Joint Segmentation and POS Tagging for Arabic Using a CRF-based Classifier

Souhir Gahbiche-Braham[1,2], Hélène Bonneau-Maynard[1,2], Thomas Lavergne[1], François Yvon[1,2]

(1) LIMSI-CNRS, (2) Université Paris-Sud 11
91403 Orsay, France
{souhir,hbm,lavergne,yvon}@limsi.fr

Abstract

Arabic is a morphologically rich language, and Arabic texts abound of complex word forms built by concatenation of multiple subparts, corresponding for instance to prepositions, articles, roots prefixes, or suffixes. The development of Arabic Natural Language Processing applications, such as Machine Translation (MT) tools, thus requires some kind of morphological analysis. In this paper, we compare various strategies for performing such preprocessing, using generic machine learning techniques. The resulting tool is compared with two open domain alternatives in the context of a statistical MT task and is shown to be faster than its competitors, with no significant difference in MT quality.

Keywords: Arabic Segmentation, Arabic POS Tagging, Arabic preprocessing, Conditional Random Fields

1. Introduction

The peculiarities of the Arabic language make the automatic processing of Arabic texts challenging. In particular, the Arabic language has a complex morphology and an ambiguous writing system. Furthermore, the clitic agglutination phenomenon results in a considerable variability of surface forms. For instance, the analysis of a parallel corpora (Nguyen and Vogel, 2008) reports that while the English side contains 6.2M tokens and 68K types, the 5.2M Arabic tokens correspond to approximately 155K different types. Therefore, data driven NLP methods, when applied to Arabic, need to be complemented with sophisticated preprocessing techniques aimed at computing a morphological decomposition for Arabic words, in order to mitigate data sparseness and the correlated estimation problems. As possible combinations of affixes and roots depend on the main category of the word, morphological decomposition and part-of-speech (POS) tagging have to be performed jointly, a combined task that has been approached in many different ways in the literature. In this study, an additional concern is speed : Statistical Machine Translation (SMT) being our main target application, it is crucial to be able to rapidly process large amounts of Arabic texts.

In this paper, we propose to perform this joint prediction task using Conditional Random Fields (CRFs), which have proven to deliver state-of-the art results in many sequence labelling tasks. Taking advantage of the ability of our CRF package, Wapiti (Lavergne et al., 2010), to cope with large label sets and extremely large sets of features, while keeping the computational cost minimum, we consider and compare various ways to perform POS tagging and clitic splitting. These strategies are evaluated both on intermediate tasks, but also on the final MT application.

The rest of this paper is organized as follows: in Section 2., we present some existing approaches for arabic segmentation. Section 3. describe the Wapiti toolkit. In Section 4., we present our approach based on CRFs, and give some results in Section 5. Finally, Section 6. concludes the paper.

2. Related work

2.1. Analyzing Arabic words

Many studies have focused on Arabic word segmentation and tagging. The most popular tool is probably MADA (Nizar Habash and Roth, 2009), which implements a two-stage process to select the best possible morphological decomposition/analysis among the ones proposed by the BAMA (Buckwalter, 2004) tool. MADA is described in section 2.2.

AMIRA (Diab, 2009) implements a different approach, where the clitic splitting is performed independently from POS tagging, using a Support Vector Machine (SVM) applied to an IOB annotation scheme on every Arabic character. Also noteworthy is the work of Marsi et al. (2005), in which memory-based learning is used for morphological analysis and part-of-speech tagging of Arabic. The authors use k-nearest neighbor classification and show that the tagger can be used to select the appropriate morphological analysis.

Mansour (2010) presents MorphTagger, a Hidden-Markov-Model segmentation tool for Arabic and compares it with MADA (Nizar Habash and Roth, 2009) and with the earlier and simpler work of El Isbihani et al. (2006). A more detailed description of this tool is given in section 2.3.

In (Kulick, 2010), affix-splitting and part-of-speech tagging are performed simultaneously with a simple classifier, without using morphological analysis. In a more recent work (Kulick, 2011), this approach is extended, using a distinction between open-class (such as preposition, relative pronoun, etc.) and closed-class (such as noun, verb, proper nouns, etc.) tokens, which differ in their possible morphological affixations and in their frequencies. A list of proper nouns extracted from the SAMA-v3.1 (Maamouri et al., 2010) database is also used as feature.

