RC24981 (W1004-083) April 21, 2010
Computer Science

IBM Research Report

ArabicWord Segmentation for Better Unit of Analysis

Yassine Benajiba
Columbia University
New York, NY

Imed Zitouni
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich

Arabic Word Segmentation for Better Unit of Analysis

Yassine Benajiba, Imed Zitouni

Columbia University, IBM T. J. Watson Research Center
ybenajiba@ccls.columbia.edu, izitouni@us.ibm.com

Abstract

The Arabic language has a very rich morphology where a word is composed of zero or more *prefixes*, a *stem* and zero or more *suffixes*. This makes Arabic data sparse compared to other languages, such as English, and consequently word segmentation becomes very important for many Natural Language Processing tasks that deal with the Arabic language. We present in this paper two segmentation schemes that are morphological segmentation and Arabic TreeBank segmentation and we show their impact on an important natural language processing task that is mention detection. Experiments on Arabic TreeBank corpus show 98.1% accuracy on morphological segmentation and 99.4% on morphological segmentation. We also discuss the importance of segmenting the text; experiments show up to 6F points improvement of the mention detection system performance when morphological segmentation is used instead of not segmenting the text. Obtained results also show up to 3F points improvement is achieved when the appropriate segmentation style is used.

1. Introduction

Due to its Semitic origins the Arabic language uses *derivation*, *inflection* and *enclitization* to form the words. Derivation is used to form a certain meaning using the root[1] and a template. For instance, the word مكتوب (mktwb — written)[2] is derived from the root كتب (ktb — write) using the template مفعول (mfEwl). By using inflection, one is able to obtain, for instance, the feminine form of this word, i.e. مكتوبة (mktwbp — written), or its plural form, i.e. مكتوبون (written — mktwbwn). Enclitization, consists of adding prefixes and suffixes to the words in order to obtain further meaning. For instance, if we wanted to express "*and written*", then we want to add the conjunction as a the prefix و (and — w) to the word *written* which we have introduced previously and thus we would obtain ومكتوب (wmktwb). It is, however, important to mention that the real world data exhibit much more complicated cases using more than one prefix and having both suffixes added during inflection and enclitization.

By adopting such a strategy to form the words, the Arabic textual data has been proved to suffer from what statisticians and scientists in the Natural Language Processing (NLP) community call "high data sparseness". From a machine learning perspective, that implies that it is not possible to achieve a good training if we do not use a huge amount of training data (Benajiba et al., 2008; Zitouni and Florian, 2008). For other tasks, such as Information Retrieval (IR) (Larkey et al., 2002; Benajiba et al., 2007), high data sparseness causes very low recall.

In the literature, many NLP scientists who have been confronted to the "high data sparseness" of the Arabic data problem have presented different solutions to "segment" each Arabic word into its different component in order to lower its sparseness and achieve a better performance (Benajiba et al., 2009; Zitouni et al., 2005; Diab et al., 2004; Diab et al., 2007).

(Darwish, 2002) uses a rule based approach for segmenting Arabic text into a sequence of prefix, stem/root and suffix. He makes the assumption that an Arabic word can have only one prefix. A manually built list of roots is used in order to change all the words with their root form. The approach also resorts to a list of prefixes and suffixes. (Habash and Rambow, 2005) propose a system named MADA that relies on the output of BAMA (Buckwalter, 2005) to render the appropriate full morphological features for all words in Modern Standard Arabic (MSA) text. MADA learns the different 10 features, namely: basic POS tag (15 tags), presence of a conjunction, presence of a particle, presence of a pronoun, presence of a determiner, gender, number, person, voice and aspect. The features are learned independently using SVM based learning. MADA also disambiguates the most probable analysis.

In this paper we use a Weighted Finite State Transducer (WFST) based segmentations system. Our goal is not only to show the obtained performance when WFST is used but we also carry out a study between two different segmentation schemes and give an analysis of the type of errors we obtain when one or another scheme is used. We also present in this paper the impact of segmentation style on the Arabic Mention Detection task.

The remainder of this paper is organized as follows. Section 2. describes the different segmentation schemes of Arabic text we use. We describe the adopted approach to build the segmentation model in Section 3. and we report segmentation performance and results in Section 4.. Section 5. shows how segmentation impact mention detection system performance. Finally, we draw a conclusions in Section 6.

