Ensemble Morphosyntactic Analyser for Classical Arabic

Abdulrahman Alosaimy and Eric Atwell

April 7, 2016

Abstract In Modern Standard Arabic text (MSA), there are at least seven available morphological analysers (MA). Several Part-of-Speech (POS) taggers use these MAs to improve accuracy. However, the choice between these analysers is challenging, and there is none designed for Classical Arabic. Several morphological analysers have been studied and combined to be evaluated on a common ground. The goal of our language resource is to build a freely accessible multi-component toolkit (named SAWAREF[1]) for part-of-speech tagging and morphological analysers that can provide a comparative evaluation, standardise the outputs of each component, combine different solutions, and analyse and vote for the best candidates. We illustrate the use of SAWAREF in tagging adjectives and shows how accuracy of tagging adjectives is still very low. This paper describes the research method and design, and discusses the key issues and obstacles.

1 Introduction

The Arabic language has several variants where each has its own characteristics in morphology, lexicon and syntax. Classical Arabic, Modern Standard Arabic (MSA) and Dialectal Arabic have been written on different genres and media: from social networks to newspapers to journals. Researchers tends to build POS taggers for specific variant or dialects. While several POS taggers exist, many of them are incompatible: incompatible tokenization and diverse tagsets. The ultimate goal of our system is to build a methodology of combining POS taggers; hence, a more robust tagger.

In Machine Learning, ensemble methods refer to the process of combining multiple learning methods to obtain a higher accuracy in classification predication that was not achieved by any of individual learning methods. Multiple types of ensemble exist in the literature, including: bagging (equally-weighted models trained on random subsets of the training data), and boosting (adaptive training where each new model focus on subset of training data that were misclassified). Many other combination techniques are available.

[1] SAWAREF is freely accessible through the following link: sawaref.al-osaimy.com.

In POS tagging, different techniques were used, including knowledge based models: (table lookup, syllable-based morphology, pattern morphology) and empirical methods: (Hidden Markov Models (HMM), Support Vector Machines (SVM), …). Each POS tagger is designed differently; however, without a full understanding of the language, no POS tagger could ensure perfect accuracy. Because of their different bases and contains different knowledge, taggers will typically produce different errors [1]. A combination of POS taggers exploit these differences, and it is reported to achieve a better accuracy for several languages, including English [1–3], Italian [4], Icelandic [5], Polish [6, 7], Telugu [8], and Swedish [9] and even for Arabic [10].

Most of the combination of POS-taggers are based on training different models on "one" training corpus. Therefore, each model uses the same tagset and morphological segmentation as the one on the training corpus. However, combining black-box taggers involves handling different issues, such as unifying taggers' tagsets into one standard tagset. In addition, the output of those taggers need to be aligned on the different levels: document, sentence, word, and morpheme.

One of the earliest attempts is The Automatic Mapping Among Lexico-Grammatical Annotation Models (AMALGAM)[11, 12] project which aims to provide "POS-tagset conversion" method for English annotation schemas; i.e. given a text tagged with one tagset, it outputs the text tagged with another tagset, no matter how the two tagsets differ in their formalism, size, … etc. The AMALGAM project mapped the tagset A to tagset B by the following steps: First, it builds a POS-tagger trained on the corpus tagged with tagset A. Next, it uses the tagger to predict the tag of the word in a corpus tagged with tagset B. In another words, there is no mapping rules from tagset A to tagset B. This decision was made as the authors discovered in earlier experiments that the n-to-m and 1-to-n mapping "predominated" over the simple 1-to-1 and n-to-1 mappings.

[13] tried to adapt several taggers by training them on the QAC and then applying learned model on tagging a MSA sample. He used BAMA as a morphological analyser and used TreeTagger to train a model from the QAC. Tagging then was constrained by the solutions of BAMA. The tagset of BAMA was reduced to only 9-tag tagset that was comparable with QAC tagset. However, his mapping countered a one-to-may cases (e.g. mapping ADV tag). In that case, he chose to map to the most common tag. The accuracy achieved in tagging a 66-word MSA sample was 76This tagging can be seen as a novel sequential tagging scheme as it uses the output of BAMA to constrain TreeTagger. The coarse mapping of the tagset is justified as the author need to compare taggers with different tagsets. However, error was raised from this mapping: LOC is about 38% of ADV cases, and the mapping of ADV to the other more common tag T constrains the TreeTagger to an incorrect tag. Additionally, the author used a test sample with only 66 words, which does not count as a representative sample of the MSA. The sample's origin, genre, and how it was annotated were not even clear. The author used an earlier version of QAC which has word-based annotation, and thus the morphological alignment was not an issue.