2.2. MADA

In this section, we give more details regarding MADA and the associated resources, as this tool will be one of our main point of comparison. MADA (Morphological Analysis and Disambiguation for Arabic) is a morphological disambiguation system. It operates in steps: first, it uses the mor-

phological analysis and generation system Almorgeana[1] to produce a list of potential analyses for each word in the text without considering the word context of occurrence. All the possible segmentations of the input into prefix-stem-suffix and bilateral compatibility are checked with respect to the BAMA database. Only valid triples will be further considered by MADA.

MADA then makes use of up to 19 features to rank the list of possible analyses. Five features use the SRILM toolkit[2] to give information about morphological disambiguation such as spelling variations and n-gram statistics. The fourteen remaining morphological features correspond to morphological information (Habash et al., 2010) such as part-of-speech, presence/absence of proclitics or enclitics, aspect, case, gender, mood, number, person, etc. Four of them are represented in the analyses returned by BAMA (Habash and Rambow, 2005) while the ten other morphological features are predicted independently using the SVMTool classifier (Giménez and Màrquez, 2004); each classifier prediction is then weighted and the collection of feature predictions is compared with the list of potential complete analyses. These are then ranked and the highest scoring one is finally selected, providing the predicted value for all morphological features. Based on this morphological analysis, words can be segmented according to predefined, deterministic, segmentation schemes.

In the specific context of SMT applications, using MADA and its a very precise morphological description, is probably an overkill, especially when all is needed is to split a small subset of affixes. The associated computational cost is indeed rather high: it requires to run, for each token, multiple SVM classifiers before combining results to take decisions. Processing large texts is then only possible by distributing the computation on several machines and typically takes a very substantial amount of the total system building time.

2.3. MorphTagger

MorphTagger is a Hidden-Markov-Model segmenter for Arabic. Since this tool has also been designed to quickly compute coarse morphological analyses in the context of MT applications, it will be another interesting point of comparison. MorphTagger was first applied for the task of POS tagging in Hebrew and then adapted to the Arabic language (Mansour et al., 2007). A segmenter level and few normalization rules were then added to the tool. The architecture is similar to MADA, as it uses BAMA database. Therefore, in the first step, Arabic text goes through the BAMA morphological analyzer, which outputs for each word all possible analyses including the corresponding POS tag. At the other end of the pipe-line, MorphTagger outputs the most probable tag sequence according to the model. Subsequently, the choice of the correct analysis is made by choosing the most probable morpheme given the tag. The SRILM toolkit is used for disambiguation.

Once the morphological analysis is performed, prepositions (excluding the Arabic determiner) and possessive and objective pronouns are splitted using several hand-crafted rules. The segmenter also performs few normalization steps, the most noticeable of which being (i) Alif maksura, reverted to the original form when a word, splitted from a suffix, ends with Alif maksura (yX → Y+X); (ii) feminine marker: reverted to its original form when a noun is split from a suffix (tX → p+X) and (iii) the definite article *Al* is reverted to the original form, after splitting, when preceded by *l* prefix (llX → l+Al+X).

(Mansour et al., 2007) compared MorphTagger to MADA and to the FST-based segmenter originally introduced in (El Isbihani et al., 2006) and shows that MorphTagger gives better translation results on different translation conditions and different test sets.

3. Wapiti

For the purpose of these experiments, we use the Wapiti[3] toolkit developed within our group (Lavergne et al., 2010). Using CRFs to (partially) reproduce the analysis performed by MADA requires to simultaneously predict several characteristics for each words. A typical setup for this kind of tasks is a cascade of predictors, each of them relying on the predictions made by the previous one(s). If the order is chosen with care, this setup can be very effective; in particular, if some predictors do not rely on the others, they can be run in parallel. However, when predicting closely related characteristics, join prediction of *composite labels* can prove to be a more effective strategy.

