The Impact of Arabic Morphological Segmentation on Broad-coverage English-to-Arabic Statistical Machine Translation

Hassan Al-Haj
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213
hhaj@cs.cmu.edu

Alon Lavie
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213
alavie@cs.cmu.edu

Abstract

Morphologically rich languages pose a challenge for statistical machine translation (SMT). This challenge is magnified when translating into a morphologically rich language. In this work we address this challenge in the framework of a broad-coverage English-to-Arabic phrase based statistical machine translation (PBSMT). We explore the full spectrum of Arabic segmentation schemes ranging from full word form to fully segmented forms and examine the effects on system performance. Our results show a difference of 2.61 BLEU points between the best and worst segmentation schemes indicating that the choice of the segmentation scheme has a significant effect on the performance of a PBSMT system in a large data scenario. We also show that a simple segmentation scheme can perform as good as the best and more complicated segmentation scheme. We also report results on a wide set of techniques for recombining the segmented Arabic output.

1 Introduction

Morphologically rich languages pose a challenge for statistical machine translation (SMT), as these languages possess a large set of morphological features producing a large number of rich surface forms. This increase in surface forms leads to larger vocabularies and higher sparsity adversely affecting the performance of SMT systems. The effect of these factors are even magnified when translating into a morphologically rich language.

In this work we address the challenge posed by the morphological richness of Arabic in the framework of a broad coverage English-to-Arabic statistical phrase based machine translation (PBSMT). We explore a full spectrum of Arabic segmentation schemes ranging from full word form to fully segmented forms separating every possible Arabic clitic and examine the effects on system performance.

The segmentation schemes, are applied in a preprocessing step to both the Arabic side of the training data and the test sets. Nine different broad coverage PBSMT systems are trained on the NIST09 Constrained Training Condition Resources (NIST09) data segmented using these various schemes. The built PBSMT systems are evaluated and compared on English-to-Arabic test sets that we construct from existing NIST09 Arabic-to-English test sets.

Based on this comparison we identify the best and the worst segmentation schemes and lay out a set of general observations on the effect of splitting off different sets of clitics (affixes) on the performance of a broad coverage PBSMT system.

As the Arabic output of systems is segmented it needs to be recombined (detokenized). We experiment with six different detokenization techniques increasing in the level of complexity. The best re tokeninzation technique is used in recombining the output of the different systems.

Previous works that addressed the effect of Arabic rich morphology and tokenization on SMT concentrated on Arabic-to-English machine translation (Habash and Sadat, 2006; Zollmann, 2006; Lee, 2004).

However, few works focused on SMT into Arabic. Sarikaya and Deng (2007) use joint morphological-lexical language models to rerank the output English-dialectal Arabic MT. A research more relevant to our work was done by Badr et. al (2008) . In their work they compare a segmented Englsih-to-Arabic system with an unsegmented system. They also experiment with a number of detokenization techniques. However, in their work they just compare a single segmentation scheme (and one variation of it) to the unsegmented baseline without mentioning what motivated the choice of this specific segmentation. They also use a training corpora of 3M words and conclude that the effect of segmentation diminishes when the corpora size is "large". In our work we experiment with a spectrum of segmentation giving a total of nine different schemes. Our results indicate that the choice of the segmentation scheme has a significant effect on the performance of a PBSMT system in a large data scenario. In this work we also explore more variation of detokenization techniques for recombining the Arabic output.

The remainder of the paper is organized as follows: In Section 2 we present some relevant background on Arabic linguistics to motivate the Arabic preprocessing schemes discussed in Section 3. All the different detokenization schemes are described in Section 4. The training and test data used is described in Section 5, while Section 6 described the experiments and result for all the different segmentation schemes.

2 Arabic Morphology and Orthography

Arabic is a morphologically rich language with a large set of morphological features[1] that are realized using both concatenative (affixes and stems) and templatic (root and patterns) morphology.

