Mach Translat (2012) 26:3–24
DOI 10.1007/s10590-011-9101-1

The impact of Arabic morphological segmentation on broad-coverage English-to-Arabic statistical machine translation

Hassan Al-Haj · Alon Lavie

Received: 5 July 2010 / Accepted: 4 August 2011 / Published online: 22 September 2011

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 largest-to-date set 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 2.31 BLEU points averaged over all test sets between the best and worst segmentation schemes indicating that the choice of the segmentation scheme has a significant effect on the performance of an English-to-Arabic PBSMT system in a large data scenario. We show that a simple segmentation scheme can perform as well as the best and more complicated segmentation scheme. An in-depth analysis on the effect of segmentation choices on the components of a PBSMT system reveals that text fragmentation has a negative effect on the perplexity of the language models and that aggressive segmentation can significantly increase the size of the phrase table and the uncertainty in choosing the candidate translation phrases during decoding. An investigation conducted on the output of the different systems, reveals the complementary nature of the output and the great potential in combining them.

Keywords Arabic machine translation · Arabic segmentation · Arabic detokenization · English to Arabic translation

H. Al-Haj (✉) · A. Lavie
Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA
e-mail: hhaj@cs.cmu.edu

A. Lavie
e-mail: alavie@cs.cmu.edu

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 sparsness, adversely affecting the performance of SMT systems. The effects of these factors are 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 the largest-to-date set of Arabic segmentation schemes ranging from full word forms to fully segmented forms separating every possible Arabic clitic, and we examine the effect on system performance. We conduct an in-depth analysis on the effect of segmentation choices on the different components that make up the PBSMT system, including the language model and the extracted phrase table. We also assess the variation of the Arabic translation output across the different segmentation schemes.

The segmentation schemes are applied in a preprocessing step to both the Arabic side of the training data and the test sets. Twelve 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 of different sets of clitics (affixes) on the performance of a broad coverage PBSMT system. We also experiment with six different detokenization techniques, of increasing level of complexity, for recombining the segmented Arabic output.

We then conduct an in-depth analysis on the effect of segmentation on the different components of the PBSMT system by comparing the systems' components along various features defined in this work. We also investigate the variation across the output of the systems trained using the different segmentation schemes.

Previous work that addressed the effect of Arabic rich morphology and tokenization on SMT concentrated on Arabic-to-English machine translation (Lee 2004; Sadat and Habash 2006; Zollmann et al. 2006). However, few works focused on SMT into Arabic. Sarikaya and Deng (2007) use joint morphological-lexical language models to rerank the output of an English-dialectal Arabic MT system. Research more relevant to our work was done by Badr et al. (2008). In their work they compare a segmented English-to-Arabic system with an unsegmented system. They also experiment with a number of detokenization techniques. A more recent work, following the steps of Badr et al. (2008), was done by El Kholy and Habash (2010a). In their work they experiment with Arabic-side normalization and segmentation, and introduce three additional segmentation schemes. They show that their best segmentation scheme outperforms the best segmentation proposed by Badr et al. (2008).

In contrast with previous works that apply segmentation schemes previously proposed for Arabic-to-English machine translation, we explore the largest-to-date set of Arabic segmentations. Starting from a full word form, we gradually peal off affixes,

creating 12 different segmentations. While some of these segmentation schemes were introduced before, other segmentations have not been used in any previous work.

Furthermore, previous works applied their Arabic segmentation to a small data scenario of at most 4.5 million words, extrapolating their conclusion to larger data scenarios. In this work we investigate the effect of Arabic segmentation in the framework of a broad coverage translation system with at least 150M words used as training data. We reveal that in the broad-coverage scenario segmentation schemes exhibit a different behavior from what has been shown previously for a small data scenario. Simple segmentation that lagged behind under small data scenario can perform as well as the best and more complicated segmentation scheme. Furthermore, our results demonstrate that the choice of segmentation scheme still has a significant effect on the performance of the PBSMT system in a large data scenario, in contrast to the diminishing effect predicted in previous works.

Finally, while previous works based their conclusion just on the comparison of the final scores of the different systems, we conduct a deeper investigation and compare the components that make up these systems, providing insight on the reasons behind the differences in the performance of the systems.

The remainder of the paper is organized as follows: In Sect. 2 we present some relevant background on Arabic linguistics to motivate the Arabic preprocessing schemes discussed in Sect. 3. All the different detokenization schemes are described in Sect. 4. The training and test data used is described in Sect. 5, while Sect. 6 describes the experiments and results for all the different segmentation schemes. In Sect. 7 we conduct an analysis on the components making up the different translation systems and investigate the variation in their output. Finally, conclusions and future work are described in Sect. 8.

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:

$$CONJ + PART + DET + \textbf{WORD\_BASE} + PRON$$

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 clitics in Table 1 are given along with their English meaning. The clitics of

[1] In Arabic words have the following fourteen morphological features: part of speech, person, number, gender, voice, aspect, determiner proclitic, conjunctive proclitic, particle proclitic, pronominal enclitic, nominal case, nunation, idafa (possessed), and mood (Sadat and Habash 2006).