2. Arabic Segmentation Schemes

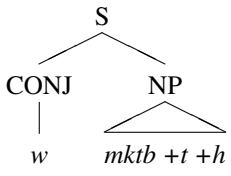
The most widely adopted segmentation schemes for natural language processing tasks are:

- **Morphological Segmentation :** aims at segmenting all affixes of a word. Thus, all the prefixes and suffixes which are attached to the stem are separated. The

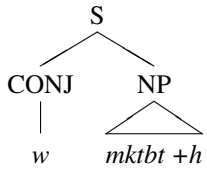
[1] In Arabic, a word root is three or four consonant words. These consonants are called the *radicals*.

[2] Throughout the paper, for each Arabic example we show between parenthesis its transliteration and English translation separated by "—".

morphological segmentation for the example mentioned earlier ومكتبته could be: و + مكتب +ت +ه (w+ mktb +t +h). As one may notice here, the suffix ة (*t* — feminine marker) is separated from the word. The parse tree of this word after full segmentation is as follows:



- **Arabic Treebank (ATB) segmentation :** this is a *light* segmentation adopted to build parse trees in the Arabic TreeBank (ATB) corpus (Maamouri et al., 2004). This type of segmentation considers splitting the word into affixes if and only if it projects an independent phrasal constituent in the parse tree. As an example, in the word ومكتبته (wmktbth — and his library) mentioned earlier, the phrasal independent constituents are: (i) conjunction و (*w* — and); (ii) noun and the head of a Noun Phrase (NP) مكتبة (*mktbt* — library); and (iii) a pronoun (PRON) ه (h — his). This would lead to the following parse tree:



As stated earlier, a full segmentation (i.e., morphological segmentation) will separate the suffix ة (*t* — feminine marker) from the word مكتبة (*mktbt* — library). Since the ة (and generally all the suffixes which are gender marks) are not independent constituents as shown in the previous parse tree, they are not considered for ATB segmentation. Thus, the ATB segmentation scheme considers splitting only a subset of prefixes and suffixes from the stem. When using ATB segmentation, the number of words is similar to its counter part in English. This is one reason why ATB segmentation is widely used in building machine translation systems for the English-Arabic language pair. For the word ومكتبته (wmktbth — and his library), the ATB segmentation would be و + مكتبت +ه (w+ mktbt +h). Prefixes that are considered for possible segmentation are[3]:

- 1: "ل", (*l* — to);
- 2: "ب" (*b* — in);
- 3: "و" (*w* — and); and
- 4: "ك" (*k* — as).

[3]we put between parenthesis the transliteration and potential equivalent English translation.

Possible segmented suffixes are the possessive personal pronouns such as:

- "ي" (*y* — my);
- "هم" (*hm* — their);
- "كم" (*km* — yours); etc.

In our study we build a segmentation tool for each of the two segmentation schemes which we have just presented. In order to do so, we train a finite state transducer (see Section 3.) for both segmentation schemes. We remind the reader that the main goal of our study is to give an error analysis for both segmentation models and to show how does the choice of the segmentation scheme impact the forthcoming NLP tools which are plugged at the output of the segmenter.

3. ATB and Morphological Segmentation Models

Both ATB and morphological segmentation systems are based on *weighted finite state transducers* (WFST) as described by (Mohri et al., 2002). The segmentation process consists of separating the Arabic normal white-space delimited words into (hypothesized) prefixes, stems, and suffixes, which become the subject of analysis (tokens). The decoder implements a general Bellman dynamic programming search for the best path on a lattice of segmentation hypotheses that match the input characters (Lee et al., 2003). The model was initially trained from a small corpus of hand segmented examples from Arabic Treebank Corpus (Maamouri et al., 2005) and then refined using unsupervised learning on a larger corpus of 155 million words (Graff, 2003).

