Using Stem-Templates to Improve Arabic POS and Gender/Number Tagging

Kareem Darwish, Ahmed Abdelali, Hamdy Mubarak

Qatar Computing Research Institute
Qatar Foundation, Doha, Qatar
{kdarwish,aabdelali,hmubarak}@qf.org.qa

Abstract

This paper presents an end-to-end automatic processing system for Arabic. The system performs: correction of common spelling errors pertaining to different forms of alef, ta marbouta and ha, and alef maqsoura and ya; context sensitive word segmentation into underlying clitics, POS tagging, and gender and number tagging of nouns and adjectives. We introduce the use of stem templates as a feature to improve POS tagging by 0.5% and to help ascertain the gender and number of nouns and adjectives. For gender and number tagging, we report accuracies that are significantly higher on previously unseen words compared to a state-of-the-art system.

Keywords: Part of Speech Tagging, Denormalization, Arabic

1. Introduction

Arabic is a Semitic language with derivational morphology. Arabic nouns, adjectives, adverbs, and verbs are typically derived from a closed set of 10,000 roots of length 3, 4, or rarely 5. Arabic nouns and verbs are derived from roots by applying templates to the roots to generate stems. Such templates may carry information that indicate morphological features of words such POS tag, gender, and number. For example, given a 3-letter root with 3 consonants CCC, a valid template may be CwACC , where the infix "wA"[1] is inserted, this template typically indicates an Arabic broken, or irregular, plural template for a noun of template CACC or CACCp if masculine or feminine respectively. Further, stems may accept prefixes and/or suffixes to form words. Prefixes include coordinating conjunctions, determiner, and prepositions, and suffixes include attached pronouns and gender and number markers. In this paper, we introduce an end-to-end Arabic processing system. The system performs several tasks, namely:

- Handling common spelling errors that are due to the erroneous use of: different forms of alef (A, >, <, |), alef maqsoura (Y) and ya (y) and ta marbouta (p), and ha (h).
- Segmenting words into their underlying clitics including properly identifying prefixes and suffixes.
- Detecting roots underlying stems and finding their stem templates.
- Performing POS tagging of words.
- Properly tagging nouns and adjectives with gender and number tags.

We use stem templates to improve POS tagging and to help tag nouns and adjectives with number and gender tags. The contributions of this paper are as follow:

- Introducing improved POS tagging using stem templates.
- Constructing a dataset of tagged nouns and adjectives that are tagged using gender and number.
- Using stem templates to improve the tagging of nouns and adjectives with gender and number.
- Developing an end-to-end system that performs common spelling mistake correction, word segmentation, POS tagging, and gender and number tagging.

Word	Segmentation	POS Tagging
تبين	تبين	تبين/V
للعلماء	ل+ال+علماء	علماء/NOUN+ال/DET +ل/PREP
أن	أن	أن/PART
عشرين	عشر+ين	ين/NSUFF+عشر/NUM
دقيقة	دقيق+ة	ة/NSUFF+دقيق/NOUN
من	من	من/PREP
الرياضة	ال+رياض+ة	ة/NSUFF+رياض/NOUN+ال/DET
في	في	في/PREP
اليوم	ال+يوم	يوم/NOUN+ال/DET
تساعد	تساعد	تساعد/V
على	على	على/PREP
إبعاد	إبعاد	إبعاد/NOUN
الإنفلونزا	ال+إنفلونزا	إنفلونزا/NOUN+ال/DET
بنسبة	ب+نسب+ة	ة/NSUFF+نسب/NOUN+ب/PREP
تقارب	تقارب	تقارب/NOUN
10%	10+%	10/NUM +%/PUNC
،	،	،/PUNC

Table 1: Sample output of the system.

The system is available for download from the QCRI web site[2]. Table 1 shows sample output of the segmentation/tokenization and POS tagging modules of the system which will be elaborated on further.