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

Table 1 Arabic clitics divided to four classes

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)

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 (Habash and Rambow 2005).

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. Added to all this is 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 bare Alif, Dotless Ya/Alif Maqsura form to Dotted Ya and deletes diacritics.

El Kholy and Habash 2010a called this type of orthographic normalization of Arabic text "reduction". This reduction may 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. According to El Kholy and Habash 2010a terminology, 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 while omitting all diacritics.

3 Arabic preprocessing schemes

We experiment with various Arabic preprocessing schemes by splitting of different subsets of the clitics mentioned in Sect. 2. The raw Arabic text is enriched and tokenized using the Morphological Analysis and Disambiguation for Arabic (MADA) toolkit (Habash and Rambow 2005; Habash 2007).[3] The various Arabic tokenization schemes that we experiment with range from coarse segmentation, which uses unsegmented text, to fine segmentation which splits off all possible clitics.

[3] We use MADA + TOKAN version 2.32. which was the most recent release of MADA when this work was done.

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

- **UT:** This scheme uses the full (un-tokenized) enriched form of the word (ST in Habash and Sadat 2006). This scheme is used as input to produce the other schemes.
- **S0:** This scheme splits off the conjunction proclitic w+ (WA in Habash and Sadat 2006).
- **S1:** This scheme splits off +f in addition to the w+ split by S0 (D1 in MADA).
- **S2:** This scheme splits off all the particle proclitics (*PART*+) in addition to the clitics split off by S1 (D2 in MADA).
- **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 Penn Arabic Treebank (PATB; Maamouri et al. 2004) 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.
- **S0PR:** This scheme splits off all suffixes from the (+*PRON*) class in addition to the w+ prefix split off by S0. The MF forms of the (+*PRON*) clitics are used here.
- **S4:** This scheme splits off all clitics split by S3 plus splitting off the s+ clitic. This scheme is equivalent to the Arabic Treebank: Part 3 v3.2 (*ATBv3.2*) tokenization. The MF forms of the (+*PRON*) clitics are used here.
- **S5:** This scheme splits off all the possible clitics appearing in Table 1. The MF form of the (+*PRON*) clitics are used here (D3 in MADA).

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).
- **S5SFT:** Similar to scheme S5 but with the prefixes concatenated together into one prefix. This scheme is similar to the best scheme suggested by Badr et al. (2008).
- **S3T:** Similar to scheme S3 but with the prefixes concatenated together into one prefixes.

Table 2 exemplifies the effect of all the different schemes on the same sentence from the training data.

As can be seen from the example in Table 2 the text's fragmentation increases as we move from coarse to fine tokenization. This increased fragmentation, as we will see in Sect. 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 18,277 unsegmented words from the NIST MT02 test set.

Table 2 Some of the different tokenization schemes exemplified on the same sentence

	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
S0PR	w+ bAlnsbp l<yTAlyA f>n +O:3MS yEny >n +O:3FS sttSrf kdwlp Sgyrp ttxlY En ms&wlyAt +P:3FS
S3	w+ b+ Alnsbp l+ <yTAlyA f+ >n +O:3MS yEny >n +O:3FS sttSrf k+ dwlp Sgyrp ttxlY En ms&wlyAt +P:3FS
S3T	wb+ 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
S4SF	w+ b+ Alnsbp l+ <yTAlyA f+ >n +h yEny >n +hA s+ ttSrf k+ dwlp Sgyrp ttxlY En ms&wlyAt +hA
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
S5SFT	wbAl+ nsbp l+ <yTAlyA f+ >n +h yEny >n +hA s+ ttSrf k+ dwlp Sgyrp ttxlY En ms&wlyAt +hA

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

Seg.	Token #	Type #	%OOV	Seg.	Token #	Type #	%OOV
UT	136,280,410	653,584	0.46	S3T	159,891,078	425,654	0.29
S0	145,826,275	566,024	0.39	S4	160,599,031	418,832	0.29
S1	146,162,567	552,150	0.39	S4SF	160,599,031	418,819	0.29
S0PR	151,465,273	490,065	0.33	S5SF	199,164,334	391,170	0.22
S2	154,974,999	475,335	0.33	S5SFT	193,378,931	391,187	0.23
S3	160,194,619	425,645	0.29	S5	199,179,300	391,190	0.22

4 Arabic automatic detokenization

The Arabic output produced by all MT systems trained using all the schemes described in Sect. 3 except UT is segmented and needs to be recombined in order to produce the

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

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 5 Examples of ambiguity in Arabic detokeniztion

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*)

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.

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 demonstrates 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.

When a possessive pronoun suffix attaches to >bnA′ then the case of the noun is marked using three different letters &, ′, and }. However, when no suffix is present then the case marker is a diacritic appearing on the last letter of the noun >bnA′. This diacritic is omitted in the Arabic enriched form used here, which creates the ambiguity that we see in the second entry of Table 5.