Alabbas and Ramsay [14, 15] reported more promising results. They performed a simple method for combining three Arabic taggers: MADA, AMIRA and a simple house-made maximum likelihood tagger (MXL). They examined four strategies of combining the results: three stratigies of majority voting (with backoff to MADA, AMIRA or MXL), majority voting with backoff to the most confident, and most confident as the main strategy. To define the most confident, they first examined how likely a tagger is correct when tagging with X (e.g. noun), e.g. MADA is 95% correct when it is tagging as noun. Most-confident strategy achieved the highest accuracy with 0.995 with a coarse-grained tagset and 0.96 with a fine-grained. To recover from the mismatches between the reference corpus used (PATB Part 1 v. 3) and AMIRA tagset, the authors used transformation-based retagging (TBR) which improves the score from 90% to 95%. However, AMIRA and MADA tokenize sometimes differently. To solve this problem, the PATB were translated to a coarser version of AMIRA's tagset, and compared with AMIRA's output: if the output is compatible with the translated tag, it is used, otherwise, the translated tag is used. This ensured that AMIRA and MADA will use same token number as the PATB. The accuracy of this study is very encouraging. The combination of tagger boosts the accuracy by 2-4%. We would like to see how this combination work in different genre or maybe a different variant of Arabic: e.g. classical Arabic. In addition, the technique used for enforcing the same tokenization leads to a bias in the test set. If AMIRA's tag is not compatible with PATB, the TBR tagger will use a translated PATB tag which is always correct. In addition, this technique also can not be applied in tagging a raw text as it relies on a tagged corpus to enforce same tokenization.

2 Overview

Currently, SAWAREF system can run 7 morphological analysers, namely: AlKhalil (KH) [16], Buckwalter (BJ) [17], Elixir-FM (EX) [18], Microsoft ATKS Sarf (MS), ALMORGEANA (AL) [19], AraComLex (AR) [20], and Xerox (XE) [21]. In addition, it can run 7 POS taggers, namely: Madamira (MA) [22], MADA (MD) [23], AMIRA (AM)[24], Stanford POS tagger (ST) [25], Microsoft ATKS POS Tagger (MT), MarMoT (MR) [26], Wapiti Arabic Model (WP) [27]. It is a web-based system, which avoids all the hassle of installation and provides a simple interface for comparing between taggers and evaluating them. It is meant not to be compared with those taggers: instead it provides a range of useful tools to compare them against each other.

In Figure 1, we illustrate the overall process of the ensemble system. The process starts with the text to be tagged being sent to a pre-processing component for each participating tagger. The results are parsed and then sent to a tool that aligns the results by the word. Next, we use the mapping list to translate and inflate the solution set. The solution set is then rearranged and each solution aligns with the morphological segmentation of the word. Finally, we vote and rank the solution sets to produce the most confident tag.

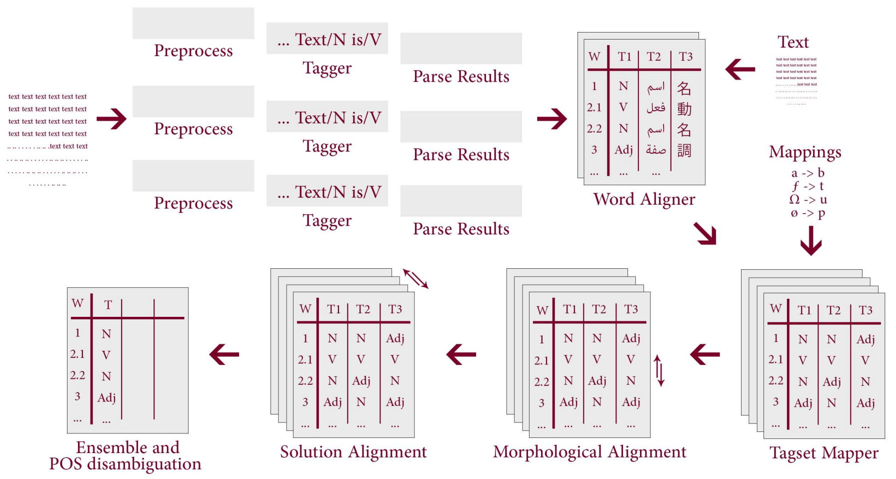

Figure 1: The overall process of the ensemble system: SAWAREF.

In the next sections, we will dive into each stage in the system. It is important that while the system is not yet thoroughly evaluated and tested, most of the stages are completed and can be useful for other applications: perform evaluation and testing of taggers, ease the choice of a tagger for specific research needs. Solution grouping and the final POS disambiguation stages are still under development.

3 Pre-processing

Most of the time, each component does the required pre-processing step on its own. That is, it transliterates, normalizes, spell corrects, and/or tokenizes the input text in the format suitable for the component's needs. However, after a series of tests to maximise the accuracy, we found that some poorly-documented components assume input in certain conditions. Some components work better when diacritics, digits, and/or punctuations are deleted, text is normalised, and/or text is transliterated. In general, we followed the documentation requirements, if such exist, and pre-processed the input the way it achieves maximal accuracy.

