Adapting Standard Open-Source Resources To Tagging A Morphologically Rich Language: A Case Study With Arabic

Hajder S. Rabiee
Royal Institute of Technology
hajder@kth.se

Abstract

In this paper we investigate the possibility of creating a PoS tagger for Modern Standard Arabic by integrating open-source tools. In particular a morphological analyser, used in the disambiguation process with a PoS tagger trained on classical Arabic. The investigation shows the scarcity of open-source tools and resources, which complicated the integration process. Among the problems are different input/output formats of each tool, granularity of tag sets and different tokenisation schemes.

The final prototype of the PoS tagger was trained on classical Arabic and tested on a sample text of modern standard Arabic. The results are not that impressive, only an accuracy of 73% is achieved. This paper however outlines the difficulties of integrating tools today and proposes ideas for future work in the field and shows that classical Arabic is not sufficient as training data for an Arabic tagger.

1 INTRODUCTION

It is estimated that about 220 million people are Arab speaking(Lewis, 2009) and that Arabic is the fourth most spoken language, thus it's a major international modern language. It is also recognised as one of the six major official languages of the United Nations. English on the other hand with 330 million speakers(Lewis, 2009), has received an unproportional attention when it comes to the development of open-source NLP tools and resources. The tools for Arabic are few and often miss certain features or do not live up to the same standard as their English counterpart (Atwell et al., 2004). The possible reasons for this are the non-Roman script and Arabic being a morphologically complex language.

The difficulties in integrating existing tools lie in the way each tool represents the texts. The morphological analysers use different encodings, e.g. CP-1256, UTF-8, ISO-8859-6 or different alphabets, e.g. transliteration scheme (Buckwalter) or the actual Arabic alphabet. The tokenisation algorithms are also different for each tool, leading to a different analysis granularity, hence a different tag set. As this is a basis for evaluation, the problem of evaluating tools on a common ground arises too. One of the fundamental parts of any linguistic application is the Part-of-Speech tagger (PoS tagger) which in turn is dependent on a morphological analyser which utilises dictionaries for lookup.

In this paper we investigate what open-source tools exist today for Arabic NLP, especially PoS taggers and morphological analysers. We compare them with regards to several aspects e.g. how easy it is to get hold of, which algorithm/model is used, how difficult it is to adapt into other tools, for which purpose it's suitable etc. For the purpose of building a prototype of a PoS tagger for Modern Standard Arabic (MSA), based on a Classical/Quranic Arabic (CA/QA) model. The problem is interesting because CA lacks many new (modern) words, e.g. *TV*; *computer*; *car*. QA has slightly different grammatical constructions than MSA. Moreover, in Arabic case endings are denoted by short vowels, these are usally omitted in written MSA; in contrast to QA which is fully diacritized.

2 BACKGROUND

In (Atwell et al., 2004) an outline of some of the most important tools is presented. Furthermore (Al-Sughaiyer and Al-Kharashi, 2004) report in their survey findings that many tools are only described generally with no measures of effectiveness and provide little in-depth investigation of available techniques. They also claim many researchers don't acknowledge the efforts of other and no systematic approach of evaluating algorithms exist either. Additionally the lack of standards is something criticised.

2.1 MORPHOLOGICAL ANALYSERS

Buckwalter Morphological Analyzer The Buckwalter Morphological Analyzer (BAMA) 1.0 (Buckwalter, 2002) was released in 2002, it can be obtained by sending an inquiry to LDC. There's

Proceedings of the Student Research Workshop associated with RANLP 2011, pages 127–132,
Hissar, Bulgaria, 13 September 2011.

also a Java port versioned 1.2 written by Pierrick Brihaye available online called *Aramorph*. The first version of BAMA has several shortcomings, as witnessed by (Altabba et al., 2010). The fact that all derivations are hard coded instead of relying on rules makes the runtime processing long. Furthermore, they state that it has a spelling problem where it converts between Arabic letters Aleph and Hamza. Problems exist with words like Hadramout

حَضرَمَوت

and problems when dealing with acts in the past tense and the pronoun is absent or past tense passive voice, e.g.