Arabic has a set of attachable clitics (affixes) to be distinguished from inflectional features such as gender, number, person, voice, aspect, etc. These clitics attach to the word increasing the ambiguity of alternative readings. Arabic clitics apply to a word base in a strict order:

CONJ+ PART+ DET+ **WORD_BASE** *+PRON*

Table 1 lists the Arabic clitics[2] divided into 4 classes: conjunction proclitics (*CONJ+),* particle proclitics (*PART+*), definite article (*DET+*) , and pronominal enclitics (*PRON+*) which comprise of possessive and object pronouns. The first three classes of clitic in Table 1 are given along with their English meaning. The clitics of the fourth class (*PRON)* are given followed by O (for object pronoun) or P (possessive pronoun) followed by their morphological features: person, gender, and number in the this order.

Arabic orthography introduces further challenges as certain letters in Arabic script are often spelled inconsistently which leads to an increase in both sparsity (multiple forms of the same word) and ambiguity (same form corresponding to multiple words). One example is the letter Alif in Arabic, which can appear with Hamza on top أ, or below إ, and with maddah on top آ. All these forms are often written as bare Alif ا. Another example is the two letters Ya ي and Alif Maqsura ى which are often used interchangeably in word final position. Add to all this the optionality of diacritics (short vowels) in Arabic script.

This inconsistent variation in raw Arabic text is typically addressed using orthographic normalization which maps all Alif to to bare Alif, Dotless Ya/Alif Maqsura form to Dotted Ya and deletes diacritics.

CONJ	w+ (*and*), f+ (*then*)
PART	l+ (*to/for*), b+ (*by/with*), k+ (*as/such*) s+ *will/future*.
DET	Al+(*the*)
PRON	+h (+O:3MS, +P:3MS) +hA (+O:3FS,+P:3FS) +hm (+O:3MP,+P:3MP) +hmA (+O:3D,+P:3D) +hn (+O:3FP, +P:3FP) +k (+O:2FS,+P:2FS,+O:2MS,+P:2MS) +km (+O:2MP,+P:2MP) +kmA (+O:2D,+P:2D) +kn (+O:2FP,+P:2FP) +nA (+O:1P,+P:1P) +y (+O:1S,+P:1S)

Table 1. Arabic clitics divided to 4 classes.

[1] Arabic words have fourteen morphological features: POS, person, number, gender, voice, aspect, determiner proclitic, conjunctive proclitic, particle proclitic, pronominal enclitic, nominal case, nunation, idafa (possessed), and mood.

[2] Arabic transliterations are provided in Buckwalter transliteration scheme (Buckwalter, 2002).

Input	wbAlnsbp lAyTAlyA fAnh yEny AnhA sttSrf kdwlp Sgyrp ttxlY En ms&wlyAthA
Gloss	and regarding to italy this means that it will act as a country small giving up its responsibilities
English	And regarding Italy, this mean that it will act as a small country giving up its responsibilities
UT	wbAlnsbp l<yTAlyA f>nh yEny >nhA sttSrf kdwlp Sgyrp ttxlY En ms&wlyAthA
S0	w+ bAlnsbp l<yTAlyA f>nh yEny >nhA sttSrf kdwlp Sgyrp ttxlY En ms&wlyAthA
S1	w+ bAlnsbp l<yTAlyA f+ >nh yEny >nhA sttSrf kdwlp Sgyrp ttxlY En ms&wlyAthA
S2	w+ b+ Alnsbp l+ <yTAlyA f+ >nh yEny >nhA s+ ttSrf k+ dwlp Sgyrp ttxlY En ms&wlyAthA
S3	w+ b+ Alnsbp l+ <yTAlyA f+ >n +O:3MS yEny >n +O:3FS sttSrf k+ dwlp Sgyrp ttxlY En ms&wlyAt +P:3FS
S4	w+ b+ Alnsbp l+ <yTAlyA f+ >n +O:3MS yEny >n +O:3FS s+ ttSrf k+ dwlp Sgyrp ttxlY En ms&wlyAt +P:3FS
S5	w+ b+ Al+ nsbp l+ <yTAlyA f+ >n +O:3MS yEny >n +O:3FS s+ ttSrf k+ dwlp Sgyrp ttxlY En ms&wlyAt +P:3FS
S5SF	w+ b+ Al+ nsbp l+ <yTAlyA f+ >n +h yEny >n +hA s+ ttSrf k+ dwlp Sgyrp ttxlY En ms&wlyAt +hA

Table 2. The different tokenization schemes exemplified on the same sentence.