4.2 Detokenization schemes

We experiment with six different detokenization techniques of 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. We use a script implementing the complete set of morphological adjustments rules as described in (El Kholy and Habash 2010b).
T: Uses a table derived from the Arabic side of the 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 three 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 backoff 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 input sentence is determined by selecting the *FullForm* which maximizes:

$$P\,(Full\ Form|Tokenized\ Form) \cdot P_{LM}\,(Full\ Form)$$

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 sentences (∼1.3M words) were randomly selected and removed from the Arabic training corpora. The remaining corpora were used to train the tables for the last four detokinzation techniques and the 5-gram language models used by the T + LM + R technique.

Table 6 lists the percentage of sentence error rate (SER) of the six detokinzation techniques for all Arabic tokenizations schemes that we experiment with. 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. We notice that the SER for the S3 and the S5SF schemes are similar to the SER of the S3T and the S5SFT schemes respectively. This is because most of the morpho-phonological rules, as discussed in Sect. 4.1 apply to the boundary of the affix and the stem when they are combined. This boundary remains the same when the prefixes are concatenated together.

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. An

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

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
S0PR	26.54	19.07	3.31	2.35	2.29	2.08
S2	36.66	11.30	2.28	1.10	1.09	1.10
S3T	50.26	23.93	3.00	1.76	1.60	1.47
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	53.45	29.92	3.56	2.20	2.04	1.81
S5SFT	53.52	30.04	3.73	2.40	2.25	2.00
S5SF	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

additional significant improvement in SER is achieved, especially on fine segmentation, 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. 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, in most of the cases , 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 a 9-fold increase in detoknization time, besides having to load the LM into memory (>1 GB). For these reasons we use the $T + R$ method for detokinizing the output of our SMT systems during evaluation in the Sect. 6.

5 Training and testing data

We use the NIST09 Constrained Training Condition (NIST09) Resources to train and test 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[4] then lower cased. The Arabic

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

side was enriched and the different tokenizations generated using the Morphological Analysis and Disambiguation for Arabic (MADA) toolkit (Habash and Rambow 2005; Habash 2007). The parallel training corpora was then filtered by first removing sentence pairs longer than 99 words on either side then deleting unbalanced sentence pairs with a ratio of more than 4-to-1 in either direction.

After preprocessing and filtering, the parallel corpora consisted of 4,867,675 sentence pairs with 152 million on the English side. The Arabic side of the training corpora is used to train twelve 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 S4 and S5 tokenization schemes in order to account for the increase in length of the segmented Arabic. Tokens and type counts of the Arabic training corpora, using 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++. The Moses toolkit (Koehn et al. 2007) is then used to symmetrize the alignment 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 set, an English-to-Arabic test or tuning set could be constructed in a number of ways. One possible way is constructing an English-to-Arabic test set by pairing each Arabic source with only one of the four English references, giving us four different single reference test sets. Alternatively, an English-to-Arabic test set could also be constructed by pairing each Arabic source sentence with all four English references resulting in a single reference test set four times larger than the test sets constructed previously.

Before deciding which of the above techniques to use in constructing the English-to-Arabic tuning set, we tested the effect of these different test set construction techniques on the overall performance of the PBSMT system. Using the techniques described above, we construct 5 different English-to-Arabic tuning sets using 728 sentences chosen from the NIST09 MT02 test set. The **UT** system is then tuned on the different tuning set and tested on an English-to-Arabic test sets constructed from the NIST MT03–MT05 test sets by pairing each Arabic source sentence with the first English reference. We report the results on the MT03–MT05 test sets using the BLEU-4 (Papineni et al. 2002) evaluation metric. All the results are given in Table 7.

UTi is the UT system tuned on a tuning set constructed from MT02 by pairing the Arabic source with the ith English reference, while UTAll is the UT system tuned on the tuning set constructed by pairing the Arabic source with all the four English references. Comparing the performance of the systems UT1–UT4, and UTAll we notice that there is no significant difference between the scores of UT1, UT3, UT4 and UTALL on MT03–MT05 while UT2 performs the worst, especially on MT04 and

Table 7 BLUE scores for all UT systems on the MT03–MT05 test sets

System	MT03	MT04	MT05
UT1	27.52	22.64	30.02
UT2	27.25	22.13	29.39
UT3	27.72	22.87	30.15
UT4	27.38	22.64	29.94
UTAll	27.79	22.70	30.39

Table 8 Number of sentences, unsegmented tokens and genres of the tuning and test sets we use

	# Sentences	# Tokens	Genre
MT02	728	18,277	Newswire
MT03	663	16,369	Newswire
MT04	1353	35,870	707 Newswire
			646 Speech/editorial
MT05	1056	28,399	Newswire

MT05. Therefore, for tuning all the systems built in this work, we use a tuning set constructed from MT02 test set by pairing each Arabic source sentence with the first English reference.

All the systems in this work are tested on the MT03–MT05 test sets used in this section. Table 8 includes information about the tuning and all the test sets, including number of sentences and tokens, and division of sentences according to their genres.