The join prediction setup however implies to manipulate larger labels set, which can be computationally challenging: recall that the training of CRFs has a quadratic complexity with respect to the size of the label set. Wapiti was designed for such tasks and can easily handle sets of several hundred of labels. Using ℓ_1 regularization (Gao et al., 2007), the different training algorithms can build highly sparse models and take advantage of this sparsity to speed-up the computations. This allows us to jointly predict some of the closely related characteristics, ensuring that only valid combinations are predicted and thus delivering better performances. Working with large labels sets also poses estimation problems, due to the data-sparsity that naturally arises in such situation. As each label occurs less frequently in the training data, the related feature weights are more difficult to estimate accurately. In the problem considered here, as it is often the case in NLP applications, these sets of composite labels have a structure, which can be used to smooth the estimates. We therefore use the ability of Wapiti to define features which only test on sub-parts of the observations; in fact, it is possible to define feature using arbitrary regular expressions over the observation sequence.

4. Our approach: Part-of-Speech tagging and Arabic segmentation

Recall that our main goal is to tokenize Arabic texts so as to strip off prefixes and reproduce the behavior of MADA for segmentation purposes, as in *i.e.* (Habash and Sadat, 2006), but to do so at a much greater speed and using as few resources as possible. This is because morphological analysis is only a preprocessing step, which must be applied

[1] Almorgeana uses the BAMA database (Buckwalter, 2004).

[2] `http://www.speech.sri.com/projects/srilm`.

[3] `http://wapiti.limsi.fr`

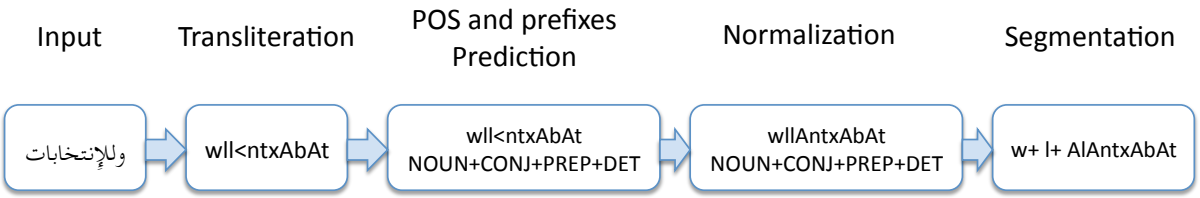

Figure 1: Overview of the segmentation process.

to large amount of parallel texts when developing statistical machine translation systems.

In this context, we only aim at predicting features that are directly related to word segmentation, in addition to the main part-of-speech tag. In principle, this would require to encode the CRF output composite label as: POS+pr1+pr2+pr3, where POS is the main part-of-speech tag, and the labels pr1, pr2 and pr3 respectively encode the absence/presence and types of possible prefixes (see Table 1). The first one concerns the conjunctions *w+* and *f+*; pr2 deals with the prefixes *b+*, *l+*, *k+* and *s+*, which can appear together with conjunctions; the last label indicates the presence of the definite article *Al+*. The value of pr1 is CONJ for conjunctions or *none*; pr2 is equal to PREP if the prefix corresponds to a preposition, to SUB if pr2 corresponds to a subordinating conjunction, to FUT to indicate the future mark, or to *none*; finally pr3 can be DET or *none*.

Prefix	Label/Value
pr1	CONJ/*w+*, *f+* or *none*
pr2	PREP/*b+*, *l+*, *k+* or SUB/l+ or FUT/*s+* or *none*
pr3	DET/*Al+* or *none*

Table 1: Prefixes, labels and values

Using this scheme, the word وللإنتخابات (*and for the votes*) would, for instance, be tagged NOUN+CONJ+PREP+DET.

Regarding syntactic categories, we used the list of the main 24 POS tags of the Arabic Treebank (Maamouri et al., 2005a; Maamouri et al., 2005b). Note however that some prefixes can only appear with words carrying a specific POS. An example is the prefix *s+*, which can only be associated with verbs, to indicate the future tense. This means that the effective number of labels is much less than the total number of possible labels (384).