TOKENIZATION: Tokenization is well-known to be difficult in Arabic, because writers often omit word spaces next to non-joining letters. Tokenization on whitespace and punctuation therefore introduces lots of errors on all but the most carefully written texts. However, our system assumes that every tool has its own word and morpheme tokenization. We later make sure that tokenized streams of morphemes are aligned (See Section 6 and 7). MarMoT required the input to be tokenized and we used the AMIRA word tokenizer and deleted signs that indicate affix type. Stanford POS tagger needed to tokenize the text and we used its sibling: Stanford word segmenter[2]. AraComLex assumed the text to be tokenized: each word in a line. We used a simple *sed* command line tool that replaced punctuation marks and white spaces with a newline.

TRANSLITERATON: We transliterate the input if the tagger does not support the UTF-8 format (e.g. MarMoT and BAMA) using the two-way table-lookup transliteration system based on the Buckwalter convention.

4 Component Manipulation

Running Most of the tools are runnable through the command line. Some components have an API (e.g. Stanford Segmenter) that allows them to be integrated in developer's code. One component (Alkhalil) is only runnable through a Graphical User Interface (GUI), and for the purpose of integrating in the SAWAREF system, we added the functionality to permit the run from the command line without interfering with the analysis code.

Wrap-To-Service Since we plan to allow the usage of those tools from the web, we wrap each component in a service. The goal here is to speed up the processing of texts by having the morphology model loaded and ready for each subsequent requests. We were able to extend some components and build a web service that accepts HTTP requests and returns component output whilst maintaining dictionaries in memory.

Special Modifications In Alkhalil morphological analyser, if a word reappears in the text, it will be ignored and no analyses will be given. We edited Alkhalil to print the analyses of each word, allowing us to align the analyses with other components' results. In addition, the word's type and POS tag in Alkhalil are printed in free text and thus are not in a good reusable format. We edited the source code and changed the format of the output to be in JSON. TODO EXAMPLE HERE

5 Parsing Results and Extracting Morphological Features

Every component has its own format of output. We built several parsers that extract analysis for each tagger and transform them to a standard JSON object. The goal is to standardise the format so that they can be reused for evaluation and ensemble tagging purposes.

For each morpheme, we maintain the following output, if they exist:

Basic POS tag The tag after being post-processed.
Morphological Features Person, gender, number, aspect, voice, is emphasised.
Syntactical Features Mood and Case.
Morphological Segmentation How the word has been segmented.

[2] Stanford Word Segmenter process raw text input according to the Penn Arabic Treebank standard [28].

Word Analysis Root, Stem and Lemma.

Since we standardized the outcomes of each tagger, we were able to show them in a convenient side-by-side way that allows any researcher to study these taggers and see its features (what features it is extracting), accuracy of POS disambiguation, its tokenization scheme, and more.

Within the parsing step, it is important to mention that we altered the result of taggers with a complex tagset. A complex tagset is a tagset that embodies morphological features (like gender, number or person) in its tags. Since our reference corpora (SALMA (Sawalha et al. 2013) and QAC (Dukes 2011)) use the lemma-plus-features representation, we extract those morphological features and map the complex tag to its base tag. For example, AMIRA has a tag NNS_MD that represents a masculine dual noun. We mapped to NN and assigned morphological features (gender, number) as appropriate. The goal of this transformation is twofold: to compare morphological features with other taggers, and to reduce the sparsity in the POS tagging. This should ease the mapping between the tagsets and improve the quality of the evaluation of those taggers.

6 Word and Morphological Alignment

It is obvious that we must align (morphologically and by token) the output of participating taggers before we can compare their tagging. However, what is not obvious is how this can be done. Alignment in level of word is an easier job as taggers mostly tag every single word. We have not encountered a single case in which two words were tagged with a single tag, as opposed to English, where "sometimes compound names or idiomatic phrases are given a single wordtag" [12, p. 11].

Some taggers drop punctuation marks from their analyses. Some do not repeat the analysis of words they already encountered. Therefore, we adapted a word aligner module that checks against the input text to align it properly. It is a simple aligner that assumes an alignment window of three words, that is, the analysis should correspond to either the current word, the previous or the next word. It aligns the word to the one that has least edit distance.

We learn the morphological alignment (i.e. alignment of morphemes of a single word) from our "multi-tagged corpus". The corpus tagged chapter 29 of the Quran (1000 words and 2000 morphemes) by 12 taggers and manually aligned and proofread (any incorrect solution is marked). While the corpus is meant for evaluation purposes, we found it very useful in building similarity matrices between tags, and use those matrices to align the tokens based on several features: its POS-tag and morphological features (like gender, person, ..etc.).

