Techniques for Arabic Morphological Detokenization and Orthographic Denormalization

Ahmed El Kholy and Nizar Habash

Center for Computational Learning Systems, Columbia University
475 Riverside Drive New York, NY 10115
{akholy,habash}@ccls.columbia.edu

Abstract

The common wisdom in the field of Natural Language Processing (NLP) is that orthographic normalization and morphological tokenization help in many NLP applications for morphologically rich languages like Arabic. However, when Arabic is the target output, it should be properly detokenized and orthographically correct. We examine a set of six detokenization techniques over various tokenization schemes. We also compare two techniques for orthographic denormalization. We discuss the effect of detokenization and denormalization on statistical machine translation as a case study. We report on results which surpass previously published efforts.

1. Introduction

Arabic is a morphologically rich language. The common wisdom in the field of natural language processing (NLP) is that tokenization of Arabic words through decliticization and reductive orthographic normalization is helpful for many applications such as language modeling and statistical machine translation (SMT). Tokenization and normalization reduce sparsity and decrease the number of out-of-vocabulary (OOV) words. However, in order to produce proper Arabic that is orthographically correct, tokenized and orthographically normalized words should be detokenized and orthographically corrected (enriched). As an example, the output of English-to-Arabic machine translation (MT) systems is reasonably expected to be proper Arabic regardless of the preprocessing used to optimize the MT performance. Anything less is comparable to producing all lower-cased English or uncliticized and undiacritized French. Detokenization is not a simple task because there are several morphological adjustments that apply in the process. In this paper we examine different detokenization techniques for various tokenization schemes and their effect on SMT output as a case study.

This paper is divided as follows. Section 2 presents the previous related work. In Section 3, we discuss the Arabic linguistic issues and complexities that motivate the detokenization techniques explained in Section 4. Section 5 describes the various experiments we had followed by an analysis of the results.

2. Related Work

Much work has been done on Arabic-to-English MT (Habash and Sadat, 2006; Lee, 2004; Zollmann et al., 2006) mostly focusing on reducing the sparsity caused by Arabic's rich morphology. There is also a growing number of publications with Arabic as target language. In previous work on Arabic language modeling, OOV reduction was accomplished using morpheme-based models (Heintz, 2008). Diehl et al. (2009) also used morphological decomposition for Arabic language modeling for speech recognition. They described an SMT approach to detokenization (or what they call morpheme-to-word conversion). Although the implementation details are different, their solution is comparable to one of our new (but not top performing) decomposition models (T+LM). We do not compare directly to their implementation approach in this paper. Regarding English-to-Arabic MT, Sarikaya and Deng (2007) use joint morphological-lexical language models to re-rank the output English-dialectal Arabic MT; and Badr et al. (2008) report results on the value of morphological tokenization of Arabic during training and describe different techniques for detokenization of Arabic in the output. The research presented here is most closely related to that of Badr et al. (2008). We extend on their contribution and present a comparison of a larger number of tokenization schemes and detokenization techniques that yield improved results over theirs.

3. Arabic Linguistic Issues

In this section, we present relevant aspects of Arabic word orthography and morphology.

3.1. Arabic Orthography

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). In particular, variants of Hamzated Alif, أ Â[1] or إ Ă are often written without their Hamza (ء '): ا *A*; and the Alif-Maqsura (or dotless Ya) ى ý and the regular dotted Ya ي *y* are often used interchangeably in word final position. This inconsistent variation in raw Arabic text is typically addressed in Arabic NLP through what is called orthographic normalization, a reductive process that converts all Hamzated Alif forms to bare Alif and dotless Ya/Alif Maqsura form to dotted Ya. We will refer to this kind of normalization as a Reduced normalization (RED). We introduce a different type of normalization that selects the appropriate form of the Alif. We call this Enriched normalization (ENR). ENR Arabic is optimally the desired correct form of Arabic to generate.

[1] All Arabic transliterations are provided in the Habash-Soudi-Buckwalter transliteration scheme (Habash et al., 2007).

Comparing a manually enriched (ENR) version of the Penn Arabic Treebank (PATB) (Maamouri et al., 2004) to its reduced (RED) version, we find that 16.2% of the words are different. However, the raw version of the PATB is only different in 7.4% of the words. This suggests a major problem in the recall of the correct ENR form in raw text.