6 Results

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

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 Sect. 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 et al. 2006), and METEOR[5] (Lavie and Denkowski 2009). Table 9 lists the translation results of all the systems on MT03 using all the evaluation

[5] METEOR v1.2, language independent version.

Table 9 BLEU, TER, and METEOR scores for all the systems on the MT03 test set

System	BLEU	TER	METEOR	System	BLEU	TER	METEOR
UT	27.52	54.47	44.10	S4	**28.24**	54.08	44.59
S0	**28.34**	54.42	44.53	S5	26.68	56.05	43.08
S1	27.69	55.16	43.94	S4SF	27.73	54.36	44.34
S2	*26.98*	57.00	42.74	S5SF	*25.27*	57.14	43.08
S0PR	28.10	54.67	44.38	S5SFT	26.67	56.16	43.02
S3	**28.16**	54.05	44.63	S4,7gram	27.84	54.65	44.17
S3T	27.72	54.72	44.19	S5,7gram	26.60	55.98	43.09

Table 10 BLEU, TER, and METEOR scores for all the systems on the MT04 test set

System	BLEU	TER	METEOR	System	BLEU	TER	METEOR
UT	22.64	60.29	38.11	S4	**23.06**	60.12	38.52
S0	**22.99**	60.50	38.48	S5	22.01	62.09	37.34
S1	22.05	61.23	37.85	S4SF	23.11	60.24	38.47
S2	*21.37*	63.64	36.68	S5SF	*21.20*	62.43	36.57
S0PR	22.80	60.76	38.20	S5SFT	22.05	62.25	37.37
S3	**23.06**	60.42	38.43	S4,7gram	22.69	60.37	38.25
S3T	23.11	60.37	38.51	S5,7gram	22.10	61.74	37.33

Table 11 BLEU, TER, and METEOR scores for all the systems on the MT05 test set

System	BLEU	TER	METEOR	System	BLEU	TER	METEOR
UT	30.02	51.41	46.61	S4	**30.24**	51.54	46.86
S0	**30.37**	51.85	46.77	S5	29.22	53.55	45.73
S1	29.91	51.73	46.19	S4SF	29.91	52.51	46.62
S2	28.79	54.38	45.23	S5SF	28.30	54.06	44.88
S0PR	29.85	52.32	46.36	S5SFT	28.92	53.65	45.64
S3	**30.26**	51.53	46.86	S4,7gram	29.91	51.77	46.66
S3T	30.16	51.79	44.88	S5,7gram	29.26	53.26	45.60

metrics discussed earlier. Table 10 shows the results on the MT04 test set while the results on MT05 test set are given in Table 11.

All statements 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/S3 perform best (highlighted with **bold**, while S2/S5SF (highlighted with *italic*) perform the worst. The performance of all the other segmentation schemes falls between these two ends.

The difference in translation scores between S0 and S5SF is 2.31 BLEU, −2.28 TER and 1.75 METEOR points averaged over all test sets. This big difference in translation quality indicates that the choice of the segmentation scheme has a significant

effect on the performance of English-to-Arabic PBSMT systems in a large data scenario. The S4 (ATBv3.2) scheme outperforms S5SFT (the best scheme in Badr et al. (2008) S5SFT) by 2.25 BLEU point averaged on all test sets.

The results also show that a simple segmentation scheme S0 which just splits off the w+ (*and*) can perform as well as the best and more complicated S4 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 detokenization processes respectively, as described in Sect. 4.

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

- S1 outperforms S2 on all test sets, which indicates that splitting off the *particle proclitics* (PART+) can **hurt the performance**.
- The effect of splitting off the (PRON+) suffixes on the system depends on the prefixes that are split off. When the only prefix that is split off is w+ as in S0, splitting off the (PRON+) suffixes in S0PR causes an insignificant drop of 0.15 (no change) BLEU points on average on all test sets. However, in case the prefixes split off are the (PART+) and (CONJ+) clitics, as in S2, then splitting off the (PRON+) suffixes as in S3 causes a significant increase of 1.44 BLEU averaged on all test sets.
- 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 instead of the surface form of the suffixes can only benefit the system.
- Concatenation of the prefixes together improved the performance of S5FT scheme by a significant 1.07 BLEU points averaged on all test set, while dropping by an insignificant −0.16 (no change) BLEU points averaged on all test sets in the case of S3. This indicates that concatenating the prefixes has a positive effect on the most fragmented scheme S5SF but this effect diminishes as the scheme becomes less and less fragmented as in the case of S3.
- Comparing S4–5.7gram with S4–5 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 Systems comparison

In previous sections we described all the different segmentation schemes and their effect on the final performance of the systems. In this section we conduct an in-depth analysis on the effect of segmentation choices on the different components that make up the PBSMT system, including the language model and the extracted phrase table. We also assess the variation of the Arabic translation output across the different segmentation schemes.