Figure 1 displays the various steps of the segmentation process. In our approach, Arabic texts are transliterated using Buckwater transliteration scheme[4]. Segmentation prediction is then performed, followed by a normalization step and finally by the splitting process itself, which is based on a handful of simple rules.

4.1. Models and feature selection

A Part-Of-Speech and segmentation prediction model makes its prediction based on simple descriptors of the input word sequences. In Wapiti, these features are described via generic patterns that simultaneously test unigram and bigram of labels and arbitrary features of the observation sequence. In our experiments, these tests on the input are defined as follows: (1) *unigram* tests, which evaluate the presence/absence of individual words in a sliding window of 7 words around the word in focus (2) *bigram tests* which evaluate the presence/absence of word bigrams in a sliding window of 5 words, and (3) *trigram tests*, which consider a sliding window of 3 words. We also used as features (4) *prefixes and suffixes tests*, which consider up to the 5 first (respectively last) characters within a sliding window of 3 words. Finally (5) *punctuation and digits* features test for the presence or absence punctuation marks and digits in a sliding window of 5 words.

4.2. Normalization

The next step concerns the normalization of four Arabic characters such as done in MADA (Habash, 2010).

The different written forms of Alif إ, أ, آ and ا are normalized by ا (in transliterated texts, respectively <, > , | , {, *A* are replaced by *A*). The Yaa Maqsura ى (*Y*) and Yaa ي (*y*) are normalized to ي (*y*) , the Taa Marbuta ة (*p*) becomes ه (*h*) and the different forms of Hamza ؤ (&), ئ (}), ء (') are normalized to ء (').

4.3. Segmentation rules

After normalization, segmentation is performed by applying set of rules to check prefixes. This task can be summarized by four major rules. The first rule splits off in two

[4] http://www.qamus.org/transliteration.htm

parts words containing only a pr1 prefix. The second rule checks whether a word containing both pr1 and pr2 prefixes is tagged as a preposition; if this is not the case, it splits off the word into three parts. The third rule verifies that a word contains only a pr2 prefix and is not a preposition. If these conditions are satisfied, the word is split into two parts.
The last rule concerns the pr3 prefix: each word will be split according to the preceding rules, with an additional condition to change the words containing *l* prefix followed by the definite article *Al*. This additional rule consists of adding the *A* character of the definite article *Al* which has been altered due to the agglutination phenomenon and Arabic morphosyntactic rules. For example, if the word begins with *ll* or *wll* or *fll*, after applying segmentation, the word becomes *l+ Al*, *w+ l+ Al* or *f+ l+ Al*.

5. Experiments and Results

5.1. Data

To train our POS-tagging and segmentation models, we used the POS tagged Arabic Treebank (ATB), which includes 498,339 tokens (18,826 sentences).
Each token as it appears in the original Newspaper is accompanied in the Arabic treebank by its transliteration according to Buckwalter's transliteration scheme and all its possible vocalizations, POS tag sequences and morphological analyses. The correct analysis is specified in the ATB with a preceding *. Figure 2 presents all the possible morphological analyses of the word *whw* وهو according to the BAMA database.

$$*wa/CONJ + huwa/PRON\_3MS$$
$$wa/CONJ + huw/NOUN\_PROP$$
$$whw/NOUN\_PROP$$
$$wa/CONJ + hw/NOUN\_PROP$$

Figure 2: Different segmentations proposed by BAMA for the word *whw* (وهو), which means *"and he"*. The correct segmentation is marked with a *).

According to the Arabic Treebank, around 17% of the tokens have to be segmented. Note that the number of observed combinations for POS+pr1+pr2+pr3 labels in the whole Arabic treebank is only 88, which is much less than the total number of composite tags (384).
For the segmentation task, we ran a series of experiments with feature sets of increasing complexity. We first evaluated the POS tagger (section 5.2., which then extended to include the prediction of prefixes. Different approaches were explored and are described in section 5.3. All these models were evaluated using 10-fold cross-validation, with test sets of about 2000 sentences.
For the translation task, we evaluate the influence of varying preprocessing tools using the AFP[5] Arabic-French news made available through the SAMAR project[6]. Standard translation systems are trained on 145K phrase pairs extracted from a comparable corpora (Gahbiche-Braham et al., 2011). The original text contains 3.3M tokens, corresponding to 106K types, while the number of tokens in the *preprocessed* text is 3.6M tokens, comprising only 75K types. The French side of the parallel corpus contains 4.2M tokens, which corresponds to 61K types. Performance is measured using automatic metrics such as BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005) and HTER (Snover et al., 2006).