3.1. Arabic Word Segmentation: Algorithm

Lets $W = \{w_1, w_2, \ldots w_Q\}$ denotes an original text of Arabic white-space delimited words to segment, and $S_k = \{s_1^k, s_2^k, \ldots s_{L_k}^k\}$ denotes one of the many possible sequences of L_k tokens (segments) obtained by choosing for each word in W one of its possible segmentations into prefix(es), stem, and suffix(es). All the possible segmentations for each word w_i can be obtained from a lookup table of all the prefixes and suffixes. Since not all words have a prefix and/or a suffix, and since words can have multiple prefixes and/or suffixes, the term L_k is necessarily greater than Q ($L_k \geq Q$). Among all the possible segmentations S_k of the input document, we chose the one $\hat{S}$ that has the highest probability:

$$\hat{S} = arg\max_k P(S_k) \quad (1)$$

The probability $P(S_k)$ is estimated using an n-gram language model on segment (morpheme) sequences:

$$P(S_k) \simeq \prod_{i=1}^{L} P(s_i^k | s_{i-1}^k, s_{i-2}^k, \ldots s_{i-n+1}^k) \quad (2)$$

The n-gram language models can be estimated in different ways, we use in particular a Kneser-Ney based back-off trigram language model as described by (Chen and Goodman, 1998).

As an example, the Arabic white-space delimited word "وفي" has two different correct segmentations: "و + في" and "وفي". In the first segmentation (i.e., "و + في"), we find one prefix "و", one stem "في", and no suffixes. Based on this segmentation, the word has the meaning of "*and in*", such as in the fragment example "وفي هذا المنزل" (meaning, "*and in this house*"). In the second segmentation (i.e., "وفي"), we find no prefixes, one stem, and no suffixes. In this case, the word show the meaning "*faithful*" such as in the example "رجل وفي", meaning "*faithful man*". Another segmentation is to split the word "وفي" into one prefix, one stem, and one suffix: "و + ف + ي". However, this segmentation has likelihood close to zero and consequently it will be discarded. For a sentence that contains the word "رجل وفي" ("*faithful man*"), the segmentation process first extracts possible segmentations into prefix(es), stem, and suffix(es): "رجل + وفي", "رجل + و + في", and "رجل + و + ف + ي". Notice that the word "رجل" (man) has only one possible segmentation, because its characters are not part of our lookup table of prefixes and suffixes. Once the possible segmentations are defined, a segment n-gram language model is used to define the segmentation that has the highest probability. As and example, the probability of the segment phrase "رجل + و + ف + ي" is almost null. On the other hand, based on the context, the probability of the segment phrase "رجل + وفي" is higher than the probability of the segment phrase "رجل + و + في". Consequently, the result of our segmentation decoder on the sentence "رجل وفي" is the segment phrase "رجل + وفي".

In our implementation of the segmentation algorithm, we have recast the segmentation strategy as the composition of three distinct finite state machines. The first machine encodes the prefix and suffix expansion rules, producing a lattice of possible segmentations. The second machine is a dictionary that accepts characters and produces identifiers corresponding to dictionary entries. The final machine is a trigram language model, specifically a Kneser-Ney based back-off language model (Chen and Goodman, 1998). Differing from (Lee et al., 2003), we have also introduced an explicit model for unknown words based upon a character unigram model, although this model is dominated by an empirically chosen unknown word penalty.

3.2. Arabic Word Segmentation: Bootstrapping

In addition to the segmentation model based upon a dictionary of stems and words, we also experimented with models based upon character n-grams. For these models, both Arabic characters and spaces, and the inserted prefix and suffix markers appear on the arcs of the finite state machine. The language model is conditioned to insert prefix and suffix markers based upon the frequency of their appearance in n-gram character contexts that appear in the training data. An analysis of the errors indicated that the character based model is more effective at segmenting words that do not appear in the training data. We then decided to exploit this ability to improve the dictionary based model. As in (Lee et al., 2003), we used unsupervised training data, which is automatically segmented, to discover previously unseen stems. In our case, the character n-gram model is used to segment a portion of the Arabic Gigaword (AG) corpus (Graff, 2003). Thereafter, we create a vocabulary of stems and affixes by requiring tokens that appear more than twice in the supervised training data or more than ten times in the unsupervised segmented corpus.

	No.	Percentage
Correct	42380	99.4%
Incorrect	274	0.6%

Table 1: ATB Segmentation Results

	No.	Percentage
Correct	41850	98.1%
Incorrect	804	1.9%

Table 2: Morph Segmentation Results

4. Segmentation Results

4.1. Evaluation

An experimental evaluation of the accuracy of the FST based segmentation models was performed for Arabic. A training set consisting of $571,743$ words extracted from Arabic Treebank 1, 2 and 3. The test set contains $42,591$ words.