Another orthographic issue is the optionality of diacritics in Arabic script. In particular, the absence of the Shadda diacritic (ـّ ∼) which indicates a doubling of the consonant it follows leads to a different number of letters in the tokenized and untokenized word forms (when the tokenization happens to split the two doubled consonants). See the example in Table 1 under (Y-Shadda). Consequently, the detokenization task for such cases is not a simple string concatenation.

3.2. Arabic Morphology

Arabic is a morphologically complex language with a large set of morphological features producing a large number of rich word forms. While the number of (morphologically untokenized) Arabic words in a parallel corpus is 20% less than the number of corresponding English words, the number of unique Arabic word types is over twice the number of unique English word types over the same corpus size.

One aspect of Arabic that contributes to this complexity is its various attachable clitics. We define three degrees of cliticization that are applicable in a strict order to a word base:

```
[cnj+ [prt+ [art+ BASE +pro]]]
```

At the deepest level, the BASE can have either the definite article (+ال *Al+* 'the') or a member of the class of pronominal enclitics, +pro, (e.g., +هم *+hm* 'their/them'). Next comes the class of particle proclitics (prt+), e.g., +ل *l+* 'to/for'. At the shallowest level of attachment we find the conjunction proclitic (cnj+), e.g., +و *w+* 'and'. The attachment of clitics to word forms is not a simple concatenation process. There are several orthographic and morphological adjustment rules that are applied to the word. An almost complete list of these rules relevant to this paper are presented and exemplified in Table 1.

It is important to make the distinction here between simple word segmentation, which splits off word substrings with no orthographic/morphological adjustments, and tokenization, which does. Although segmentation by itself can have important advantages, it leads to the creation of inconsistent or ambiguous word forms: consider the words مكتبة *mktbħ* 'library' and مكتبهم *mktbthm* 'their library'. A simple segmentation of the second word creates the non-word string مكتبت *mktbt*; however, applying adjustment rules as part of the tokenization generates the same form of the basic word in the two cases. For more details, see (Habash, 2007). In this paper, we do not explore morphological tokenization beyond decliticization.

4. Approach

We would like to study the value of a variety of detokenization techniques over different tokenization schemes and orthographic normalization. We report results on naturally occurring Arabic text and English-Arabic SMT outputs. To that end, we consider the following variants:

4.1. Tokenization

We consider five tokenization schemes discussed in the literature, in addition to a baseline no-tokenization scheme (D0). The D1, D2, TB and D3 schemes were first presented by Habash and Sadat (2006) and the S2 scheme was presented by Badr et al. (2008). The S1 scheme used by Badr et al. (2008) is the same as Habash and Sadat (2006)'s D3 scheme. TB is the PATB tokenization scheme. We use the Morphological Analysis and Disambiguation for Arabic (MADA) toolkit (Habash and Rambow, 2005) to produce the various tokenization schemes. The schemes are presented in Table 2 with various relevant statistics. The schemes differ widely in terms of the increase of number of tokens and the corresponding type count reduction. The more verbose schemes, i.e., schemes with more splitting, have lower out-of-vocabulary (OOV) rates and lower perplexity but are also harder to predict correctly.

4.2. Detokenization

We compare the following techniques for detokenization:

- Simple (S): concatenate clitics to word without applying any orthographic or morphological adjustments.
- Rule-based (R): use deterministic rules to handle all of the cases described in Table 1. We pick the most frequent decision for ambiguous cases.
- Table-based (T): use a lookup table mapping tokenized forms to detokenized forms. The table is based on pairs of tokenized and detokenized words from our language model data which had been processed by MADA. We pick the most frequent decision for ambiguous cases. Words not in the table are handled with the (S) technique. This technique essentially selects the detokenized form with the highest conditional probability $P(detokenized|tokenized)$.
- Table+Rule(T+R): same as (T) except that we back off to (R) not (S).

The above four techniques are the same as those used by Badr et al. (2008). We introduce two new techniques that use a 5-gram untokenized-form language model and the `disambig` utility in the SRILM toolkit (Stolcke, 2002) to decide among different alternatives:

- T+LM: we use all the forms in the (T) approach. Alternatives are given different conditional probabilities, $P(detokenized|tokenized)$, derived from the tables. Backoff is the (S) technique. This technique essentially selects the detokenized form with the highest $P(detokenized|tokenized) \times P_{LM}(detokenized)$.
- T+R+LM: same as (T+LM) but with (R) as backoff.