7.1 Language models

The Arabic side of the training corpora for all the different tokenization schemes was used to train twelve 5-gram language models using the modified Kneser–Ney smoothing and cutoffs of 1 for orders bigger than 2. The size of the training corpora used to build the different language model is given in Table 3, Sect. 3.

The different language models are compared by computing the n-gram precession (coverage) and perplexities on the Arabic side of the MT03 test set. The n-gram precision is defined as the percentage of n-grams in the test set which appears in the language model. Table 12 lists the size of the MT03 test set and the type/token n-gram precision for all the language models trained using the different segmentation schemes. The perplexity of all the language models is evaluated on the MT03 test set and is given in Table 12.

Looking at Table 12, we notice that the more fragmented the scheme the higher is the n-gram precision. We also notice that the difference between the n-gram precision of a fine and a coarse scheme becomes more significant for higher order n-grams. This difference in n-gram precession between coarse and fine segmentations is reflected in perplexity scores on the test set. The perplexity steadily decreases from 108.682 for the UT scheme down to 33.24 for the most fragmented scheme S5. However, the n-gram precision and the perplexity were computed over tokens where the definition of a token varies across the different segmentation schemes. This variation is expressed in the different sizes of the MT03 test sets for each scheme, which makes a comparison of the language models based on n-gram precision and perplexity much less meaningful. One way to make the comparison of the different language models perplexities more meaningful is to use "normalized perplexity" (Kirchhof et al. 2006).

$$NNP(w_1, \ldots, w_M) = 2^{-\frac{1}{N}\sum_{i=1}^{M} \log(P(w_i|w_{i-1,\ldots,}w_{i-k+1}))} \quad (1)$$

The normalized perplexity of an k-gram language model on a test set of size M is given in Eq. 1. As we see in Eq. 1, the normalized perplexity differs from the regular perplexity only in the normalization factor. In the case of normalized perplexity the log likelihood of the data is averaged by dividing it by the number of the unsegmented words N in the test set, as opposed to the number of tokens in test set M. This is done in order to compensate for the effect that perplexity tends to be lower for a text containing more individual units, since the sum of log probabilities is divided by a larger denominator.

The normalized perplexities of all language models are given in the last column of Table 12. Looking at the normalized perplexities gives us a totally different picture than the one we got from comparing regular perplexities. We see that normalized perplexities **increase** as we move from coarse to fine segmentation. The most significant change in normalized perplexity occurs when moving from S4 to S5, where the normalized perplexity increases by 12.79%. As S5 differs from S4 in splitting off an additional prefix Al+ (the), this big increase in normalized perplexity indicates that splitting off the Al+ has a significant negative effect on the language model.

Table 12 Number of tokens, type/token n-gram precision, perplexity and normalized perplexity on the MT03 test for all the language models

Scheme	# MT03	1 Prec.	2 Prec.	3 Prec.	4 Prec.	5. Prec	Perp	NPerp
UT	16,369	99.04/99.68	89.56/91.77	53.88/56.96	32.97/31.07	20.12/19.00	108.682	108.682
S0	17,270	99.15/99.74	91.36/93.36	57.65/60.62	33.78/35.67	20.38/21.46	86.635	109.767
S1	17,302	99.17/99.75	91.40/93.40	57.72/60.69	33.85/35.73	20.43/21.51	86.196	110.100
S2	18,189	99.15/99.77	93.13/94.89	62.98/65.91	37.60/39.66	22.54/23.79	69.9172	110.199
S0PRON	17,883	99.16/99.76	92.33/94.23	60.58/63.51	36.28/38.16	21.79/22.88	76.2412	112.183
S3	18,719	99.12/99.78	93.78/95.46	65.30/68.19	39.77/41.86	23.99/25.26	63.332	112.415
S3T	18,700	99.12/99.78	93.75/95.35	65.13/68.00	41.65/39.59	23.87/25.12	63.608	112.449
S4	18,802	99.11/99.78	93.91/95.57	65.69/68.55	40.16/42.21	24.18/25.44	62.1369	112.243
S4SF	18,802	99.11/99.78	93.93/95.60	65.76/68.62	40.25/42.30	24.26/25.53	62.0578	112.080
S5SFT	22,389	98.99/99.81	94.98/97.01	74.97/79.08	54.41/57.78	37.65/39.74	35.376	125.158
S5SF	22,852	98.99/99.82	94.91/97.10	76.13/80.33	56.54/60.04	39.74/41.98	33.2266	126.488
S5	22,852	98.99/99.82	97.08/94.90	76.07/80.25	56.49/59.98	39.67/41.90	33.2467	126.594

The low normalized perplexities that we see in Table 12 for the **UT** and **S0** language models contributes to the fact that coarse segmentation systems can perform as good as systems built using the more complicated schemes. Furthermore, we notice that using morphological features instead of surface forms for the suffixes has no significant effect on the perplexity of the language model, as can be seen from comparing S5 to S5SF and S4 to S4SF. We also notice that the difference in normalized perplexities between the language model of S5SF and S5SFT is 1.33 points compared to the 0.034 difference between S3 and S3T. This contributes to the significant difference in the performance between the S5SF and S5SFT compared to the much smaller difference between S3 and S3T systems in Tables 9, 10, and 11.