5.2. Part-Of-Speech Tagging

Our "small" POS tagger only includes unigram tests in a window of three words and limited prefix and suffix tests. This system achieves an error rate of 5.34%. By extending the context window to include the previous and next three words, and by using bigram and trigram tests on the observation, we managed to improve this baseline by more than 1 point. Our best results to date are obtained with a large system comprising more than 210M features, out of which the ℓ_1 penalty term only selects 2.4M; the corresponding error rate is **4.2%** to be compared with MADA, which has a POS error rate of 3.77%[7]. It should be stressed that these results are obtained *without using BAMA*. This means that we are in fact solving a more complex problem than MADA, as we do not use any prior morphological analyzer.

5.3. Segmentation Prediction

Three different schemes for segmentation are compared. In the first scheme (SEG), prefixes are predicted without using any POS feature. In the second scheme (POS-then-SEG), the POS tags are first predicted, and then used as supplementary feature to predict prefixes. In the last scheme (POS+SEG), POS tags and prefixes are predicted simultaneously.

5.3.1. Results

Table 2 reports the number of features according to the segmentation scheme as well as the number of active features which are selected by Wapiti for each model. For POS and SEG tasks, we used the largest possible number of patterns to build the models. As these patterns generate too many features for Wapiti to cope with, the number of patterns was reduced for the POS+SEG schemes (the sliding window for prefixes/suffixes is reduced to one), and accordingly for the POS-then-SEG condition. For this latter task, the number of features represents the sum of the active features for POS and for predicting the segmentation labels. Therefore the total number of possible features for the POS+SEG condition is 1,511M, generating more than 4M active features; with the same patterns, the total number of possible features is 245M for POS-then-SEG condition, with 3M active features (more than the active features of SEG), since we used the POS feature which is not used for SEG scheme.
Table 3 shows comparative results with the three schemes and reports error rates for (i) predicting the composite label pr1+pr2+pr3, (ii) predicting each prefix independently and (iii) after applying normalization and segmentation. This latter score only evaluates the output segmentation, which

[5] AFP news agency
[6] http://www.samar.fr/

[7] We reduced the 34 POStags of MADA to 24 to compare the two segmenters

Scheme	#features	#active features
POS	195.6M	2,262.4K
SEG	64.3M	574K
POS-then-SEG	245.0M	3,024.4K
POS+SEG	1,511.5M	4,168.9K

Table 2: Total number of features and active features for each Scheme.

can be correct even when the predicted prefix category is erroneous.

Scheme	SEG	POS-then-SEG	POS+SEG
pr1+pr2+pr3	0.78%	0.64%	**0.60%**
pr1	0.22%	**0.18%**	**0.18%**
pr2	0.46%	0.35%	**0.34%**
pr3	0.13%	0.13%	**0.11%**
POS	-	4.20%	**3.72%**
After segmentation	0.55%	0.42%	**0.40%**

Table 3: Segmentation Error Rate of the different schemes

Even for the smaller model (SEG scheme), the error rate is less than 1% (only 0.78% for the joint prediction of the 3 proclitics). Taking into account the POS feature in a first step (POS-then-SEG scheme) allows to decrease error rate for both proclitic prediction and segmentation. Finally, the POS+SEG scheme yields the best results, with a 0.6% error rate compared to 0.64% for POS-then-SEG . Note that jointly predicting proclitics and POS allows to even improve over the simpler POS tag prediction task, and also compared to MADA, with a POS error rate of **3.72%** to be compared to 4.2% when predicting POS alone. As may be observed, the POS tag is an important feature, as it allows to achieve a 0.18% improvement for predicting pr1+pr2+pr3. Finally, the segmentation error rate is 0.55% for SEG scheme and is reduced to 0.40% for the POS+SEG scheme. Since MADA D2 (Habash and Sadat, 2006) splits off the same prefixes as in our scheme, a comparison at the segmentation level method has been performed and has shown that our approach allows us to obtain better segmentation performance (the error rate for the POS+SEG scheme is 0.40%, slightly better than the 0.57% segmentation error rate obtained by MADA D2 on the same data). Such comparison is not possible with MorphTagger, since it splits off prefixes and possessive and objective pronouns while in our reference only prefixes are split.