To facilitate future comparisons with work presented here, and to simulate a realistic scenario, the splits are created based on article dates: the test data is selected as the latest 5% of the data in chronological order, in each of the covered genres (newswire and weblog) and corpora (ATB 1, 2 and 3). The time span of the test set is intentionally non-overlapping, and posterior to that of the training set within each data source, as this models how the system will perform in the real world. The evaluation was done by comparing the resulting segmentation with test sets segmented by human annotators. The accuracy is computed as the percentage of words with a final segmentation that is in agreement with the one provided by the human annotator.

Tables 1 and 2 show the obtained results for both ATB and morphological segmentation schemes.

4.2. Error Analysis

Tables 1 and 2 show the performance of ATB and morphological segmentation. It is important to note that it is not appropriate to compare the performance of those two segmentation schemes because they do not perform same style of segmentation. The morphological segmentation model deals with a much greater number of prefixes and suffixes (see Section 2.) which renders the task much harder. This explains why error rate of the ATB segmentation model is much lower than the morphological segmentation model.

Our error analysis also showed that the major part of the incorrectly segmented words can be classified in two categories:

1- Ambiguous words: These are words which are polysemous and accept different segmentations. For such words, the segmentation models relied mainly on the lexical context to disambiguate. However, the context does not always provide enough information for a correct disambiguation which results in incorrect segmentations. Some of these examples are the following:

- فان (polysemous — fAn): meaning either *so it*, or *mortal* where in the first case it should be segmented as "f +An" and in the second case as "fAn".
- بعيد (polysemous — bEyd): meaning either *in holiday* or *far* where the former case should be segmented as "b +Eyd" and the second as "bEyd".
- الا (polysemous — AlA): meaning either *so that no* resulting from merging "An" and "lA" or *except* where the first case should be segmented as "A +lA" and the second as "AlA".

2- OOVs: The second major category consists of Out-Of-Vocabulary words (OOVs). In such cases, the segmentation system proceeds to segment a word which has never been seen in the training phase. Some of these cases are the following:

- بتيس(Batice — btys): is a proper noun, both segmentation systems have segmented the first character (*b*) as the prefix "in". Other cases of proper names starting with "b" are: بخيت (Bekhit — bxyt) and بشور (Bashour — bA$wr).
- فيليني (Felini — fylyny) and كالاس (Kalas, kAlAs) have also been incorrectly segmented by both models for confusing the first character as the prefixes ف (and — f) and ك (like — k), respectively.
- والبكم (and the def people — wAlbkm), where both segmentation models separated the two last characters, i.e. كم (km), as the pronoun suffix "you".

5. Impact on Mention Detection

In order to study the impact of using different segmentation schemes we have chosen a sequence classification task: Arabic Mention Detection. Mention Detection (MD) task consists of the detection and classification of all the named, nominal and pronominal entity mentions within a text and classifies them (Zitouni and Florian, 2008). We adopt here the ACE 2007 (NIST, 2007) nomenclature. This task is similar to the Named Entity Recognition (NER) task with the additional twist of also identifying nominal and pronominal mentions. Mention detection represents one of the crucial steps in the information extraction processing pipeline, as identifying the participants in a discourse is essential to the understanding of the text: it is the first step in determining *who* did *what where*. Its applications are wide spread, from information extraction and template filling, to search and information retrieval, to machine translation and data mining.

Similarly to classical NLP tasks such as base noun phrase chunking (Ramshaw and Marcus, 1994), text chunking (Ramshaw and Marcus, 1995) or named entity recognition (Tjong Kim Sang, 2002), we formulate the mention detection problem as a classification problem, by assigning to each segment (token) in the text a label, indicating whether it starts a specific mention, is inside a specific mention, or is outside any mentions. Those segments are the result of running the segmentation models on the input raw Arabic text. The segment becomes the unit of analysis when doing classification where it can be a morph segment, an ATB segment or also a word when no segmentation is conducted. When segmentation is performed, the unit of analysis (i.e., segment) is a prefix, a stem or a suffix.