The process starts with calculating a matrix of the probability of alignment of one particular POS tag in the source tagger to each tag in the target tagset. The element a,b matrix counts the number tag a from tagger A match tag b from gold-standard tagger. We normalize the results by the column total: We use a sequence alignment algorithm (Needleman-Wunch algorithm) to align the two stream of tokens using the matrix. See Table 1 for an example of the output of this stage.

Table 1: Aligned morphemes of the word ولقد *walqd* tagged by several taggers.

MT	**KH**	**AR**	**EX**	**MD & MA**	**BP**	**ST**	**SW**	**MR**	**AM**
Wa	وَلَ حرف العطف	conj	C-	wa_conj	CONJ	CC	p–c–	PUNC	CC
—–	حرف الابتداء	—–	F-	—–	—–	—–	p–z–	—–	—–
Lqd	حَرف تَحقِيق وتَقرِيب	part	F-	part_verb	FUNC_WORD	RP	p–b–	RP	RP

7 Mapping Morphological Features

Mapping means the conversion from one format or value described in the source tagger to the standardised format. We chose the SALMA tagset [29, 30] as it is the most fine-grained tagset in two dimensions: its number of features (15 features) and the possible tags of each word (100 tags). The SALMA tagset has thirty-four possible tags for nouns, one for verbs[3], twenty-two for particles, twenty for others, and twelve for punctuations. It is the most finely-grained tagset in Arabic in terms of tagset size and feature set size.

The mapping involves two components: First, we map morphological features to the values of the SALMA tagset. This mapping is straightforward and it is mostly one-to-one renaming, e.g. mapping from gender=male to gender=m.

However, we made some necessary modifications to the SALMA tagset. In the number feature, the SALMA has nine possible values; typical plural value, has six possible values (i.e. sound plural, broken, etc). Instead, we diminish these differences and use a single value: "p" to represent all types of plurals as all taggers have 3 values in maximum. In addition, we map e for energetic in the Case or Mood possible values. Furthermore, we add *n* for emphatic verb in the case and *m* for non-empahatic. For the full mapping rules of morphological features please see tables 2 and 3.

8 Mapping Main POS tagsets

The second mapping is the mapping of main POS tagsets. While many mappings in the literature involves reducing tagset size, our mapping is the inverse. We "inflate" the solution set by mapping to the most finely-grained tagset (the assumption: more finely-grained than any participating tagset) and then disambiguate the results.

[3] Originally three values that represents the aspect of the verb: perfect, imperfect, and imperative, but we decided to consider them as a morphological feature.

Table 2: The mapping rules of morphological features from all participated taggers to SALMA convention. Part 1

Mood	Tool	SW	Gender	Tool	SW	Case	Tool	SW	Voice	Tool	SW	State	Tool	SW
i	MA	n	masc	AR	m	2	EX	a	active	AR	a	IsDefinitiveAL	MS	d
s	MA	a	fem	AR	f	4	EX	g	pass	AR	p	-IsDefinitiveAL	MS	i
j	MA	j	Masc	MS	m	1	EX	n	a	QT	a	DET	AM	d
u	MA	-	Fem	MS	f	Nom	MS	a	p	QT	p	DT*	ST	d
na	MA	-	Masc	XE	m	Acc	MS	g	Active	XE	a	d	QT	d
I	EX	n	Fem	XE	f	Gen	MS	n	Passive	XE	p	a	QT	-
E	EX	j	_F??	AM	f	n	QT	n	VB	AM	-	i	QT	i
J	EX	j	_M??	AM	m	a	QT	a	VBD	AM	a	i	MA	i
S	EX	a	m	QT	m	g	QT	g	VBP	AM	a	d	MA	d
Ind	MT	n	f	QT	f	na	MA	-	VBN	AM	p	c	MA	-
Jus	MT	j	x	QT	-	u	MA	-	VB	ST	-	na	MA	-
Sub	MT	a	m	MA	m	a	MA	a	VBD	ST	a	u	MA	-
Eng	MT	j	f	MA	f	g	MA	g	VBP	ST	a	i	MD	i
Indicative	XE	n	na	MA	-	n	MA	n	VBN	ST	p	d	MD	d
Subjunctive	XE	a	m	MD	m	na	MD	-	a	MA	a	c	MD	-
Jussive	XE	j	f	MD	f	u	MD	-	p	MA	p	na	MD	-
Energetic	XE	j	na	MD	-	a	MD	a	na	MA	-	u	MD	-
n	QT	n	M	EX	m	g	MD	g	u	MA	-	defArt	AR	d
a	QT	a	F	EX	f	n	MD	n	a	MD	a	indef	AR	i
g	QT	j	-	EX	-	accgen	AR	-	p	MD	p	I	EX	i
i	MD	n	NSUFF_MASC	BP	m	acc	AR	a	na	MD	-	D	EX	d
s	MD	a	NSUFF_FEM	BP	f	NSUFF_*_ACCGEN	BP	-	u	MD	-	R	EX	-
j	MD	j	IV?M?	BP	m	NSUFF_*_ACC	BP	a	A	EX	a	A	EX	-
u	MD	-	IV?F?	BP	f	NSUFF_*_NOM	BP	n	P	EX	p	C	EX	d
na	MD	-	IVSUFF_SUBJ:M	BP	m	ACC	QA	a	PV	BP	a	L	EX	i
acc	AR	a	IVSUFF_SUBJ:F	BP	f	GEN	QA	g	IV	BP	a	NSUFF_*_INDEF	BP	i
_MOOD:I	BP	n	PVSUFF_SUBJ:M	BP	m	NOM	QA	n	CV	BP	a	-NSUFF_*_INDEF	BP	d
_MOOD:SJ	BP	-	PVSUFF_SUBJ:F	BP	f				PV_PASS	BP	p	INDEF	QA	i
_MOOD:S	BP	a	M	QA	m				IV_PASS	BP	p	DEF	QA	d
_MOOD:J	BP	j	F	QA	f				ACT	QA	a			
IND	QA	n							PASS	QA	p			
ENG	QA	j												
JUS	QA	j												
SUBJ	QA	a												