2. Background

Most recent work on Arabic word segmentation and POS tagging has used statistical methods. For example, Darwish (2002) attempted to solve this problem by developing

[1] Buckwalter encoding is used in the paper

[2] http://alt.qcri.org/tools/ArabicPOSTaggerLib/

a statistical morphological analyzer for Arabic called Sebawai that attempts to rank possible analyses and to pick the most likely one. Lee et al. (2003) developed IBM-LM, which adopted a trigram language model (LM) trained on a portion of the manually segmented Penn Arabic Treebank (PATB) in developing an Arabic morphology system, which attempts to improve the coverage and linguistic correctness over existing statistical analyzers such as Sebawai. IBM-LMs analyzer combined the trigram LM (to analyze a word within its context in the sentence) with a prefix-suffix filter (to eliminate illegal prefix suffix combinations, hence improving correctness) and unsupervised stem acquisition (to improve coverage). Diab (2009) used an SVM classifier to ascertain the optimal segmentation for a word in context and POS tagging. The classifier was trained on the PATB data. Essentially, she treated the problem as a sequence-labeling problem. Other popular systems for performing segmentation and POS tagging are MADA (Habash et al., 2009) and the Stanford POS tagger (Toutanova et al., 2003) both of which are trained on the PATB. Alkuhlani and Habash (2001) reported on their work on gender, number, and rationality tagging. For features, they used the leading and trailing letters in words, POS tags, form-based features, the lemmas, and syntactic features from a dependency parser. Their best accuracy results for previously unseen words were 89.7% and 91.2% for gender and number respectively.

3. System Description

3.1. Handling common mistakes

Some letter forms are frequently confused by authors and lead to common spelling mistakes. These letters are: “y” and “Y”, “h” and “p”, “A”, “|”, “>”, and “<” Replacing on form for another can change the word completely. For example, the word “ktbh” means “his books” while “ktbp” means “writers”; the word “ElY” is the preposition ”on” while “Ely” is the proper name ”Ali”. Other examples abound. To handle these errors we partially re-implemented the work of Moussa et al. (2012). Their experiments concluded that the technique that yielded the best accuracy was a bigram word-level language model with cascaded back-off to a unigram stem language model combined with a CRF model to handle the OOVs. In our re-implementation:

- We trained large unigram and bigram language models on a large corpora composed of Arabic Wikipedia and ten years of Aljazeera.net articles. These corpora generally had a small percentage of the aforementioned spelling mistakes.
- Given an input word, we consulted the unigram language model to generate other words that can be confused with it due to spelling mistakes.
- Given the different possible forms of the given word, we used the Viterbi algorithm (with the bigram language model) to find the best form in context.

Moussa et al. (2012) reported that such an approach yielded 98.9% accuracy on their test set. We tested on all of PATB and we achieved an accuracy of 99%.

3.2. Word Segmentation

Since prefixes, such determiners, coordinating conjunctions, and preposition, and suffixes, such as pronouns and gender and number markers, are often attached to words, it is important to properly segment words to ascertain the different clitics in words. We trained a statistical word segmenter that given a word aims to identify prefixes, stem, and suffixes. We trained it on 90% of parts 1, 2, and 3 of the PATB, and we retained 10% for testing. The stemmer uses sequence labeling with IOB notation at character level to identify the boundaries of clitics in a word. Every character is to be tagged with a label I (inside a sequence), O (outside a sequence) and B (beginning of a sequence). For example, the word ”wAlktAbyn”, the correct segmentation (and labels) would be as follows:
w - B; A - B; l - I; k - B; t - I; A - I; b - I; y - B; n - I
This would result in decomposing the word into the following clitics: w+Al+ktAb+yn. The stemmer uses a closed set of prefixes and suffixes to remove, such as: coordinating conjunctions (w, f), prepositions (l, k, b), determiners (Al), noun suffixes (yn, An, wn), verb suffixes (wA, wn) and pronouns (h, hA, hm). A complete list of prefixes and suffixes can be gleaned from the implementation. We treated the problem as a sequence labeling, which we performed using the CRF++ conditional random fields (CRF) sequence labeler (Lafferty et al., 2001). The features that we used included:

- the previous and next 2 characters relative to the current character. We tested a larger character window, but improvements were marginal.
- the word unigram probability of the combination of characters from the beginning of the word to the current character
- the word unigram probability of the combination of characters from the current character to the end of the word
- position of character from beginning and end of word

When we tested the segmenter on the retained PATB test set. We obtained an accuracy of 98.6% (at word level).