حَاول، اَضرب

Many of the shortcoming mentioned by (Altabba et al., 2010) can probably be remedied if the lexical files would not apply a coarse representations of the affixes; collecting clitics together with prefixes or suffixes is not the best way. As argued by (Sawalha and Atwell, 2010) a more fine-grained representation of words in general is needed to account for the complexities of the Arabic language. The latest version, BAMA 2.0 and Standard Morphological Analyzer 3.1 (SAMA), which is based on BAMA 2.0, is only available through LDC membership though. Thus it was not possible for us to experiment with it.

Alkhalil The Alkhalil Morphological Analyzer is written in Java, the lexical resources consist of several classes, each representing a type of the same nature and morphological features. Analysis is carried out in the following steps: preprocessing, removal of diacritics; segmentation, each word is considered as (proclitic+stem+enclitic) too (Boudlal et al., 2011). According to (Altabba et al., 2010) the Alkhalil analyzer is the best one, although it has some problems with its database. It won the first prize at a competition by The Arab League Educational, Cultural Scientific Organization (ALESCO) in 2010. It has some limitations such as it does not provide PoS tags in good reusable format, e.g only in Arabic. Neither does it differentiate between clitics and affixes fully, it detects proclitics and enclitics but they are referred to either as prefix or suffix.

2.2 PART-OF-SPEECH TAGGERS

Stanford PoS tagger is originally developed for English at Stanford University and is described in (Toutanova and Manning, 2000). The tagger is based on the maximum-entropy model. The improved version, which is described in (Toutanova et al., 2003) adds support for other languages together with speed and usability improvements.

The latest version comes with trained models for Chinese, German and Arabic, it claims a 96.42% accuracy on Arabic. The tagger was trained on the training part of the Arabic Penn Treebank (ATB). It uses augmented Bies mapping of ATB tags(Bies, 2003). Which is not so fine-grain, as the authors also confirm, for example it does not tokenize clitics when tagging, e.g. the word

بسم

is tagged as noun, while it should be separated into the proclitic and noun as

ب + سم

tagging it as *preposition* and *noun* respectively. This smaller tag set makes it harder to assign a ”wrong” tag, and probably one factor contributing to the high accuracy.

BrillTagger(Brill, 1995) combines the ideas of rule-based tagging with a general machine-learning approach which is *transformation-based*. The idea behind is to initially let the text pass through a annotator, in part-of-speech context this might be assigning each word its most likely tag. Then the text is compared to the gold standard, in order to create *transformations* that can be applied to improve the initial text as much as possible.

a rewrite rule - e.g. *change the word from modal to noun*

a triggering environment - e.g. *preceding word is a determiner*

TreeTagger is another language-independent tagger by (Schmid, 1994) and is based on decision trees. The tagger successfully tags many European languages, and it is adaptable to other languages if a lexicon and a manually tagged training corpus are available.

2.3 EVALUATION METHODS

Several methods for evaluating a tagger exist, among the most common are precision, recall and accuracy/success rate.

For a better understanding of how well a tagger performs, one can use tag-wise evaluation. Tag-wise measurement is a good way of evaluating a tagger, because by measuring one tag at a time one can get a better picture of what tags are harder to distinguish than others. The error measures are *precision* and *recall*. Precision is the fraction of tokens tagged T in the gold standard of those tagged T by the tagger. Recall is the fraction of tokens tagged T by the tagger of those tagged T in the gold standard.

2.4 OTHER RESOURCES

If we come to look at the situation of corpora or stemmers, the situation is similar (Al-Sughaiyer and Al-Kharashi, 2004), or even worse in the case of corpora. Not a single tagged MSA corpus exists freely or publicly. The only exception is Shereen Khoja who distributes her 50000 word tagged corpus for research purposes(Khoja, 2001). For our project, we were not able to obtain a copy.

3 METHOD

The first tools selected were the Alkhalil morphological analyser and the Stanford PoS tagger. The first one was selected because of its availability, portability and good support from the authors. The Stanford PoS tagger additionally seemed good as it belongs to a renowned NLP group and as the authors claim performs very well on Arabic. Furthermore it is written in the same language as the morphological analyser (Java), anticipating assembling the two would make it easy to create a prototype of a tagger.

The main aim of the PoS tagger is to see how well a tagger can perform on MSA text when trained on CA, i.e. tagging texts from a different lexicon than the tagger was trained on. We were further motivated by (Habash and Rambow, 2005) who reported positive results on using a morphological analyser during the tagging process, their work is based on (Hajič, 2000) who argues that a morphological analyser aids the morphological disambiguation process during tagging.