7.2 Phrase table

The phrase table is one of the most important components of a PBSMT system. In this section we compare and analyze the differences between all the phrase tables built and trained on the various segmentation schemes defined in this work.

All the phrase tables are first filtered to the MT03 test set then contrasted according to several features:

- **Number of source phrases and Phrase pairs:** For each scheme we calculate the number of phrase pairs and source phrases. The results are given in the first two columns of Table 13.
- **Phrase Table Entropy (PTE):** Phrase Table Entropy (Koehn et al. 2009) captures the amount of uncertainty involved in choosing candidate translation phrases. For each source phrase s with a set of possible translations (target sides) in the phrase table T, the phrase entropy of s PE(s) is defined in Eq. 2. The Phrase Table Entropy is defined as the average of phrase entropy for all the source phrases in the phrase table. Table 13 gives the phrase table entropy for all schemes.

$$PE(s) = -\sum_{t \in T} P(t|s).\log P(t|s) \tag{2}$$

- **Average number of target phrases per phrase length**: The phrase table entropy provides a measure to the amount of uncertainty in choosing a translation averaged over the whole phrases in the phrase table. However, it would be very useful to zoom in on the phrase table entropy and look into the phrase table target side ambiguity for each phrase length. Therefore, we compute the average number of target phrases (ANTP) per phrase lengths of 1–7 (max phrase length). All the results are given in Table 13.

Looking at Table 13, we notice that the number of phrase pairs steadily and gradually grows when moving from the coarse UT to the fine S4SF scheme, while the number of source phrases relatively remains the same. The PTE for these segmentations does not significantly change and remains in the range 3.33–3.41. However the most significant increase in phrase table size and PTE happens when moving from S4SF to the S5 scheme and its variants S5SF and S5SFT. The size of the phrase table

Table 13 All the features calculated for the different phrase tables of the various segmentation schemes

Scheme	# Phrase pairs	# Source phrases	PTE	ANTP1	ANTP2	ANTP3	ANTP4	ANTP5	ANTP6	ANTP7
UT	15,111,038	29,678	3.411	3317.58	436.15	98.15	41.62	18.69	7.68	5.71
S0	15,575,350	29,870	3.371	3483.66	434.05	95.48	40.43	17.62	7.32	5.43
S1	15,641,938	29,849	3.372	3498.44	435.38	96.73	40.46	17.54	7.38	5.43
S2	16,180,001	29,983	3.332	3674.34	439.06	95.01	39.39	17.46	6.93	4.95
S0PRON	16,489,620	29,896	3.402	3705.43	455.44	99.93	41.41	18.15	7.20	5.57
S3	16,906,278	29,971	3.367	3847.85	455.76	98.37	40.82	17.83	7.03	5.45
S3T	16,910,558	29,949	3.364	3842.83	458.02	98.31	40.80	17.89	7.20	5.13
S4	16,937,625	29,984	3.363	3856.77	455.86	98.47	40.95	17.90	7.12	5.26
S4SF	16,923,937	30,008	3.361	3849.77	457.36	98.62	41.07	17.76	6.92	5.01
S5SFT	20,273,498	29,266	3.611	4776.88	517.70	103.68	40.26	16.63	5.59	3.88
S5SF	20,580,967	29,080	3.634	4877.90	521.23	103.68	39.42	15.86	5.34	3.82
S5	20,596,688	29,045	3.635	4883.26	520.62	103.13	39.68	16.16	5.29	3.69

increases by 21.6% relative when comparing S5 to S4, while the number of source phrases decreases by 3.31%. This significant increase in the size of the phrase table compared to a small increase in the number of the source phrases adds to the uncertainty in choosing the candidate translation phrases as can be seen by comparing the PTEs of the two systems. We see a relatively significant jump in the phrase table entropies (PTE) of S5 compared to S4. The PTE increases by 10% relative when moving from S4 to S5. A clearer explanation for this increase in PTE can be found by comparing the ANTPs of the S4 and S5 system. We notice that the ANTP of S5 is higher from the ANTP of S4 for short phrases but is lower for longer phrases. The ANTP1 of the S5 system is 26.62% higher than the ANTP1 of S4. This difference drops to 14.21% for ANTP2 and 4.73% for ANTP3. A total change in the trend occurs for ANTP4 and higher, where the ANTP of S5 becomes lower than for S4. The ANTP4 of S5 is −3.1% lower than the ANTP4 of S4, this difference increases to −9.72% for ANTP5, −34.62% for ANTP6, and −42.55% for ANTP7. This relatively high PTE, and ANTPs for S5 and its variants contribute to the fact that these segmentation are among the worst performing segmentation as seen in Sect. 6.