3.3. Detecting Stem Templates

As stated earlier, stem-templates are used to produce stems from root. For example, applying the “CCAC” stem template to the Arabic triliteral root “ktb” would generate the stem “ktAb” (book). Using the template CACC on “ktb” would generate the stem “kAtb” (writer). Stem-templates and patterns can help detect the POS, gender, and/or number, often deterministically. For example:

- Certain templates indicate precise POS, such as the case of “CCA}C” almost alway produces a plural noun such “bdA}l” (choices).
- Some templates strongly indicate gender, like the template “>CCwCp”, which almost exclusively produces a feminine singular noun such as “>n$wdp” (song or chant).

Such examples highlight the importance of knowing the template of a given word. Mubarak et al. (2009) reported that a manually POS tagged corpus of 7M words contained 347 stem templates. The number of noun and verb templates for triliteral roots were 227, such as "<CtCAC", and 69, such as "AstCCC", respectively. The number of noun and verb templates for four-letter roots were 40, such as "CCACC", and 11, such as "ytCCCC", respectively. If diacritics are removed from templates, the number of unique templates drops to 187.

To detect stem templates, we analyzed stems (after removing prefixes and suffixes from words using the aforementioned word segmenter) using Sebawai (Darwish, 2002). Sebawai is able to find roots of stems and their stem-templates. However, the stem templates that were produced by Sebawai were often linguistically incorrect in which all letters preceding the first letter in the root and all letters trailing the last letter in the root were removed from the template. For example, for the word "mktwb" (written), which has the stem template "mCCwC", Sebawai produced the template "CCwC". We reimplemented the root and stem template detection to overcome this problem and to integrate it into our system. To obtain stem templates, we used the stem-template and root inventory in Sebawai, which numbered 613 and 10,405 respectively. We ran Sebawai on the stems in the PATB corpus to generate stem templates and compute the maximum likelihood estimate that a stem template would be observed. We excluded Sebawai stem templates that were not observed, leaving only 91 stem templates. We also retained the root probabilities supplied by Sebawai that were estimated by running Sebawai on a large Arabic corpus. Stem template identification proceeded as follows:

```
Require: Root set $R$, stem template set $T$, stem $s$
  Declare Array of candidate templates $CT$
  if $s.length()$ = 2 then
    double second letter in $s$
    if $s \in R$ then
      $CT.add(s)$;
    end if
  else if $s.length()$ = 3 and $s \in R$ then
    $CT.add(s)$;
  else
    get $\forall t_i \in T$, where $t_i.length() = s.length()$;
    for all $t_i$ do
      if $\forall$ non-root letters $l_j \in t_i = \forall$ letters $\in S$ in same position then
        remove $\forall$ $lj$ from $s$
        if $s \in R$ then
          $CT.add(s)$
        else if $s.length() = 4$ & $s[2] = s[3]$ then
          remove $s[3]$
          if $s \in R$ then
            $CT.add(s)$
          end if
        end if
      end if
    end for
  end if
  return CT;
```

For testing, we processed 7,282 stems using our template extractor and then we manually examined the output of the system. The output was as follows:

Total number	**7,282**	
Correct templates	5,620	77.2%
Incorrect templates	840	11.5%
No template produced	822	11.3%

Upon examining the errors, we found that problems were generally due to: missing template; stems derived from so-called hollow roots, which contain vowels; stem derived from four letter roots; and stems derived from three letters roots where the second and third letters are identical.