Good performance in many natural language processing tasks has been shown to depend heavily on integrating many sources of information (Florian et al., 2004; Benajiba and Zitouni, 2009). Given this goal, one can use algorithms that can easily integrate and make effective use of diverse input types. The Maximum Entropy (MaxEnt henceforth) classifier is a good choice. It integrates arbitrary types of information and make a classification decision by aggregating all information available for a given classification, but the reader can replace it with his/her favorite feature-based classifier.

In order to validate the impact of segmentation on MD system performance, we tested on systems that employ different feature sets:

1. **Lex**$_f$ - lexical features: system that has access to n-grams spanning the current segment; both preceding and following it. A number of n equal to 3 turned out to be a good choice.
2. **Stem**$_f$ - Lex_f + morphological features: system that has access to lexical features and morphological features computed as stem trigram spanning the current stem; both preceding and following it (Zitouni et al., 2005).
3. **Synt**$_f$ - $Stem_f$ + syntactic features; system that has access to lexical and morphological features as well as POS tags and shallow parsing information in a window of 2 segments.

5.1. Data

Experiments are conducted on the Arabic ACE 2007 data [4] (NIST, 2007). There are 379 Arabic documents and almost $98,000$ words. We find 7 types of mentions in ACE'07 data:

- Facility: FAC;
- Geopolitical Entity: GPE;
- Location: LOC;

[4] Enclitic pronouns are not annotated in ACE-2007.

- Organization: ORG;
- Person: PER;
- Vehicle: VEH; and
- Weapon: WEA.

Since the evaluation tests set are not publicly available, we have split the publicly available *training* corpus into an 85%/15% data split. We use 323 documents (80,000 words) for training and 56 documents (18,000 words) as a test set. This results in 17,634 mentions (7,816 named, 8,831 nominal and 987 pronominal) for training and 3,566 for test (1,673 named, 1,682 nominal and 211 pronominal). To facilitate future comparisons with work presented here, and to simulate a realistic scenario, the splits are created based on article dates: the test data is selected as the latest 15% of the data in chronological order, in each of the covered genres (newswire and weblog). The time span of the test set is intentionally non-overlapping, and posterior to that of the training set within each data source, as this models how the system will perform in the real world.

While performance on the ACE data is usually evaluated using a special-purpose measure - the ACE value metric (NIST, 2007), given that we are interested in the mention detection task only, we decided to use the more intuitive and popular (un-weighted) F-measure, the harmonic mean of precision and recall.

5.2. Results

Tables 3 and 4 show the obtained results when we have used the ATB (ATB_s) and the morphological segmentation ($Morph_s$) schemes, respectively. We show the results per mention class for the different feature sets which we have introduced earlier.

These results also show that classifiers trained on $Morph_s$ have better performance than similar ones trained on ATB_s. We believe that this is due to the fact that $Morph_s$ is less sparse than ATB_s. At the same time, our analysis has shown that when using ATB_s, the classifier has access to a greater context, and it showed to perform better on long span mentions.

When using full morphological segmentation, the data is less sparse, which leads to less Out-Of-Vocabulary tokens (OOVs): the number of OOVs in the $Morph_s$ data is 1,518 whereas it is 2,464 in the ATB_s. As an example, the word الرهينة (Alrhynp — the hostage), which is person mention in the training data. This word is kept unchanged after ATB segmentation and is segmented to "ال + رهين +ة" (Al+ rhyn +p) in $Morph_s$. In the development set the same word appears in its dual form without definite article, i.e. رهينتين. This word is unchanged in ATB_s and is segmented to "رهين +ت +ين" (rhyn +p +yn) in $Morph_s$. For the model built on ATB_s, this word is an OOV, whereas for the model built on $Morph_s$ the stem has been seen as part of a person mention and consequently has a better chance to tag it correctly. These phenomena are frequent, which makes the classifier trained on $Morph_s$ more robust for such cases. Also, we observed that models trained on ATB_s perform better on long span mentions. A representative example of a frequent case would be the organization named mention:

"هيئة العلماء المسلمين"
(hy'T AlElmA' Almslmyn — Association of the Muslim Scholars)

which is kept unchanged in the ATB_s and appears in the $Morph_s$ as:

"هيء +ة ال +علماء ال +مسلم +ين"
(hy' +T Al +ElmA' Al +mslm +yn)

Across the board, the results show that although ATB_s might help to capture long span mentions. $Morph_s$ is much more adequate to model sequence classification problems, such as Mention Detection, as it helps to obtain up to 3 F-measure points of improvement.