We chose to “inflate” instead of “reduce” because mapping a tagset to a reduced “standard” tagset will cause a loss of information. Thus, for evaluating POS taggers, we will not evaluate the “full tagging” performance of the tagger. In addition, such mapping would force our ensemble tagger to use its reduced tagset.

We want to ensure that we will only have one-to-one and one-to-many situations. That is a tag can be mapped to one or many tags, but no tag on the target can originate from two tags. If we find a “congestion” on one tag (many-to-one), we will extend the target tagset to break this congestion. For example, in QAC tagset, EXP and RES tags (exception and restriction particles) map to one tag in SALMA (p—x- exceptive particle).

Since one-to-many and many-to-many mappings predominated one-to-one and many-to-one [12], we always extend the target tagset (SALMA) to avoid having “many” possible mapping targets.

This “inflating” mapping process we followed can be divided into two stages. In the first stage, we constructed a list of possible mappings from each tagset. To build the list, we first run the tagger on our development corpus. If the morphological analyser provides different possible solutions, we pick the solution that has the least distance to the original voweled word. Next, we log every mapping pair with its example. From this long log of tag pairs, we constructed the top mappings for each tagger that constitutes more than 2%, each assigned

Table 3: The mapping rules of morphological features from all participated taggers to SALMA convention. Part 2

Aspect	Tool	SW	Person	Tool	SW	Number	Tool	SW
pres	AR	c	1	MS	f	sg	AR	s
past	AR	p	2	MS	s	pl	AR	p
imp	AR	i	3	MS	t	dual	AR	d
imperfect	BP	c	1stPer	XE	f	Sing	MS	s
imperative	BP	i	2ndPer	XE	s	Plu	MS	p
perfect	BP	p	3rdPer	XE	t	Dual	MS	d
Pst	MS	p	_??1	AM	f	Sing	XE	s
Prs	MS	c	_??2	AM	s	Dual	XE	d
Imp	MS	i	_??3	AM	t	Plur	XE	p
Perfect	XE	p	f	QT	f	_?S?	AM	s
Imperfect	XE	c	s	QT	s	_?D?	AM	d
Imperative	XE	i	t	QT	t	_?P?	AM	p
VB	AM	i	1	MA	f	NNS	ST	p
VBD	AM	p	2	MA	s	NNS	ST	s
VBP	AM	c	3	MA	t	NNP	ST	s
VBN	AM	-	na	MA	-	NNPS	ST	p
VB	ST	i	1	MD	f	s	QT	s
VBD	ST	p	2	MD	s	d	QT	d
VBP	ST	c	3	MD	t	p	QT	p
VBN	ST	-	na	MD	-	s	MA	s
IV	BP	c	1pers	AR	f	d	MA	d
PV	BP	p	2pers	AR	s	p	MA	p
CV	BP	i	3pers	AR	t	na	MA	-
p	QT	p	1	EX	f	u	MA	-
c	QT	c	2	EX	s	s	MD	s
i	QT	i	3	EX	t	d	MD	d
i	MA	c	IV1??	BP	f	p	MD	p
c	MA	i	IV2??	BP	s	na	MD	-
p	MA	p	IV3??	BP	t	u	MD	-
na	MA	-	IVSUFF_SUBJ:1	BP	f	S	EX	s
i	MD	c	IVSUFF_SUBJ:2	BP	s	D	EX	d
c	MD	i	IVSUFF_SUBJ:3	BP	t	P	EX	p
p	MD	p	PVSUFF_SUBJ:1	BP	f	IV??S	BP	s
na	MD	-	PVSUFF_SUBJ:2	BP	s	IV??D	BP	d
VI	EX	c	PVSUFF_SUBJ:3	BP	t	IV??P	BP	p
VC	EX	i	IVSUFF_DO:1	BP	f	NSUFF_?_SG	BP	s
VP	EX	p	IVSUFF_DO:2	BP	s	NSUFF_?_DU	BP	d
IMPF	QA	c	IVSUFF_DO:3	BP	t	NSUFF_?_PL	BP	p
IMPV	QA	i	PVSUFF_DO:1	BP	f	IVSUFF_SUBJ:*S	BP	s
PERF	QA	p	PVSUFF_DO:2	BP	s	IVSUFF_SUBJ:D	BP	d
			PVSUFF_DO:3	BP	t	IVSUFF_SUBJ:*P	BP	p
			1	QA	f	PVSUFF_SUBJ:*S	BP	s
			2	QA	s	PVSUFF_SUBJ:D	BP	d
			3	QA	t	PVSUFF_SUBJ:*P	BP	p
						IVSUFF_DO:*S	BP	s
						IVSUFF_DO:D	BP	d
						IVSUFF_DO:*P	BP	p
						PVSUFF_DO:*S	BP	s
						PVSUFF_DO:D	BP	d
						PVSUFF_DO:*P	BP	p
						S	QA	s
						D	QA	d
						P	QA	p