3.1 TRAINING CORPUS

The only corpus freely available to us was the Quranic Arabic Corpus (Dukes, 2009) for retraining the tagger. The corpus has 77430 words each annotated with tag, prefix, lemma and is fully diacritized. Only whitespace tokenisation was used, this has the drawback of the tagging not being very fine-grain. As Arabic is a highly inflectional language and many words have affixes that are discarded in the analysis. For the purpose of this investigation though, whose main goal is to tag MSA with a CA model, the decision was justified.

3.2 BUILDING A PROTOTYPE

The kind of flow we had in mind is illustrated in Figure 1. During the process it was discovered that the tagger didn't have a solution to tagging unknown words for a language, i.e. words that were not encountered during training. The tagger "only" develops rules from the training corpus and defines so called *extractors* internally that recognise morphological features, these are sufficient for English, but certainly not for a morphologically complex language as Arabic. The tagger also lacked a way of integrating a morphological analyser into it. There does not exist a way of getting a particular tag's confidence or any other useful measure.

In order to continue the investigation and build a prototype the Stanford tagger had to be abandoned. Instead the TreeTagger was selected, it allowed for the usage of the MA by constraining a word's possible tags in the text file. Thereby overriding the lexical information in the tagger parameter file, see Table 2 for an illustration of an input text file to be tagged. The Alkhalil analyser was abandoned at this stage too. Instead the BAMA 1.2 was chosen because it outputs the POS tags in English and not as Alkhalil, which outputs them only in Arabic. The Table 1 contains the exact mapping that is performed between the output from the MA to the Quranic corpus' tagset. The ABBREV and INTERJ from the MA, does not have any equivalent in the Quran corpus tag set, we mapped them to the common tag N (noun). A minor mapping issue occured with the tag ADV (adverb). From the MA it was ambiguous due to the fact that the Quran Corpus tag set actually distinguishes between T (time adverb) and LOC (location adverb), the output from the MA does not produce such a separation of the adverb. Therefore we mapped all ADV to T, which was the most common tag in the training corpus (T=1115 vs LOC=656 times). All morphological features were removed, e.g. N_3PERSON_PL, N_2PERSON_SG and collapsed to N. They both contribute to the count of the "N"-tag. This made the decision of choosing the most likely tag from the MA easier.

4 EVALUATION

The tagger was trained on Quranic Arabic (QA) which is both a smaller set than Modern Standard Arabic (MSA) and contains some more complex