This type of Arabic text "reduction" could be acceptable when Arabic is the source language, but is clearly problematic when translating into Arabic. Therefore, we use the "enriched" form of the Arabic raw text throughout this work. The enriched form of text uses the correct form of Alif ا and the right form of Ya ي and Alif Maqsura ى in word final position.

3 Arabic Preprocessing Schemes

We experiment with various Arabic preprocessing schemes by splitting off different subsets of the clitics mentioned in Section 2. The raw Arabic text is enriched and tokenized using the Morphological Analysis and Disambiguation for Arabic (MADA) toolkit (Habash and Rambow, 2005). The various Arabic tokenization schemes that we experiment with, span a segmentation spectrum ranging from coarse segmentation, which uses unsegmented text, to fine segmentation which splits off all possible clitics.

All the different tokenization schemes are described in detail below from coarse to fine :

- **UT:** This scheme uses the full (untokenized) enriched form of the word. This scheme is used as input to produce the other schemes.
- **S0**: This scheme splits off the conjunction proclitic w+.
- **S1**: This scheme splits off +f in addition to the w+ split by **S0**.
- **S2:** This scheme splits off all the particle proclitics (*PART)* in addition to the clitics split off by S1
- **S3**: This scheme splits off all clitics from the *CONJ* class and all clitics of PART class except s+ prefix. It also splits off all the suffixes from the *PRON* class. This scheme is equivalent to the *PATB* tokenization, but to distinguish between the possessive and object pronouns ,which have the same surface form, we use their morphological features (henceforth, MF form) instead as given in Table 1 between parentheses .
- **S4:** This scheme splits off all clitics split by S3 plus splitting of the s+ clitic. This scheme is equivalent to the *ATBv3* tokenization. As in S3 we also use the MF form of the *PRON* clitics.
- **S5**: This scheme splits off all the possible clitics appearing in Table 1.

We also experiment with a number of variations of these schemes:

- **S4SF**: Similar to scheme S4 but with the PRON clitics in their surface form.
- **S5SF:** Similar to scheme S5 but with the PRON clitics in their surface form. This scheme is similar to the main segmentation scheme suggested by Badr et. al (2008).

Table 2 exemplifies the effect of all the different schemes on the same sentence in the training data. As can be seen from the example in Table 2 the text's fragmentation degree increases as we move from coarse to fine tokenization. This increased fragmentation, as we will see in Section 4, enhances the complexity of recombining the tokens of the Arabic output. However, this also has a positive effect, as it decreases the vocabulary (word types), which results in lower out-of-vocabulary counts on a held out test set. For each tokenization scheme, Table 3 shows the number of tokens and

types of the Arabic side of the training data, and the OOV on a held-out set.

The held-out set comprises of 728 sentences and 18277 unsegmented words from the NIST MT02 test set .

S	Token#	Type #	OOV#
UT	136,280,410	653,584	85
S0	145,826,275	566,024	76
S1	146,162,567	552,150	76
S2	154,974,999	475,335	68
S3	160,194,619	425,645	62
S4	160,599,031	418,832	62
S5	199,179,300	391,190	59

Table 3. tokens, and types count of the Arabic side of the training data for the different schemes and the out-of-vocabulary tokens on NIST MT02 test set.

4 Arabic Automatic Detokenization

The Arabic output produced by all MT systems trained using schemes **S0-S5, S5SF, S5SF** is segmented and need to be recombined in order to produce the final Arabic text. We call the process of recombining the Arabic output as *detokenization.*

4.1 Challenges of Arabic Detokenization

Arabic detokenization is far from being a simple concatenation of the tokens, as several morphological adjustments, driven by morpho-phonological rules, apply to the tokens when they are combined. The first three rows of Table 4. include examples of such morphological adjustments.

Rule	Example
l+ Al+ → ll+	*l+ Al+ >wlad → ll>wlad* "for the kids"
p+ pron → t+pron	*lEbp +hm → lEbthm "their game"*
Y+ pron → A+pron	*rmY +h → rmAh "threw him/it"*

Table 4. Examples of morphological adjustments that govern the process of Arabic detokenization.