with its proportion of all mapping pairs and a list of examples. We used this list to build SAWAREF mapper component.

The second stage involves manually choosing target tags that are most appropriate. We already asked two Arabic linguistic researchers to do a mapping from one tagset (MADAMIRA's); we unite the two mapping rules to build the mapping.

Our method expands the solution set of each tagger, which increases ambiguity. However, this could make for a fairer voting between taggers and the most probable mapping should be selected in the POS disambiguation stage.

9 Case study: Tagging Adjectives

While the complete system aims to morphologically and syntactically tag a text in a more robust way, we found it useful for some other purposes. We studied the accuracy of tagging adjectives, and show that they are commonly mistagged as *nouns*. The cause of this confusion is the definition of adjectives in Arabic. In traditional Arabic grammar, an adjective is marked when it qualifies its preceding corresponding noun, i.e. attributive adjective. In this case, attributive adjectives agree with their corresponding noun in definiteness, number, case and gender. For example, رجل طويل *Rajol Taweel* (a tall man). Taggers agree mostly on tagging "tall" as an adjective. However, taggers often vary in tagging "tall" in predicative adjectives: هذا الرجل طويل *Hatha Rajol Taweel* (This man is tall).

We used chapter twenty-nine of the holy book which is a parallel annotated corpus (PAC) by [31, 32] to study the tagging of adjectives. We feed the chapter to the SAWAREF toolkit up until the morphological analysis stage. We ended up with a spreadsheet CSV file where each morpheme is tagged by SAWAREF taggers and the results are aligned on the morpheme level. We used the file to analyse the case of tagging adjectives.

Table 4: Tagging adjective morphemes by two manually annotated corpora. Recall = 0.38, Precision=0.85.

		SALMA	
		Nj—-	**Others**
QAC	**ADJ**	11	2
	N (noun)	18	N/A

Surprisingly, the two manually annotated corpora were not in agreement in tagging adjectives. Table 4 shows the confusion matrix of tagging adjective morphemes. We can see that in only 11 out of 31 cases, the two manually annotated corpora agree on the tagging. In 18 cases, QAC tagged them as NOUN. One reason behind this low recall and precision is the incompatibility of tagsets: QAC tagset is "syntax"-driven while SALMA is morphologically driven. The following verse (no. 26) illustrates the difference in tagging adjectives:

Table 5: One sentence shows how linguistics do not agree on tagging predicative adjectives.

Word (AR)	QAC	SALMA	Translation
انه	innahu	innahu	(Indeed,
هو	howa	howa	He is
العزيز	alaziz/N	alaziz/nj—-	the Exalted in Might,
الحكيم	alhakeem/ADJ	alhakeem/nj—-	the Wise).

The word *alaziz* was tagged by QAC as a *noun* as it is acting as a predict (*khabar*) in traditional grammar. SALMA tagged it however as an *adjective*. However, QAC is not consistent in this matter; verse 19 of chapter 29: إن ذلك على اللّٰه يسير "that for Allah is easy/ADJ" is not consistent with its following verse: إن اللّٰه على كل شيء قدير "Indeed Allah , over all things, is competent/N". The words: "easy/ADJ" and "competent/N" are both adjectives acting as predict (*khabar*) and should be treated similarly.

The same confusion carried over to SAWAREF participant taggers: in the case QAC is gold standard, the average f-score is 0.11 (precision=0.14, recall= 0.2). With SALMA, the average f-score is 0.12 (precision=0.22, recall= 0.14). These very low scores show how hard it is to tag adjectives correctly. The full precision and recall of each tagger is reported in table 6. As a conclusion, even though adjectives play an important role in the semantic level, they need more robust definition and more investigation on how to predict them in Arabic specifically.