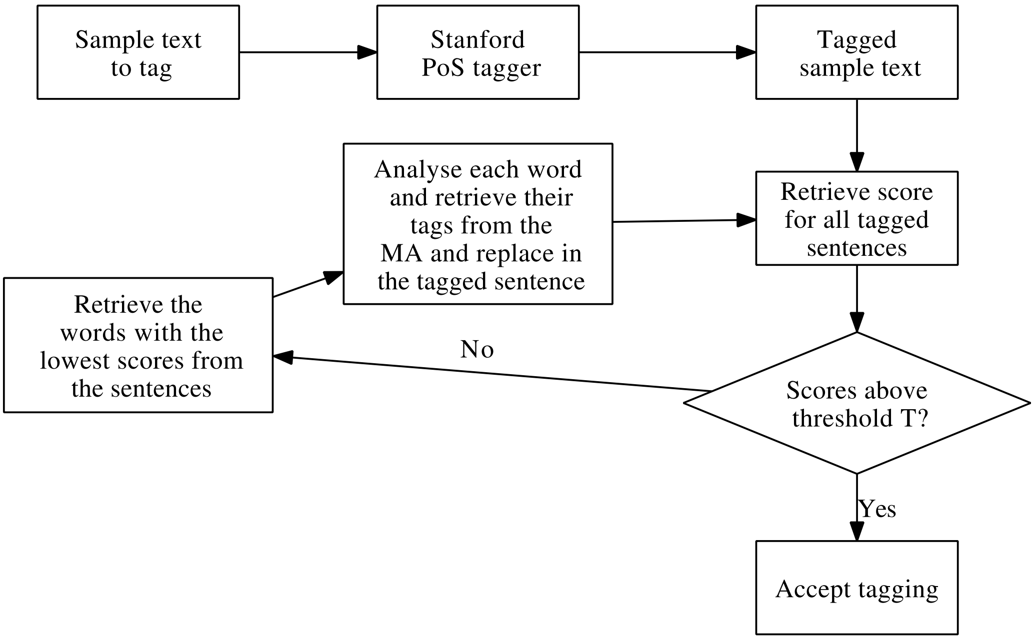

Figure 1: Initial thought of the integration between the MA and the PoS tagger

MA output	Quran Corpus equivalent
NOUN	N
N_PROP	PN
VERB.*[1]	V
PREP	P
REL_PRON	REL
ADV	T
INTERROG_PART	INTG
NEG_PART	NEG
EMPHATIC_PARTICLE	EMPH
INTERJ	N
ABBREV	N

Table 1: The mapping from BAMA's tag set to the Quran Corpus' tag set

WORD1	TAG1
WORD2	TAG1 TAG2 TAG3
WORD3	TAG1 TAG2
WORD4	TAG2 TAG3 TAG5
etc	...

Table 2: Sample input text with tag constraints of one tag

morphological and syntactic constructs, these are however much less in comparison to the words available in MSA, which includes *modern* words e.g. TV, mobile phone etc. From this perspective it would be interesting to see how the tagger - trained on QA - would perform on MSA together with the morphological analyser. The accuracy results from the initial tagging experiments are shown in Table 4. For the MSA sample text we chose an extract of an article from the Arabic BBC newspage[2] containing 66 words, they were manually annotated by an Arab speaker, and considered the "gold standard" during the evaluation. The tag set used is a very simple subset extracted from the training corpus (Quran corpus) and is described in (Dukes, 2009).

The tagger allowed for specifying the open class set and from the Quran Corpus those presented in Table 3 were extracted. *Baseline* was simply tagging each word as N (noun).

When more than one tag is appended to the sample text file, the tagger will be involved in making decisions between the different tags. If only one tag is chosen and input to the tagger, the tag's probability is implicitly 1; it is only the output from the MA that is considered. We experimented with both settings. Another configuration for our experiments was adding a probability to the tags, as well as setting an option to output maximum

[2] `http://www.bbc.co.uk/arabic`

Tag	Description
N	Noun
PN	Proper Noun
ADJ	Adjective
T	Time adverb
LOC	Location adverb
V	Verb
IMPN	Imperative Verbal Noun

Table 3: The open tag class

Experiment	Accuracy
Baseline on MSA	44%
Baseline on QA	36%
Stanford on QA	98%
TreeTagger on QA	96%
Stanford on MSA	39%
TreeTagger on MSA	35%
BAMA on MSA	69%

Table 4: Initial experiments accuracy

Tag	Precision	Recall	F-Measure	Accuracy
N	76%	89%	82%	73%
PRON	100%	25%	40%	
ADJ	0	0	0	
LOC	-	-	-	
T	-	-	-	
V	82%	60%	69%	
P	79%	100%	88%	
IMPN	-	-	-	

Table 5: MA tagging and tagger experiment with three appended tags on MSA text, no probabilities.

Tag	Precision	Recall	F-Measure	Accuracy
N	75%	86%	80%	73%
PRON	100%	25%	40%	
ADJ	0	0	0	
LOC	-	-	-	
T	-	-	-	
V	91%	67%	77%	
P	85%	100%	92%	
IMPN	-	-	-	

Table 6: MA tagging and tagger experiment on MSA text, three appended tags with frequency probability distribution

three tags to the appended file.

5 CONCLUSIONS

Using a training corpus with different characteristics than the text to tag, yielded expected results: very low. The results on the QA training text, were also expected: high. The *baseline* was tagging all words as a noun. It is interesting that both the Stanford tagger and the TreeTagger had a lower accuracy on MSA than the baseline. Changing parameters and settings for the appended tags leads to a slight improvement, see Table 6, which was the experiment with the highest accuracy and best values on the tags' error measures. The other experiment with no probability associated, in Table 5 also scored high. The accuracy remains the same as when choosing the frequency probability, see the results from Table 6. There's only a slight exchange of the error measures between the two. In general though, an accuracy of 70% is probably not good enough for many applications. It can be argued that a text with more words could have been used for tagging. Howevery, open-source tagged texts for gold standard, is a rare resource in Arabic NLP. Tagging a text manually is a time-consuming task and was not suitable for this case study. A high account of the accuracy is due to the morphological analysis, we see in Table 4 that the MA only achieves a 69% accuracy. While the usage of TreeTagger increases it to roughly 73%. By this we can draw the conclusion that the tagger contributes very little to the overall accuracy.