3.4. POS tagging

For POS tagging, we simplified the PATB tag set to the following tags: ABBREV (abbreviation), ADJ (adjective), ADV (adverb), CASE (only in the case of alef inserted because of tanween with fatha), CONJ (conjunction), DET (determiner), FOREIGN (includes non-MSA words), FUT_PART ("s" suffix and "swf" particle indicating future), JUS ("|" for jussification attached to verbs), NOUN, NSUFF (noun suffix), NUM (number), PART (particles), PREP (preposition), PRON (pronoun), PUNC (punctuation), V (verb), VSUFF (verb suffix).

To train the POS sequence labeler, we used the following features:

- **List Match:** Whether the token matches one the following:
  - A gazetteer of spelled out numbers or digits. The gazetteer, which we manually constructed, contains 74 primitives that include single digits, tens, hundreds, thousands, tens of thousands, hundreds of thousands, millions, and billions in feminine and masculine forms as well as single and dual forms. An Arabic number would be constructed using a combination of these primitive.
  - A sequence of Arabic letters
  - A sequence of non-Arabic letters
  - A list of punctuations
- **Template:** The aforementioned stem template. Stem templates can helpful in identifying the POS of words. For example, the template '>CCAC' almost always produces words, such as '>wSAf' (descriptions), that have NOUN as their POS tag.
- **Prefixes:** The prefixes of the current word and of the previous word. This feature is applied to stems, while prefix and suffixes are given the value 'Y' for this feature. This helps capture agreement between word sequences. For example, if a word has the prefixes 'Al' (the) and the preceding word has the prefix 'Al' (the), then stem of the preceding word is likely a NOUN and the POS of the current stem part of the current word is likely a NOUN or ADJ.

- The position of the word in the sentence. This can help identify verbs that may appear in the first position in VSO sentences.

For sequence labeling, we trained a CRF sequence tagger using 90% of PATB parts 1, 2, and 3 on the segmented words. We kept 10% for testing. We used the CRF++ implementation[3] for this task. Table2 shows the results of POS using: the raw words only, raw words + list match, raw words + template, raw words + prefixes, and all the features. All results assume perfect segmentation. As the results show, though all features improved POS tagging accuracy, using stem templates yielded the most gain. The overall gain over the baseline system was 0.5%.

Run	Accuracy
Words	97.6%
Words + list match	97.9%
words + template	97.9%
words + prefixes	97.8%
All features	98.1%

Table 2: Results of using different features for POS tagging.

3.5. Determining Gender and Number Tags

All Arabic nouns and adjectives have gender (masculine or feminine) and number (singular, dual, or plural). Assigning proper gender and number tags to nouns and adjectives can be beneficial to a number of NLP applications such as parsing, where an adjective needs to match the noun it modifies in number and gender, and the verb has to match the number and gender of its subject. Table 3 provides gender and number tags for the previously analyzed tokens.

Word	G/N Tagging
تبين	تبين/V
للعلماء	للعلماء/PREP+DET+NOUN-MP
أن	أن/PART
عشرين	عشرين/NUM-MD+NSUFF
دقيقة	دقيقه/NOUN-FS+NSUFF
من	من/PREP
الرياضة	الرياضه/DET+NOUN-FS+NSUFF
في	في/PREP
اليوم	اليوم/DET+NOUN-MS
تساعد	تساعد/V
على	على/PREP
إبعاد	إبعاد/NOUN-MS
الإنفلونزا	الانفلونزا/DET+NOUN-MS
بنسبة	بنسبه/PREP+NOUN-FS+NSUFF
تقارب	تقارب/NOUN-MS
10%	10/NUM-MS+%/PUNC
،	،/PUNC

Table 3: system output with G/N tagging.

Some challenging cases that we considered for gender and number classification were as follows:

- The pervasive usage of broken plurals where noun suffixes are not used to indicate number. For example, the plural of ‘ktAb’ (book) is ‘ktb’ (books).