5.3.2. Error Analysis

More details regarding the generated errors by the various segmentation schemes are presented in Table 4. Most of the segmentation errors generated by pr1 and pr2 are words which are not segmented (seg- in the table) whereas for pr3, most of the errors are over-segmentations (seg+ in the table). These over-segmentations of the pr3 prefix are due, in addition to the fact that the definite article *Al* can exists attached to nouns, it can be a part of Arabic proper nouns. For this reason, proper nouns containing *Al* can be split erroneously. According to these results, segmentation is done by splitting only pr1 and pr2 prefixes if they exist.

Scheme	pr1 (%)		pr2 (%)		pr3 (%)	
	seg+	seg-	seg+	seg-	seg+	seg-
POS+SEG	39.5	60.5	28.9	65.3	76.8	23.2

Table 4: Details for Segmentation Errors

As described in Section 4.3., the detection of the definite article *Al* is very helpful for segmentation, especially for words containing the *l* prefix followed by *Al*. For instance, the word وللوطن (*llwtn*, which means *for the homeland*) would be tagged as NOUN+none+PREP+DET and becomes after segmentation *l+ Alwtn* according to the specified rules. If the definite article *Al* is not detected, this word would be erroneously analysed as *l+lwtn* (*for for a homeland*).

Analysing more closely the segmentation errors, we notice that 6.70% of the errors are related to over-segmented named entities. For instance, the surname بلقاضي (*blqADy*) was tagged as noun+none+PREP+none instead of noun_prop+none+none+none and as a result the word will be segmented to *b+ lqADy* since it begins with the letter *b*.

5.4. Translation experiments

The last series of experiments aims at evaluating the effect of the segmentation process on Arabic to French translation performance. Different SMT systems were trained using the same data with the Arabic part of the bitext preprocessed by different tools (MADA, MorphTagger, and our segmenter) with a classical setup. The preprocessed Arabic and French data are aligned using MGiza++ (Gao and Vogel, 2008). The Moses toolkit (Koehn et al., 2007) is then used to make the alignments symmetric and to extract phrases with maximum length of 7 words. Feature weights are set by running MERT (Och, 2003). Two schemes were tested for MADA: MADA-D2, since it splits off the same prefixes as our segmenter (Habash and Sadat, 2006), and MADA-TB since, it is the recommanded segmentation scheme for Machine Translation applications (Kholy and Habash, 2012). BLEU scores on a are summarized in Table 5, along with METEOR and TER values.

	BLEU	METEOR	TER
MADA D2	32.8	54.2	60.3
MADA TB	32.9	54.2	59.1
Morphtagger	33.2	54.5	58.8
SEG	32.8	53.4	59.6
POS-then-SEG	33.1	53.7	59.4
POS+SEG	33.3	54.0	59.1

Table 5: Machine Translation Results

It can be observed that our segmenter allows us to achieve the same translation results as MADA and MorphTagger (32.8 to 33.3 BLEU points compared to 32.8 and 32.9 for

MADA and 33.2 for MorphTagger). Improving the segmentation level using POS information yields small improvements in translation results: POS+SEG scheme gives the best scores over our three segmentation schemes and over the three metrics. Therefore, it seems that predicting POS tagging and segmentation simultaneously is slightly better than predicting them separately, even for the MT task. Our POS+SEG scheme achieves improvements respectively of 0.5 BLEU and 0.4 BLEU relatively to MADA D2 and MADA TB schemes and gives the same performance as MorphTagger.