Another challenging aspect of Arabic detokenization is that in some cases it could be ambiguous i.e. tokens could be combined into more than one grammatically correct form. Examples of Arabic detokenization ambiguity are given in Table 5. The first column in Table 5 gives the token sequence while the second column lists all the possible combined forms for this sequence. Each possible combined from is followed by the probability, computed over the training data, of this word being the combined form of the given token sequence appearing in the training data. The second line of Table 5 demonstrated that the combined form corresponding to the sequence token could depend on the morphological case of the word base. In this case the word base *>bnA* "sons" is a noun which could have three cases: nominative, accusative, genitive.

Tokens sequence	Possible combinations
ftyAn +nA	*ftyAnA (0.88) "our boys"* *ftyAnnA (0.12) "our boys"*
>bnA' +hA	*>bnA&hA (0.22) "her sons" ,(. nom)* *>bnA'hA (0.1) "her sons", (.acc)* *>bnA}hA (0.68) "her sons", (.gen)*

Table 5. Examples of ambiguity in Arabic detonization.

4.2 Detokenization techniques

We experiment with six different detokenization techniques with increasing complexity:

- **C:** This is the most trivial technique which just concatenates the tokens of the segmented form together.
- **R:** This technique uses manually defined morphological adjustments rules to combine the Arabic tokens. Examples of such rules are given in Table 4.
- **T:** Uses a table derived from the Arabic side of training data to map the segmented form of the word to its original enriched form. If a segmented word has more than one original form then it is mapped to the most frequent one. A segmented word that does not appear in the table will be mapped to the output as is. For example, in Table 5. the segmented word *>bnA' +hA* is associated with 3 original forms in training data with different frequencies (normalized to probabilities). According to the T technique, it will be mapped to *>bnA}hA* as it is the form with the highest probability.
- **T+C:** Similar to the **T** technique but backs off to the **C** method when encountering an unknown token sequence.

- **T+R**: Similar to the **T** technique but backs off to the **R** method when encountering an unknown token sequence.
- **T+LM+R**: In addition to the table used by **T+R,** this technique also uses a (5-gram language model trained on the full enriched form. The full enriched form of the tokenized (source) sentence is determined by applying Viterbi decoding on it using both the probabilities in the table and the language model probabilities on the target side (enriched full form). This was implemented using the **disambig** utility available within the SRILM toolkit (Stolcke, 2002).

For evaluating the detokenization schemes described above, a test set of 50k (~1.3M words) sentences were randomly selected from the Arabic training corpora. The rest of the Arabic corpora was used to train the tables for the last four detokinzation techniques. The 5-gram language models used by the T+LM+R technique was trained using the whole training data.

Table 6. lists the percentage of sentence error rate (SER), of the six detokinzation techniques for all Arabic tokenizations schemes that we experiment with.

Tok.	C	R	T	T+C	T+R	T+LM+R
S0	3.30	3.37	1.07	0.41	0.48	0.49
S1	4.41	4.48	1.32	0.55	0.60	0.60
S2	36.66	11.30	2.28	1.10	1.09	1. 10
S3	50.26	23.93	3.00	1.76	1.59	1.47
S4	50.59	24.51	3.21	1.94	1.77	1.64
S5	53.52	30.04	3.73	2.40	2.25	1.99
S4SF	50.59	24.51	3.20	1.96	1.79	1.65

Table 6. SER for different tokenization scheme using the six different detokenization scheme.

A general theme that we notice by looking at Table 6. is that the SER increases as we move from coarse to fine tokenization scheme: The more fragmented the text the harder it is to recombine.

Going from left to right over the results in Table 6, we notice that the SER drops with the increase in the complexity of the detokeniztion technique. However, this drop in SER diminishes as we move up the complexity ladder. The extremely high SER of the **C** technique demonstrates that detokenization is far from being a simple concatenation of the tokens. From the **R** column we see that introducing morphological adjustments rules gives a significant improvement over the simple concatenation especially for the fine segmentations. An additional significant improvement in SER is achieved, when using tables learned from the data as in the **T** technique. In an analysis of the output of the **R** technique we found that some of the combination errors are caused by tokenization errors introduced by the morphological analyzer[3]. These kind of errors are fixed using the **T** method, which demonstrates the advantageous ability of the **T** method to successfully cope with errors introduced by the morphological analyzer.