- Some nouns are referred to as masculine and feminine. For example, the word ‘r>s’ (head) can accept the demonstrative article ‘h*A’ (this – masculine) and ‘h*h’ (this – feminine).

- Masculine nouns may end by “At” or “p”, which are typically feminine noun suffixes. Consider the words “mmAt” (death)and “dAEyp” (preacher).

- The noun suffix “yn” may ambiguously indicate plural and dual forms of a noun as in “mslmyn” (Muslims/Two Muslims).

- Two words may share the same letters and diacritics but have different gender and/or number. Consider the words “HuDuwr” which could mean attendees or attendance.

- Some polysemous words might have the same gender and number in singular form, but their plurals may have different genders. Consider the word ‘EAml’ (worker/-factor) which has the broken plurals ‘EmAl’ (workers – masculine) and ‘EwAml’ (factors – feminine).

- Some regular nouns are commonly used as proper nouns, where the gender and number would be different. For example, the word ‘<ymAn’ (faith) is typically masculine, but it is a common female name (Iman).

- Words in the category of “plural of plural” (jmE AljmE). For example, the word ‘jndy’ (soldier) has the plural ‘jnd’, which in turn has the plurals ‘>jnAd’ and ‘jnwd’. The plural “jnd” is a masculine plural, while the plural of the plural can be musculine or feminine.

- Words in the category of “name of plural” (Asm AljmE) where some words behave like singular or plural nouns though they are plural. For example, the word “$Eb” (people) behaves like a singular masculine noun, while the word “qwm” (people) behaves like a plural noun.

- Comparative adjective (‘>fEl AltfDyl’) can be used with nouns of any gender and number. Consider the adjective ‘>kbr’ (larger).

- There are rules for determining gender for numbers written in letters, however some numbers can be masculine and feminine, and the same case applies for quantifiers (words that indicate quantities or parts). Numbers generally disagree in gender with their quantifiers.

- Some plurals don’t have a singulars derived from the same root. Consider the “>$lA’’ (debris).

- Certain nouns are masculine in the singular form but feminine in their broken plural form or vice versa. For example, the word “$rT” (condition) is masculine, while its plural form “$rwT” is feminine. In the other direction, “nmlp” (ant) is feminine, and the plural form “nml” is masculine. The latter case is called (name of a kind) ‘Asm Aljns’.

[3] https://code.google.com/p/crfpp/

For classification of such cases, we used the random forest classifier implementation in Weka(Breiman, 2001). The parameterized the random forest classifier to generate 10 trees, with 5 attributes for each tree with unlimited depth. We manually tagged 8,400 randomly selected unique nouns and adjectives from PATB for gender and number. It is noteworthy that the currently available PATB does not have gender and number information. However, such tags will be available in future releases of the PATB. When tagging for gender and number, if a surface form is polysemous with multiple senses, then: the most popular form of the word (or sense) is assumed if the alternative is obscure; else multiple gender and number tags are assigned to the surface form if the different words (or senses) are common.
We used the following features for training:

- Stem template
- Length of the stem template
- POS tag
- Attached suffix
- Whether the word ends with a feminine marker ("At" or "p")
- Tags that were obtained from a large word list that was extracted from the "the Modern Arabic Language Dictionary".[4] The dictionary was parsed automatically producing 28,383 entries with associated tags of "feminine", "masculine", "dual", "plural", "singular", and "particle". Such a list is particularly helpful in determining the gender of broken plurals.
- Bigram language model probability that the word is preceded by one the following demonstrative articles: h*A, h*h, h*An, hAtAn, h*yn, hAtyn, h&lA'. The language model was trained on a dump of Arabic Wikipedia[5] and 10 years worth of Aljazeera.net[6] news articles.
- Whether the word appears in a gazetteer of proper nouns that have associated gender tags. The list was obtained using two methods, namely:
  - We obtained a list of high school graduates from Palestine. The list published on the web provides along with the full name of the students their gender. In the case of male students, we assumed that their first names and those of their parents and grandparents were masculine. As for female students, we assumed the first name is feminine and the names of the parents and grandparents to be masculine. Using this method, we obtained 4,649 unique names.
  - Given all the Arabic Wikipedia articles that have English equivalents and that belong to the Wikipedia categories containing the words 'person', 'birth', and 'death' indicating that the title is a person's name, we automatically tagged English Wikipedia articles with gender information. To do so, we simply counted the words 'he', 'his', 'she', 'her' in the first 100 words in the Wikipedia article. If the masculine pronouns out numbered feminine ones more than 2:1, then the title of the article is considered to point to a male person. Similarly, if feminine pronouns out numbered masculine pronouns more than 2:1, the name is considered feminine. The tags were propagated to the Arabic Wikipedia titles. Using this method we obtained more than 6,071 unique names. The advantage of using Wikipedia is that we were able to capture many non-Arabic names for which our classifier would not work.