Table 6 presents a comparison of the speed - given in words per second (w/s)[8] - of the different segmenters. The differences in speed performance are very significant: our segmenter is about 30 times faster than MADA and 30% faster than MorphTagger. As shown in Tables 5 and 6, our segmenter is as efficient as MADA and MorphTagger to preprocess SMT input. The main advantage of our tool is that it is considerably faster than its competitors and does not require install any other additional resource. In fact, both MADA and MorphTagger require to install the SRILM toolkit (Stolcke, 2002) and to have access to the BAMA database. Using BAMA as a pre-processor is an obvious choice; but without using it, segmentation can be performed at a much faster pace. MADA uses also SVMTool to predict some features (section 2.2.), and this is what makes it extremely slow.

	speed (w/s)
MADA	90
MorphTagger	2020
POS+SEG	2960

Table 6: Preprocessing speed in words per second

By comparison, to use our tool, we just need to install the Wapiti toolkit together with the accompanying trained model and a couple of segmentation wrappers.

6. Conclusion

In this paper, we proposed an efficient approach to perform the POS tagging and clitic splitting for Arabic language. For this purpose, we used the Wapiti toolkit based on CRFs which is able to cope with large label sets and very large sets of features. We have evaluated this Arabic preprocessing module as a stand alone module as well as a front-end to a standard statistical machine translation task, and have compared it to the MADA and MorphTagger toolkits.

First results have shown that the Wapiti model is almost as good as MADA when used as a mere perform POS tagging. It also allows us to achieve a very low error rate on the prefix segmentation task. Furthermore, we have found that performing these two tasks in a joint fashion yield slightly improved performance: for instance, it allowed us to reduce POS error rate from 4.2% down to 3.72%.

We have also checked that using this tool in a complete SMT pipeline delivers results that are in the same ballpark as the results obtained with other preprocessing chains. As compared to existing alternatives, this tool is (i) able to process several thousands of words per second and (ii) is totally independant of any other resource, such as morphological analyzer or disambiguator. The results obtained so far are promising and suggests several perspectives for further improving the Arabic preprocessing chain. In particular, we intend to complement the preprocessing chain with named entity recognition using Wapiti as well; here again, various scenarios can be considered for performing this series of tasks.

[8]For these experiments we used a 8 x 2.3Hz Xeon HT CPU server.

7. Acknowledgements

This work was partially supported by the CapDigital cluster under the SAMAR project and by OSEO under the Quaero program. The authors also wish to thank Nizar Habash and Saab Mansour for many helpful discussions.

8. References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation*, pages 65–72, Ann Arbor, Michigan.

Tim Buckwalter. 2004. *Buckwalter Arabic Morphological Analyzer Version 2.0*. LDC2004L02.

Mona Diab. 2009. Second generation tools (amira 2.0): Fast and robust tokenization, POS tagging, and base phrase chunking. In Khalid Choukri and Bente Maegaard, editors, *Proc. of the Second International Conference on Arabic Language Resources and Tools*, Cairo, Egypt, April. The MEDAR Consortium.

Anas El Isbihani, Shahram Khadivi, Oliver Bender, and Hermann Ney. 2006. Morpho-syntactic Arabic preprocessing for Arabic to English statistical machine translation. In *Human Language Technology Conf. / North American Chapter of the ACL Annual Meeting (HLT-NAACL), Workshop on SMT*, pages 15–22, New York. Association for Computational Linguistics.

Souhir Gahbiche-Braham, Hélène Bonneau-Maynard, and François Yvon. 2011. Two ways to use a noisy parallel news corpus for improving statistical machine translation. In *Proceedings of the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web*, pages 44–51, Portland, Oregon, June. Association for Computational Linguistics.

Qin Gao and Stephan Vogel. 2008. Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, SETQA-NLP '08, pages 49–57.

Jianfeng Gao, Galen Andrew, Mark Johnson, and Kristina Toutanova. 2007. A comparative study of parameter estimation methods for statistical natural language processing. In *Proc. of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 824–831, Prague, Czech Republic, June. Association for Computational Linguistics.