One reason for the significant difference in phrase table size, PTE, and ANTP between S5 and S4 (and the other schemes) can be found when looking into the set of affixes that these two schemes split off. The only difference between the S4 and S5 scheme is that the S5 scheme splits off the Al+ (*the*) in addition to all the affixes split of in S4. From the results discussed above, we conclude the splitting off the Al+ causes a significant increase in the size of the phrase table and magnifies the ambiguity and the uncertainty inherited in the target side choice in the phrase table, especially for shorter phrases.

We looked into the phrase tables of both S4 and S5 and found several cases of source phrases for which the splitting off the Al+ caused an increase in the average number of target phrases. One of the most frequent cases was source phrases with the "noun adjective" POS pattern. In Arabic, the adjective follows the noun in definiteness which is expressed by attaching the Al+ before the word. For example, the expression ***Al***$rq ***Al***>wsT (lit. "the east the middle") "the middle east", could also appear in the indefinite form as $rq > wsT (lit. " east middle") "middle east", but never in the ungrammatical form $rq Al>wsT. However, we found that when splitting of the Al+ prefix as in S5 an Arabic phrase such as $rq Al# >wsT could be extracted from the Arabic text and end up as a target phrase for the English source phrase "middle east". Such cases are frequent and increase the average number of target phrases by introducing ungrammatical target phrases that did not exist in the S4 phrase table, especially for short source phrases (<3).

7.3 Output variation

One important question which could be asked here is how different are the outputs of the PBSMT systems that were trained using the different segmentation schemes?

One way for quantifying the output variation is to find out how much gain in performance, compared to the best single system, could be achieved when performing an

oracle combination over the output of all the systems. Therefore, we conduct here an oracle study into system combination.

An oracle combination output was created by selecting for each input sentence the output of the system with the highest sentence-level METEOR score. One way for doing this oracle combination is to include in the combination the output of all the systems built in this work then to evaluate the combined output. However, it would be much more useful to divide the systems into intra-related groups in order to isolate their contribution to the performance of the final combined system. This will give us an insight into the variation of the output across the different systems groups.

We start by performing an oracle combination on the systems in the first group (G1). Then we gradually add each group to the combined systems. Table 14 lists the five system groups and the names of the systems in each group. The results of the combined systems on MT03, MT04, and MT05 are given in Tables 15, 16 and 17 respectively. The best single system (BSS) for each test set is used as a baseline.

Looking at Tables 15, 16, and 17 we notice a significant improvement in the performance of the oracle combination of all the systems (G5) over the best single system (BSS). The G5 system outperforms the BSS by 7.28 BLEU points averaged over all test sets. This great difference between the combined system and the BSS is an indication of the complementary nature of the output produced by the systems using different schemes. It also demonstrates the great potential in automatically combining the output of the different systems. These results are consistent with the results of Sadat and Habash Sadat and Habash (2006). In their work, they demonstrate, using oracle combination, the great potential in automatically combining the output of different Arabic-to-English systems which use different Arabic segmentations in a small data scenario.

Table 14 The systems in each of the five groups

Group	Systems
G1	UT, S0-S5, S0PR
G2	G1+ S4SF, S5SF
G3	G2+ S4, 7gram, S5, 7gram
G4	G3+ S3T, S5SFT
G5	G4+ UT2–4, UTALL

Table 15 Combined systems scores on MT03

The best performing system is indicated in bold, while the poorest performing system is indicated in italics

System	BLEU	TER	METEOR
BSS	*28.34*	*54.42*	*44.53*
G1	33.54	48.28	50.56
G2	33.88	47.98	51.01
G3	34.24	47.54	51.41
G4	34.49	47.33	51.65
G5	**35.51**	**46.67**	**52.48**

Table 16 Combined systems scores on MT04

System	BLEU	TER	METEOR
BSS	*23.11*	*60.37*	*38.51*
G1	27.98	54.81	44.26
G2	28.39	54.20	44.78
G3	28.70	53.72	45.18
G4	29.03	53.37	45.52
G5	**29.76**	**52.60**	**46.30**

The best performing system is indicated in bold, while the poorest performing system is indicated in italics

Table 17 Combined systems scores on MT05

System	BLEU	TER	METEOR
BSS	*30.37*	*51.85*	*46.77*
G1	36.21	45.56	53.12
G2	36.82	45.03	53.71
G3	37.29	44.68	54.13
G4	37.53	44.53	54.32
G5	**38.39**	**43.57**	**55.25**

The best performing system is indicated in bold, while the poorest performing system is indicated in italics

8 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 largest-to-date set of Arabic segmentation schemes ranging from full word forms to fully segmented forms, and we examined the effects on system performance. Our results show a difference of 2.31 BLEU points averaged over all test sets between the best and worst segmentation schemes, indicating that the choice of segmentation scheme has a significant effect on the performance of English-to-Arabic PBSMT systems in a large data scenario. We also show that a simple segmentation scheme which just splits off the w+ (*and*) can perform as well as the best and more complicated (ATBv3.2) segmentation scheme.