Table 6: The precision, recall and f-score of predicting adjectives in the chapter twenty-nine of the holy quran. We used QAC as actual tagging (tag = ADJ) for the top scores. We used SALMA gold standard (tag = nj—-) in bottom scores.

QA as GS	**MT**	**KH**	**AR**	**EX**	**MD**	**MA**	**AL**	**BP**	**BJ**	**ST**	**MR**	**WP**	**AM**	**SW**
Precision	0.11	1.00	0.07	0.04	0.03	0.12	0.07	0.05	0.11	0.18	0.00	0.00	0.14	0.38
Recall	0.62	0.08	0.15	0.23	0.08	0.23	0.15	0.15	0.23	0.46	0.00	0.00	0.31	0.85
f-score	0.19	0.14	0.10	0.07	0.05	0.16	0.10	0.08	0.15	0.26	0.00	0.00	0.20	0.52
SW as GS	**MT**	**KH**	**AR**	**EX**	**MD**	**MA**	**AL**	**BP**	**BJ**	**ST**	**MR**	**WP**	**AM**	**QA**
Precision	0.16	1.00	0.07	0.05	0.13	0.24	0.07	0.08	0.14	0.29	0.00	0.03	0.21	0.85
Recall	0.41	0.03	0.07	0.14	0.14	0.21	0.07	0.10	0.14	0.34	0.00	0.03	0.21	0.38
f-score	0.24	0.07	0.07	0.08	0.14	0.22	0.07	0.09	0.14	0.32	0.00	0.03	0.21	0.52

10 Next steps: Solutions Alignments and POS disambiguation

In this section we describe our plans for the remaining stages of the system: solution alignment and grouping and POS disambiguation. Because Arabic is highly ambiguous, a word analysis usually has several analysis solutions. Some taggers (i.e. morphological analysers) produce different possible analysis with no ranking. We mentioned also in the last section that we might increase the solution set size after the mapping process. The pool of solutions needs to be sorted so the ensemble system can vote for the best solution. This requires those solutions to be aligned such that similar solutions are grouped together.

We plan to perform this step by defining the distance function between the two solutions. Each solution is represented in a vector of values, and there are a plenty of distance measures to compute the similarity of two vectors (e.g. cosine similarity). We then simply group the solutions that reach a certain threshold.

Finally, with a list of solutions (originated from the taggers), the system will use a variety of POS disambiguation techniques to select the most probable tags. This includes using tagger predictions, weighting solutions based on frequency, and context-aware sequence labelling techniques (e.g. HMM).

Since we use taggers that vary in their kernel technique, we expect the accuracy to be higher than any participating tagger. But this comes with a price: the ensemble tagger is computationally much more expensive, and the throughput is lower. However, we target applications where tagging is performed offline (e.g. annotating a corpus) and the accuracy is more important than efficiency. The system is designed such that some taggers can be turned off, and we plan to employ an adaptive technique to switch off taggers that do not add to the accuracy.

11 Conclusion

We identified the key parts of the SAWAREF system and showed the stages of the ensemble POS tagger process. We showed how we deal with key obstacles in the ensemble method, namely morphological alignment, diversity in tagset, and multiple solutions per token. SAWAREF currently runs multiple taggers, standardizes their results, and aligns the result of each analysis. An expected issue is low agreement among Arabic linguists on the definitions of grammatical categories, as exemplified by the tagging of adjectives.

Acknowledgments

Abdulrahman Alosaimy owes his deepest gratitude to Al-Imam University in Riyadh, Saudi Arabia for funding this research and his PhD study. We thank Abdullah Alfaifi (Imam University-Riyadh and University of Leeds) and Ayman Alghamdi (Umm AlQura University-Mecca and University of Leeds) for volunteering their Arabic linguistics expertise in advising our research.

References

[1] Halteren, H.V., Zavrel, J., Daelemans, W.: Improving Accuracy in Word Class Tagging through the Combination of Machine Learning Systems. Computational Linguistics **27** (2001) 199–229

[2] Montolio, J.: Improving POS Tagging Using Machine-Learning Techniques. (1997) 53–62

[3] Schroder, I.: A case study in part-of-speech- tagging using the ICOPOST toolkit. (2002)

[4] Søgaard, A.: Ensemble-based POS tagging of Italian. IAAI-EVALITA, Reggio Emilia, Italy (2009)

[5] Henrich, V., Reuter, T., Loftsson, H.: CombiTagger: A System for Developing Combined Taggers. (2007) 254–259

[6] Kobyliński, L.: PoliTa: a Multitagger for Polish. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), Reykjavik, Iceland, European Language Resources Association (ELRA) (2014) 2949–2954