Jesús Giménez and Lluís Màrquez. 2004. SVMtool: A general POS tagger generator based on support vector

machines. In *In Proc. of the 4th International Conference on Language Resources and Evaluation*, pages 43–46.

Nizar Habash and Owen Rambow. 2005. Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In *Proc. of the 43rd Annual Meeting on ACL*, ACL '05, pages 573–580, Stroudsburg, PA, USA. Association for Computational Linguistics.

Nizar Habash and Fatiha Sadat. 2006. Arabic preprocessing schemes for statistical machine translation. In *Proc. of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, NAACL-Short '06, pages 49–52, USA. ACL.

Nizar Habash, Owen Rambow, and Ryan Roth. 2010. Mada+tokan manual. Technical Report CCLS-07-01, Columbia University.

Nizar Habash. 2010. *Introduction to Arabic Natural Language Processing*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Ahmed El Kholy and Nizar Habash. 2012. Orthographic and morphological processing for english-arabic statistical machine translation. *Machine Translation*, 26(1-2):25–45.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Morristown, NJ, USA. Association for Computational Linguistics.

Seth Kulick. 2010. Simultaneous tokenization and part-of-speech tagging for Arabic without a morphological analyzer. In *Proc. of the ACL 2010 Conference Short Papers*, ACLShort '10, pages 342–347, Stroudsburg, PA, USA. Association for Computational Linguistics.

Seth Kulick. 2011. Exploiting separation of closed-class categories for arabic tokenization and part-of-speech tagging. *ACM Transactions on Asian Language Information Processing (TALIP)*, 10:4.

Thomas Lavergne, Olivier Cappé, and François Yvon. 2010. Practical very large scale CRFs. In *Proc. the 48th ACL*, pages 504–513.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, Hubert Jin, and Wigdan Mekki. 2005a. *Arabic Treebank: Part 3 (full corpus) v 2.0 (MPG + Syntactic Analysis)*. Number LDC2005T20. Linguistic Data Consortium.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, Hubert Jin, and Wigdan Mekki. 2005b. *Arabic Treebank: Part 4 v 1.0 (MPG Annotation)*. Number LDC2005T30. Linguistic Data Consortium.

Mohamed Maamouri, Dave Graff, Basma Bouziri, Sondos Krouna, Ann Bies, and Seth Kulick. 2010. *LDC Standard Arabic Morphological Analyzer (SAMA) Version 3.1*. Number LDC2010L01. Linguistic Data Consortium.

Saab Mansour, Khalil Sima'an, and Yoad Winter. 2007. Smoothing a lexicon-based pos tagger for arabic and hebrew. In *Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources*, Semitic '07, pages 97–103, Stroudsburg, PA, USA. Association for Computational Linguistics.

Saab Mansour. 2010. Morphtagger: HMM-based Arabic segmentation for statistical machine translation. In *International Workshop on Spoken Language Translation*, pages 321–327, Paris, France, December.

Erwin Marsi, Antal van den Bosch, and Abdelhadi Soudi. 2005. Memory-based morphological analysis generation and part-of-speech tagging of Arabic. In *Proc. of the ACL 2005 workshop on computational approaches to Semitic languages*, pages 1–8, Ann Arbor, USA.

ThuyLinh Nguyen and Stephan Vogel. 2008. Context-based Arabic morphological analysis for machine translation. In *Proc. of the 12th Conference on Computational Natural Language Learning*, CoNLL '08, pages 135–142, Stroudsburg, PA, USA. ACL.

Owen Rambow Nizar Habash and Ryan Roth. 2009. Mada+tokan: A toolkit for Arabic tokenization, diacritization, morphological disambiguation, POS tagging, stemming and lemmatization. In Khalid Choukri and Bente Maegaard, editors, *Proc. of the 2nd International Conference on Arabic Language Resources and Tools*, Cairo, Egypt, April. The MEDAR Consortium.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the conference of the Association for Machine Translation in the America (AMTA)*, pages 223–231.

Andreas Stolcke. 2002. SRILM - An Extensible Language Modeling Toolkit. In *Proceedings International Conference on Spoken Language Processing*, pages 901–904.