Additional improvement in SER is obtained when backing off to the **C** method, as can be seen from the **T+C** column in Table 6. Backing off to **R** gives minor improvement over backing off to **C**. Furthermore, using a language model in the detokinization process, as in the **T+LM+R,** gives a very small improvement over the T+R technique. This very small improvement in SER comes at a costly price of 9X increase in detoknization time besides having to load the LM into memory (>1GB). For these reasons we use the (T+R) method for detokinizing the output of our SMT systems during evaluation in the Section 6.

5 Training and Testing Data

We use the NIST09 Constrained Training Condition (NIST09) Resources[4] to train and test the broad-coverage English-to-Arabic phrase based statistical machine translation systems.

5.1 Training Data

The Arabic-English parallel training data available within the NIST09 resources consists of about 5 million sentence pairs with about 150 million and 172 million words on the Arabic and English side respectively. The English side of the training corpora was first tokenized using the Stanford English tokenizer[5] then lower cased. The Arabic side was

[3] We use MADA+TOKAN version 2.32., the most recent release of MADA.

[4] http://www.itl.nist.gov/iad/mig/tests/mt/2009/MT09_ConstrainedResources.pdf

[5] The main reason for this preprocessing step is that in future works the best system build here will be extended with syntactic information based on parsing the training data using the Stanford parser.

enriched and the different tokenization generated using the Morphological Analysis and Disambiguation for Arabic (MADA) toolkit (Habash and Rambow, 2005). The parallel training corpora was then filtered by first removing sentence pairs longer than 99 on either side then deleting unbalanced sentence pairs with ratio more than a 4-to-1 in either direction.

After preprocessing and filtering, the parallel corpora consisted of 4,867,675 sentence pairs with 152 million words on the English side. The Arabic side of the training corpora is used to train nine 5-gram language models for the different tokenization schemes using the SRILM toolkit (Stolcke, 2002). An additional two 7-gram language models were trained for the S3-S5 tokenization schemes in order to account for the increase in length of the segmented Arabic. Tokens and type counts of the processed Arabic training corpora, for the different tokenization schemes, is given in Table 3.

The processed and filtered parallel corpora was then aligned using MGIZA++ (Gao and Vogel, 2008); an extended and optimized multi-threaded version of GIZA++ (Och and Ney, 2003). The Moses toolkit (Koehn et. al, 2007) is then used to symmetrize the alignement using the *grow-diag-final-and* heuristic and to extract phrases with maximum length of 7. A distortion model lexically conditioned on both the Arabic phrase and English phrase is then trained.

5.2 Tuning and testing sets

We use existing Arabic-to-English test sets available within the NIST09 resources to construct our English-to-Arabic tuning and test sets. As all NIST09 test sets were intended for use in Arabic-to-English machine translation, each Arabic source sentences is associated with four English references. From such a test sets, we construct an English-to-Arabic test set by pairing the Arabic source with the first English reference giving us single reference test set. We use 728 sentences from NIST09 MT02 test to construct a tuning set while the whole MT03, MT04, and MT05 are used to construct the test sets. Both Source and target sides of the tuning and test sets were preprocessed as described above. Table 7 includes information about all these test sets (after preprocessing), including number of sentences and tokens, and division of sentences according to their genres.

6 Results

We test and compare the performance of nine PBSMT systems trained using the different tokenization schemes. The systems use the translation, reordering and language models described in Section 5.

	#Sentences	#Tokens	Genres
MT02	728	18277	Newswire
MT03	663	16369	Newswire
MT04	1353	35870	707 Newswire 646Speech/editorial
MT05	1056	28399	Newswire

Table 7. Number of sentences, unsegmented tokens and genres of the tuning and test sets we use.

The decoding weights for these components were optimized for Bleu-4 (Papineni et al., 2002) on the MT02 tuning set using an implementation of the Minimum Error Rate Training procedure (Och, 2003). We use the Moses (Koehn et. al, 2007) decoder with a distortion window of 6 is to decode the systems on the MT03, MT04, and MT05 test sets. As discussed in 4.2 we use the **T+R** detokenization technique to recombine the Arabic tokens of the different segmentation schemes. The evaluation results reported are all on the detokenized output of systems evaluated against unsegmented enriched single reference test sets.