[7] Śniatowski, T., Piasecki, M.: Combining. In: Combining polish morphosyntactic taggers. Springer Berlin Heidelberg, Berlin, Heidelberg (2012) 359–369

[8] RamaSree, R., Kumari, P.K.: Combining pos taggers for improved accuracy to create telugu annotated texts for information retrieval. Dept. of Telugu Studies, Tirupathi, India (2007)

[9] Sjöbergh, J.: Combining POS-taggers for improved accuracy on Swedish text. Proceedings of NoDaLiDa 2003 (2003)

[10] Aliwy, A.H.: Arabic Morphosyntactic Raw Text Part of Speech Tagging System. (2013)

[11] Atwell, E., Hughes, J., Souter, C.: AMALGAM: Automatic Mapping Among Lexico-Grammatical Annotation Models. Proceedings of ACL workshop on The Balancing Act: Combining Symbolic and Statistical Approaches to Language (1994) 11–20

[12] Atwell, E., Demetriou, G., Hughes, J., Schiffrin, A., Souter, C., Wilcock, S.: A comparative evaluation of modern English corpus grammatical annotation schemes. (1998) 7–24

[13] Rabiee, H.S.: Adapting Standard Open-Source Resources To Tagging A Morphologically Rich Language: A Case Study With Arabic. Volume 0. (2011) 127–132

[14] Alabbas, M., Ramsay, A.: Combining strategies for tagging and parsing Arabic. ANLP 2014 (2014) 73–100

[15] Alabbas, M., Ramsay, A.: Combining black-box taggers and parsers for modern standard Arabic. In: Computer Science and Information Systems (FedCSIS), 2012 Federated Conference on. Number Ldc, IEEE (2012) 19–26

[16] Boudlal, A., Lakhouaja, A., Mazroui, A., Meziane, A., BEBAH, M.O.A.O., SHOUL, M.: Alkhalil morpho sys1: A morphosyntactic analysis system for arabic texts. In: International Arab Conference on Information Technology. (2010)

[17] Buckwalter, T.: Buckwalter Arabic Morphological Analyzer Version 1.0. (2002)
[18] Smrz, O.: Functional Arabic Morphology. Formal System and Implementation. The Prague Bulletin of Mathematical Linguistics (2007)
[19] Habash, N.: Arabic Morphological Representations for Machine Translation. In: Arabic Computational Morphology. Text, Speech and Language Technology. Springer Netherlands (2007) 263–285
[20] Attia, M., Pecina, P., Toral, A.: An open-source finite state morphological transducer for modern standard Arabic. Proceedings of the 9th International Workshop on Finite State Methods and Natural Language Processing (2011) 125–133
[21] Beesley, K.R.: Arabic Morphology Using Only Finite-State Operations. Proceedings of the Workshop on Computational Approaches to Semitic languages (1998) 50–57
[22] Pasha, A., Al-Badrashiny, M., Diab, M., El Kholy, A., Eskander, R., Habash, N., Pooleery, M., Rambow, O., Roth, R.M.: Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of arabic. In: Proceedings of the Language Resources and Evaluation Conference (LREC), Reykjavik, Iceland. (2014)
[23] Habash, N., Rambow, O., Roth, R.: MADA+TOKAN: A Toolkit for Arabic Tokenization, Diacritization, Morphological Disambiguation, POS Tagging, Stemming and Lemmatization. Proceedings of the Second International Conference on Arabic Language Resources and Tools (2009) 102–109
[24] Diab, M.: Second generation AMIRA tools for Arabic processing: Fast and robust tokenization, POS tagging, and base phrase chunking. Conference on Arabic Language Resources and Tools (2009) 285–288
[25] Toutanova, K., Klein, D., Manning, C.D.: 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 (NAACL '03), (2003) 252–259
[26] Thomas, M.: Efficient Higher-Order CRFs for Morphological Tagging. Emnlp (2013) 322–332
[27] Gahbiche-Braham, S., Bonneau-Maynard, H.: Joint Segmentation and POS Tagging for Arabic Using a CRF-based Classifier. Lrec (2012) 2107–2113
[28] Monroe, W., Green, S., Manning, C.D.: Word segmentation of informal Arabic with domain adaptation. ACL, Short Papers (2014)
[29] Sawalha, M., Atwell, E.: A standard tag set expounding traditional morphological features for Arabic language part-of-speech tagging. Word Structure **6** (2013) 43–99
[30] Sawalha, M., Atwell, E., Abushariah, M.a.M.: SALMA: Standard arabic language morphological analysis. 2013 1st International Conference on Communications, Signal Processing and Their Applications, ICCSPA 2013 (2013)
[31] Dukes, K.: The Quranic Arabic Corpus. School of Computing, University of Leeds, UK (2011)

[32] Sawalha, M.: Open-source resources and standards for Arabic word structure analysis: Fine grained morphological analysis of Arabic text corpora. (2011)