To test classification effectiveness, we used 20-fold cross validation where 19 folds were used for training and the remaining fold for testing. Since we picked unique nouns from PATB, all classified nouns and adjectives in the test folds were previously unseen during training. For classification, the average accuracy for the folds for gender and number classification were 95.6% and 94.9% respectively. We achieved much higher results for both gender and number tagging compared to the work of (Alkuhlani and Habash, 2001). In fact, our results for the gender tag are very close to their results for previously seen test cases.

4. Conclusion

We presented in this paper a complete end-to-end automatic processing system for Arabic. The system performs corrections of common spelling errors, word segmentation, POS tagging, and gender/number classification. The system is built on a large set of resources. Even though, the system shown to produce good results; tuning and expanding the training resources used by the system would improve it further. The system is provided as an open source package for the community to accelerate research in the area or Arabic NLP. We intend to extend the system and incorporate other modules such as Named-Entity Recognizer (NER) as well as packaging the system to seamlessly be deployed within other pipeline and processing tools. Also, plan to look at ways to improve stem-template detection to lower template identification error rates.

5. References

Sarah Alkuhlani and Nizar Habash. 2012. Identifying broken plurals, irregular gender, and rationality in Arabic text. Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics.

Leo Breiman. 2001. Random Forests. Machine Learning. 45(1):5-32.

Kareem Darwish. 2002. Building a Shallow Morphological Analyzer in One Day. ACL Workshop on Computational Approaches to Semitic Languages. 2002.

Mona T. Diab. 2009. Second Generation Tools (AMIRA 2.0): Fast and Robust Tok- enization, POS tagging, and Base Phrase Chunking. 2nd Int. Conf. on Arabic Language Resources and Tools, 2009.

[4] http://www.sh.rewayat2.com/gharib/Web/31852/

[5] http://ar.wikipedia.org

[6] http://aljazeera.net

Nizar Habash, Owen Rambow and Ryan Roth. 2009. MADA+TOKAN: A Toolkit for Arabic Tokenization, Diacritization, Morphological Disambiguation, POS Tagging, Stemming and Lemmatization. In Proceedings of the 2nd International Conference on Arabic Language Resources and Tools (MEDAR), Cairo, Egypt, 2009

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. Proceedings of the 18th International Conference on Machine Learning 2001 (ICML 2001), pp. 282-289.

M Moussa, MW Fakhr, Kareem Darwish. 2012. Statistical Denormalization for Arabic Text. Empirical Methods in Natural Language Processing, 228.

H. Mubarak, K. Shaban, and F. Adel. 2009. Lexical and Morphological Statistics of an Arabic POS-Tagged Corpus. In Proceedings of the 9th Conference on Language Engineering ESOLEC2009, 23-24 December 2009, Cairo, Egypt, pp. 147-161.

Y. Lee, K. Papineni, S. Roukos, O. Emam, H. Hassan. 2003. Language Model. Based Arabic Word Segmentation. In the Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, July 2003, Sapporo, Japan. p. 399 - 406.

Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In Proceedings of HLT-NAACL 2003, pp. 252-259.