLP 2013), Nagoya, Japan*.

Inès Zribi, Rahma Boujelbane, Abir Masmoudi, Mariem Ellouze, Lamia Belguith, and Nizar Habash. 2014. A conventional orthography for Tunisian Arabic. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*.