Table 5 shows the obtained overall F-measure when the data is not segmented.

Lex_f	$Stem_f$	$Synt_f$
66.4	66.6	69.0

Table 5: Results in terms of F-measure when the data is not segmented

Results in Table 5 show that it is very hard to learn a classification task when the Arabic data is not segmented: we show more that 6F points decrease in MD system performance when compared to results on text that is morphologically segmented. By contrasting these results with the ones shown in Tables 3 and 4 it is possible to see the error-rate induced by the "high data sparseness" problem which we have described in Section 1.

6. Conclusions

We addressed in this paper an important component for Arabic NLP systems, i.e. text segmentation. Arabic text is sparse and a segmentation of words into zero or more prefixes, a stem and zero or more suffixes is necessary to achieve good performance when building an NLP system. Two of the most used segmentation schemes in the literature are ATB and morphological segmentation which are motivated by very different linguistic reasons. The former one states that only segmentations which would result in independent phrasal constituents are necessary. The latter one, however, aims at segmenting each and every morphological component of a word.

Both ATB and morphological segmentation models presented in this paper are trained using Weighted Finite State Transducer on the same corpus (Arabic TreeBank Part 1,2,3 corpus from LDC). Results show state-of-the-art performance of 1.8% and 0.6% error rate for the morphological and ATB segmentation model, respectively. However, it is important to remember that the Morphological and ATB models deal with different segmentation styles where the former one has to deal with higher number of affixes (larger search). The major part of the errors for both segmentation

	Lex_f	$Stem_f$	$Synt_f$
All	*70.1*	*69.8*	*72.1*
PER	68.9	68.1	71.5
ORG	63.2	63.5	63.7
GPE	81.0	81.4	83.4
FAC	40.0	42.3	45.4
LOC	57.9	56.2	51.8
VEH	43.1	36.0	35.3
WEA	50.0	48.6	54.0

Table 3: Results in terms of F-measure per feature-set and mention type using ATB segmentation scheme

	Lex_f	$Stem_f$	$Synt_f$
All	*74.1*	*74.5*	*75.5*
PER	72.8	74.1	75.3
ORG	64.8	64.0	65.1
GPE	85.5	85.3	85.8
FAC	49.0	50.7	52.9
LOC	59.8	57.5	58.0
VEH	60.0	58.8	61.2
WEA	70.0	70.0	73.7

Table 4: Results in terms of F-measure per feature-set and mention type using morphological segmentation scheme

models are encountered in words that are ambiguous and the lexical context provided was not enough.

To measure the impact of the segmentation scheme choice on Arabic NLP applications, we trained a MD system on text that is: (i) morphologically segmented; (ii) ATB segmented; and (iii) not segmented (only punctuation separated from the words). Experiments show that using morphological segmentation lead to better performance (75.3) than using the ATB one (71.5) even though the ATB segmentation model has lower error rate. In order to validate the effectiveness of our results we conducted experiments using different levels of richness for the MD system feature-set. Across the board, results show that MD model built over morphological segmentation always obtains a better performance.

Habash et al. in (Habash and Sadat, 2006) experimented different segmentation schemes for Arabic-to-English Machine Translation task. They show that best results were obtained when an ATB-like segmentation was used. This was expected since the number of segments when doing ATB segmentation is close to the number of words in the English translation. Also, putting the results reported in (Habash and Sadat, 2006) together with the ones we present in this paper, we show with empirical proof that the obtained performance for most Arabic NLP tasks depends on the segmentation scheme. Hence, we believe that when building an Arabic NLP system one should first investigate the best segmentation scheme.

As future work we plan to combine morphological and ATB segmentation and test their affect on MD performance. We also plan to test the effectiveness of using segmentation when larger resources are available for the Arabic MD system.

7. References

Y. Benajiba and I. Zitouni. 2009. Morphology-based segmentation combination for arabic mention detection. *Special Issue on Arabic Natural Language Processing of ACM Transactions on Asian Language Information Processing (TALIP)*, 8(4).