6 FUTURE WORK

First improvement is trying to experiment with a more fine-grain tag set. That would involve some more sophisticated methods on choosing the best solution from the MA, one way is to assign some sort of score to a solution that aids in the decision. This would open up for example building tools to adjust tagging granularity, depending on end application. The number of tagged corpora needs to increase. Our idea is to build on the work of (Sawalha and Atwell, 2010) and try to develop a corpora tagged with that new tag set.

Many resources are presented in (Nizar Habash, 2010), however many of those tools are licensed and/or not available publicly. This is a real impediment for those that wish to to take their steps into the area. Attracting new researchers requires having tools at hand easily. It is necessary if we wish to see more and better results. Finally, we believe it is only a matter of time until see more and better applications are being built for Arabic NLP.

ACKNOWLEDGEMENTS

I would like to thank my supervisor Dr. Eric Atwell. This work would has also been supported by my supervisor Viggo Kann from the Royal Institute of Technology.

References

[Al-Sughaiyer and Al-Kharashi2004] I. Al-Sughaiyer and A. Al-Kharashi. 2004. Arabic morphological analysis techniques: A comprehensive survey. *Journal of the American Society for Information Science and Technology*, 55:189–213.

[Altabba et al.2010] M. Altabba, A Al-Zaraee, and M A Shukairy. 2010. An Arabic morphological analyzer and part-of-speech tagger. Master's thesis, Arab International University, Damascus, Syria.

[Atwell et al.2004] E. Atwell, L. Al-Sulaiti, S. Al-Osaimi, and B. Abu Shawar. 2004. A review of Arabic corpus analysis tools. In *Bel, B and Marlien, I (editors) Proceedings of TALN04: XI Conference sur le Traitement Automatique des Langues Naturelles*, volume 2, pages 229–234.

[Bies2003] Ann Bies. 2003. http://www.ircs.upenn.edu/arabic/Jan03release/arabic-POStags-collapse-to-PennPOStags.txt.

[Boudlal et al.2011] A. Boudlal, A. Lakhouaja, A. Mazroui, A. Meziane, and M. Bebah. 2011. Alkhalil morpho sys: A morphosyntactic analysis system for Arabic texts. azze.mazroui@gmail.com.

[Brill1995] E. Brill. 1995. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Lingustics*, 21(4):543–566.

[Buckwalter2002] T. Buckwalter. 2002. *Buckwalter Arabic Morphological Analyzer 1.0*. Linguistic Data Consortium.

[Dukes2009] K. Dukes. 2009. The Quranic Arabic Corpus. http://corpus.quran.com/.

[Habash and Rambow2005] N. Habash and O. 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*. ACL.

[Hajič2000] J. Hajič. 2000. Morphological tagging: data vs dictionaries. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*.

[Khoja2001] S. Khoja. 2001. Shereen khoja. http://zeus.cs.pacificu.edu/shereen/research.htm#corpora. Accessed 2011-08-30.

[Lewis2009] M. Paul (ed.) Lewis. 2009. *Ethnologue: Languages of the World*. SIL International, Dallas, Texas, sixth, edition. Online version: http://www.ethnologue.com/.

[Nizar Habash2010] Y. Nizar Habash. 2010. *Introduction to Arabic natural language processing*. Morgan and Claypool.

[Sawalha and Atwell2010] M. Sawalha and E. Atwell. 2010. Fine-grain moprhological analyzer and part-of-speech tagger for Arabic text. In *Language Resources and Evaluation Conference*.

[Schmid1994] H. Schmid. 1994. Probabilistic part-of-speech tagging using decisions trees. In *International Conference on New Methods in Language Processing*.

[Toutanova and Manning2000] K. Toutanova and C D. Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000)*, pages 63–70.

[Toutanova et al.2003] K. Toutanova, D. Klein, C. Manning, and Y. Singer. 2003. Feature-rich part-of-speech tagging with cyclic dependency network. In *Proceedings of HLT-NAACL*, pages 252–259.