We report the results on all test sets using a number of evaluation metrics including BLEU-4, TER 5 (Snover and Dorr, 2006), and METEOR[6] (Lavie and Denkowski, 2008). Table 8 lists the translation results of all the systems on MT03 using all the evaluation metrics discussed earlier. Table 9 shows the results on the MT04 test set while the results on MT05 test set are given in Table 10.

All statement below about the difference in BLEU score were tested for Statistical significance using paired bootstrap resampling (Koehn, 2004) with 95% confidence interval. Looking at the results we see that across all test sets S0/S4 perform

.

[6] METEOR v1.0, HTER (English version)

best (highlighted with **Bold)** while S2/S5SF (highlighted with *Italic*) performs the worst. The performance of all the other segmentation schemes fall between these two ends.

System	BLEU	TER	METEOR
UT	35.66	50.76	51.21
S0	**36.25**	**50.98**	**51.60**
S1	35.74	51.47	50.98
S2	*35.05*	*53.16*	*49.81*
S3	36.19	50.49	51.75
S4	**36.22**	**50.61**	**51.58**
S5	34.93	51.77	49.96
S4SF	35.83	50.88	51.48
S5SF	*33.64*	*52.73*	*48.90*
S4,7gram	35.81	50.92	51.26
S5,7gram	34.84	51.88	50.10

Table 8. BLEU, TER, and METEOR for all the systems on the MT03 test set.

System	BLEU	TER	METEOR
UT	31.53	56.15	45.55
S0	**31.80**	**56.26**	**45.87**
S1	31.46	57.08	45.17
S2	*29.89*	*59.49*	*44.03*
S3	31.73	56.25	45.81
S4	**31.90**	**55.86**	**45.90**
S5	30.87	57.56	44.52
S4SF	31.99	55.90	45.84
S5SF	*30.06*	*57.83*	*43.67*
S4,7gram	31.46	56.04	45.60
S5,7gram	30.91	57.31	44.47

Table 9. BLEU, TER, and METEOR for all the systems on the MT04 test set.

System	BLEU	TER	METEOR
UT	38.40	47.94	53.96
S0	**38.83**	**48.42**	**54.13**
S1	38.29	48.84	53.40
S2	*37.29*	*51.00*	*52.72*
S3	38.55	48.22	54.33
S4	**38.55**	**48.01**	**54.21**
S5	37.72	49.65	52.94
S4SF	38.15	48.28	54.01
S5SF	*36.80*	*49.91*	*52.00*
S4,7gram	38.32	48.19	54.07
S5,7gram	37.72	49.23	52.81

Table 10. BLEU, TER, and METEOR for all the systems on the MT05 test set.

The difference in translation scores between S0 and S5SF (the main scheme used in Badr et. al 2008) on the MT03 test set is 2.61 BLEU, -1.75 TER and 2.70 METEOR points indicating that the choice of the segmentation scheme has a significant effect on the performance of PBSMT systems in a large data scenario.

The results also show that a simple segmentation scheme S0 which just splits off the w+ (*and*) can perform as good as the best and more complicated S4 (ATBv3 equivalent) scheme. The simplicity of S0 gives it advantage over the S4 as it can be both generated and recombined with lower error rate in the tokenization and the detokenization processes respectively as described in Section 4.

Comparing the scores of different schemes across all test sets we are also able to come up with the following observation:

- **S1** outperforms **S2** on all test sets which indicates that splitting off the **particle proclitics** (*PART+*) can **hurt the performance**.
- **S4** outperforms **S2** on all test sets indicating that splitting off **pronominal enclitics** (*PRON+*) has a **positive effect** on the performance of the system
- **S4** outperforms **S5** on all test sets indicating that splitting off the **definite article** Al+ **hurts the performance**.
- **S3** and **S4** perform about the same on all test sets indicating that splitting off the s+ (*will*) clititc has no significant effect on the performance of the system.
- Comparing **S4** with **S4SF** and **S5** with **S5SF** we see that using **morphological features** could only **benefit the system.**
- Comparing S4 with S4.7gram and S5 with S5,7gram on all test sets indicates that using higher order (>5) n-grams for highly fragmented schemes has no significant effect on the performance of the system.