Y. Benajiba, P. Rosso, and J.M. Gomez. 2007. Adapting jirs passage retrieval system to the arabic. In *CICLing'07*, pages 530–541.

Y. Benajiba, M. Diab, and P. Rosso. 2008. Arabic named entity recognition using optimized feature sets. In *Proceedings of EMNLP'08*, pages 284–293.

Y. Benajiba, M. Diab, and P. Rosso. 2009. Arabic named entity recognition: A feature-driven study. *In the special issue on Processing Morphologically Rich Languages of the IEEE Transaction on Audio, Speech and Language*.

T. Buckwalter. 2005. Buckwalter arabic morphological analyzer. http://www.ldc.upenn.edu/. LDC Catalog number LDC2002L49.

S. F. Chen and J. Goodman. 1998. An empirical study of smoothing techinques for language modeling. Technical Report TR-10-98, Center for Research in Computing Technology, Harvard University, Cambridge, Massachusettes, August.

K. Darwish. 2002. Building a shallow arabic morphological analyser in one day. In *ACL02 Workshop on Computational Approaches to Semitic Languages*.

M. Diab, K. Hacioglu, and D. Jurafsky. 2004. Automatic tagging of arabic text: from raw text to base phrase chunks. In *HLT/NAACL*.

M. Diab, K. Hacioglu, and D. Jurafsky, 2007. *Arabic Computational Morphology: Knowledge-based and Empirical Methods*, chapter 9. Springer.

R. Florian, H. Hassan, A. Ittycheriah, H. Jing, N. Kambhatla, X. Luo, N Nicolov, and S Roukos. 2004. A statistical model for multilingual entity detection and tracking. In *Proceedings of HLT-NAACL 2004*, pages 1–8.

D. Graff. 2003. Arabic gigaword. http://www.ldc.upenn.edu/, June. LDC Catalog number LDC2003T12.

N. Habash and O. Rambow. 2005. Arabic Tokenization, Part-of-Speech Tagging and Morphological Disambiguation in One Fell Swoop. In *Proc. of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 573–580, Ann Arbor, Michigan, June. Association for Computational Linguistics.

N. Habash and F. Sadat. 2006. Combination of arabic preprocessing schemes for statistical machine translation. In *Proceedings of ACL'06*, pages 1–8, Sydney, Australia.

L.S. Larkey, L. Ballesteros, and M.E. Connell. 2002. mproving stemming for arabic information retrieval: light stemming and co-occurrence analysis. In *SIGIR'02*, pages 275–282.

Y.-S. Lee, K. Papineni, S. Roukos, O. Emam, and H. Hassan. 2003. Language model based Arabic word segmentation. In *Proceedings of the ACL'03*, pages 399–406.

M. Maamouri, A. Bies, T. Buckwalter, and W. Mekki. 2004. The penn arabic treebank: Building a large-scale annotated arabic corpus. In *Proceedings of NEMLAR Conference on Arabic Language Resources and Tools*, Cairo, Egypt.

M. Maamouri, A. Bies, T. Buckwalter, and H. Jin. 2005. Arabic treebank: Part 1 v 3.0. http://www.ldc.upenn.edu/, February. LDC Catalog number LDC2005T02.

M. Mohri, F. Pereira, and M. Riley. 2002. Weighted finite-state transducers in speech recognition. *Computer Speech and Language*, 16(1):69–88.

NIST. 2007. The ACE evaluation plan. www.nist.gov/speech/tests/ace/index.htm.

L. Ramshaw and M. Marcus. 1994. Exploring the statistical derivation of transformational rule sequences for part-of-speech tagging. In *The Balancing Act: Proceedings of the ACL Workshop on Combining Symbolic and Statistical Approaches to Language*, pages 128–135, New Mexico State University, July.

L. Ramshaw and M. Marcus. 1995. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora, ACL*.

E. F. Tjong Kim Sang. 2002. Introduction to the conll-2002 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2002*, pages 155–158. Taipei, Taiwan.

I. Zitouni and R. Florian. 2008. Mention detection crossing the language barrier. In *Proceedings of EMNLP'08*, Honolulu, Hawaii, October.

I. Zitouni, J. Sorensen, X. Luo, and R. Florian. 2005. The impact of morphological stemming on arabic mention detection and coreference resolution. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, pages 63–70, Ann Arbor, June.