An in-depth analysis on the effect of segmentation choices on the components that make up a PBSMT system reveals that the normalized perplexities of the language models increase as we move from coarse to fine segmentation. The analysis also shows that aggressive segmentation such as S5, which splits of all possible affixes including Al+ (*the*) can significantly increase the size of the phrase table and the uncertainty in choosing the candidate translation phrases during decoding which has a negative effect on the machine translation quality. A significant improvement of 7.28 BLEU averaged over all test sets is achieved over the best single system in an oracle combination of the output of the different systems. This demonstrates the complementary nature of the output and the great potential in automatically combining the output of the different systems.

Following the findings in this work we plan to experiment with automatic system combination on the output of the systems built here. We also plan to explore whether

current findings extend to English-to-Arabic syntax-based and hierarchical SMT systems.

Acknowledgements The work described in this article was supported in part by NSF grant IIS-0915327 and by the International Fulbright Science & Technology Award for Outstanding Foreign Students (Fulbright S&T). Many thank to Nizar Habash for his valuable feedback on our work and to Ahmed El Kholy for providing us with the rule based detokenization script used in this work.

References

Badr I, Zbib R, Glass J (2008) Segmentation for English-to-Arabic statistical machine translation. In: Proceedings of ACL-08: HLT, Short Papers, Columbus, June, pp 153–156

Buckwalter T (2002) Buckwalter Arabic morphological analyzer. Linguistic Data Consortium. (LDC2002L49)

El Kholy A, Habash N (2010a) Orthographic and morphological processing for English-Arabic statistical machine translation. In: Proceedings of TALN 2010, Montréal, 19–23 July 2010

El Kholy A, Habash N (2010b) Techniques for Arabic morphological detokenization and orthographic denormalization. In: Proceedings of the seventh international conference on language resources and evaluation (LREC) 2010, Valletta, Malta

Gao Q, Vogel S (2008) Parallel implementations of word alignment tool. In: Software engineering, testing, and quality assurance for natural language processing, Columbus, June, pp 49–57

Habash N (2007) Arabic morphological representations for machine translation book chapter. In: van den Bosch A, Soudi A (eds) Arabic computational morphology: knowledge-based and empirical methods. Springer, Berlin

Habash N, Rambow O (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), Ann Arbor

Habash N, Sadat F (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, Barcelona

Kirchhof K, Vergyri D, Bilmes J, Duh K (2006) Andreas Stolcke morphology-based language modeling for conversational Arabic speech recognition. Comput Speech Lang 20:589–608

Koehn P (2004) Statistical significance tests for machine translation evaluation. In: Proceedings of the empirical methods in natural language processing conference (EMNLP'04), Barcelona

Koehn P, Hoang H, Birch A, Callison-Burch C, Federico M, Bertoldi N, Cowan B, Shen W, Moran C, Zens R, Dyer C, Bojar O, Constantin A, Herbst E (2007) Moses: open source toolkit for statistical machine translation. In: Annual meeting of the association for computational linguistics (ACL), demonstration session, Prague, June

Koehn P, Birch A, Steinberger R (2009) 462 machine translation systems for Europe. In: MT summit XII: proceedings of the twelfth machine translation summit, Ontario, 26–30 Aug 2009, pp 65–72

Lavie A, Denkowski M (2009) The METEOR metric for automatic evaluation of machine translation. Mach Transl J 23(2–3): 105–115. doi:10.1007/s10590-009-9059-4

Lee Y-S (2004) Morphological analysis for statistical machine translation. In: Proceedings of the 5th meeting of the North American chapter of the association for computational linguistics/human language technologies conference (HLT NAACL04), Boston, pp 57–60

Maamouri M, Bies A, Buckwalter T (2004) The Penn Arabic treebank: building a large-scale annotated Arabic corpus. In: NEMLAR conference on Arabic language resources and tools, Cairo

Och F (2003) Minimum error rate training in statistical machine translation. In: Proceedings of ACL, Sapporo, pp 160–167

Papineni K, Roukos S, Ward T, Zhu W-J (2002) BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting of the association for computational linguistics, Philadelphia, pp 311–318

Sadat F, Habash N (2006) Morphological preprocessing scheme combination for statistical MT. In: Proceedings of COLING-ACL, Sydney. HLT-NAACL06, New York, pp 49–52

Sarikaya R, Deng Y (2007) Joint morphological-lexical language modeling for machine translation. In: Proceedings of NAACL HLT 2007, Companion Volume, Rochester, NY, April 2007, pp 145–148

Snover M, Dorr B, Schwartz R, Micciulla L, Makhoul J (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), Cambridge, Aug, pp 223–231

Stolcke A (2002) SRILM—an extensible language modeling toolkit. In: Proceedings of the international conference on spoken language processing (ICSLP), vol 2, Denver, pp 901–904

Zollmann A, Venugopal A, Vogel S (2006) Bridging the inflection morphology gap for Arabic statistical machine translation. In: Short papers in the proceedings of the human language technology and North American association for computational linguistics conference (HLT/NAACL), New York, 4–9 June 2006