7 Conclusions and Future Work

In this work we investigated the impact of Arabic morphological segmentation on the performance of a broad-coverage English-to-Arabic SMT system. We explored the full spectrum of Arabic segmentation schemes ranging from full word form to fully

segmented forms and examined the effects on system performance. Our results show a difference of 2.61 BLEU points between the best and worst segmentation schemes indicating that the choice of the segmentation scheme has a significant effect on the performance of a PBSMT system in a large data scenario. We also show that a simple segmentation scheme which just splits off the w+ (*and*) can perform as good as the best and more complicated (ATBv3) segmentation scheme. We also report results on a wide set of techniques for combining the segmented Arabic output.

In future work we intend to conduct an in depth analysis on the effect of segmentation scheme on the different components that make up the PBSMT system, including word alignment, the extracted phrase table, and the trained language models. We also plan to explore whether current findings extend to English-to-Arabic syntax-based and hierarchical SMT systems.

Acknowledgments

The work described in this article was supported by NSF grant IIS-0915327 and by the International Fulbright Science & Technology Award for Outstanding Foreign Students (Fulbright S&T).

References

Tim Buckwalter. 2002. *Buckwalter Arabic Morphological Analyzer*. Linguistic Data Consortium. (LDC2002L49).

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

Ibrahim Badr, Rabih Zbib, and James Glass. 2008. *Segmentation for english-to-arabic statistical machine translation*. In Proceedings of ACL-08: HLT, Short Papers, pages 153–156, Columbus, Ohio, June. Association for Computational Linguistics.

Andreas Stolcke. 2002. SRILM - an Extensible Lan - guage Modeling Toolkit. In Proceedings of the Inter national Conference on Spoken Language Processing (ICSLP), volume 2, pages 901–904, Denver,CO.

Qin Gao, Stephan Vogel, "*Parallel Implementations of Word Alignment Tool*", Software Engineering, Testing, and Quality Assurance for Natural Language Processing, pp. 49-57, June, 2008.

Franz Josef Och, Hermann Ney. *A Systematic Comparison of Various Statistical Alignment Models*, Computational Linguistics, volume 29, number 1, pp. 19-51 March 2003.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, Evan Herbst, *Moses: Open Source Tool kit for Statistical Machine Translation*, Annual Meeting of the Association for Computational Lin guistics (ACL), demonstration session, Prague, Czech Republic, June 2007.

F. Och. 2003. *Minimum Error Rate Training in Statistical Machine Translation.*In Proceedings of ACL, pages 160-167.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. *BLEU: a Method for Automatic Evaluation of Machine Translation*. In Pro ceedings of the 40th Annual Meeting of the As sociation for Computational Linguistics, pages 311–318,Philadelphia, PA

Alon Lavie and Michael Denkowski, *The METEOR Metric for Automatic Evaluation of Machine Translation*. Machine Translation Journal, 23(2-3). 2009. Pages 105-115. DOI 10.1007/s10590-009-9059-4.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Lin nea Micciulla, and John Makhoul. 2006. *A Study of Translation Edit Rate with Targeted Human Annotation*. In Proceedings of the 7th Conference of the Association for Machine Translation in the Americas (AMTA-2006), pages 223–231, Cambridge, MA, August.

Nizar Habash and Fatiha Sadat. 2006. *Arabic Preprocessing Schemes for Statistical Machine Translation.* In Proceedings of the 7th Meeting of the North American Chapter of the Association for Computational Linguistics/Human Language Technologies Conference (HLT-NAACL06), pages 49–52, New York, NY.

Young-Suk Lee. 2004. *Morphological analysis for statistical machine translation*. In Proceedings of the 5th Meeting of the North American Chapter of the Association for Computational Linguitics/Human Language Technologies Conference (HLT NAACL04), pages 57–60, Boston, MA.

Andreas Zollmann, Ashish Venugopal and Stephan Vogel. 2006. *Bridging the inflection morphology gap for Arabic statistical machine translaion.* In Short Papers in the Proceedings of the Human Language Technology and North American Association for Computatioal Linguistics Confrence(HLT/NAACL), New York. June 4-9.

Philipp Koehn. 2004. *Statistical significance tests for machine translation evaluation*. In Proceedings of the Empirical Methods in Natural Language Processing Conference (EMNLP'04), Bar-

celona, Spain.
Ruhi Sarikaya and Yonggang Deng 2007. *Joint Morphological-Lexical Language Modeling for Machine Translation*. In Proceeding of NAACL HLT.