Rule Name	Condition	Result	Example		
Definite Article	?ل+ال+ل *l+Al+l?*	+لل *ll+*	مكتب+ال+ل *l+Al+mktb*	للمكتب *llmktb*	‘for the office’
			الجنة+ال+ل *l+Al+ljnħ*	للجنة *lljnħ*	‘for the committee’
Ta-Marbuta	ة- *-ħ* +pron	ت- *-t* +pron	مكتبة+هم *mktbħ+hm*	مكتبتهم *mktbthm*	‘their library’
Alif-Maqsura	ى- *-ý* +pron	ا- *-A* +pron	روى+ه *rwY+h*	رواه *rwAh*	‘he watered it’
	exceptionally	ي- *-y* +pron	على+ه ς*lY+h*	عليه ς*lyh*	‘on him’
Waw-of-Plurality	وا- *-wA* +pron	و- *-w* +pron	كتبوا+ه *ktbwA+h*	كتبوه *ktbwh*	‘they wrote it’
	تم- *-tm* +pron	تمو- *-tmw* +pron	كتبتم+ه *ktbtmw+h*	كتبتموه *ktbtmwh*	‘you [pl.] wrote it’
Hamza	ء- *-’* +pron	ئ- *-ŷ* +pron	بهاء+ه *bhA’+h*	بهائه *bhAŷh*	‘his glory [gen.]’
	less frequently	ؤ- *-ŵ* +pron	بهاء+ه *bhA’+h*	بهاؤه *bhAŵh*	‘his glory [nom.]’
	less frequently	ء- *-’* +pron	بهاء+ه *bhA’+h*	بهاءه *bhA’h*	‘his glory [acc.]’
Y-Shadda	ي+ي- *-y* +*y*	ي *y*	قاضي+ي *qADy+y*	قاضيّ *qADy*	‘my judge’
N-Assimilation	من *mn* +*m/n*	م *m* +*m/n*	من+ما *mn+mA*	ممـا *mmA*	‘from which’
	عن ς*n* +*m/n*	ع ς +*m/n*	عن+من ς*n+mn*	عمن ς*mn*	‘about whom’
	لا+أن *Ân* +*lA*	ألّا *ÂlA*	أن+لا *Ân+lA*	ألّا *ÂlA*	‘that ... not’

Table 1: Orthographic and Morphological Adjustment Rules

	Definition	Change Relative to D0			Prediction Error Rate			OOV		Perplexity	
		Token#	ENR Type#	RED Type#	ENR	RED	SEG	ENR	RED	ENR	RED
D0	word				0.62	0.09	0.00	2.22	2.17	412.3	410.6
D1	cnj+ word	+7.2	-17.6	-17.8	0.76	0.23	0.14	1.91	1.89	259.3	258.2
D2	cnj+ prt+ word	+13.3	-32.3	-32.6	0.89	0.37	0.25	1.50	1.50	185.5	184.7
TB	cnj+ prt+ word +pro	+17.9	-43.9	-44.2	1.07	0.57	0.42	1.22	1.22	142.2	141.5
S2	cnj+prt+art word +pro	+40.6	-53.0	-53.3	1.20	0.73	0.60	0.91	0.91	69.3	69.0
D3	cnj+ prt+ art+ word +pro	+44.2	-53.0	-53.3	1.20	0.73	0.60	0.90	0.90	61.9	61.7

Table 2: A comparison of the different tokenization schemes studied in this paper in terms of their definition, the relative change from no-tokenization (D0) in tokens (Token#) and enriched and reduced word types (ENR Type# and RED Type#), MADA’s error rate in producing the enriched tokens, the reduced tokens and just segmentation (SEG); the out-of-vocabulary (OOV) rate; and finally the perplexity value associated with different tokenization. OOV rates and perplexity values are measured against the NIST MT04 test set while prediction error rates are measured against a Penn Arabic Treebank devset.

4.3. Normalization

We consider two kinds of orthographic normalization schemes, enriched Arabic (ENR) and reduced Arabic (RED). For tokenized enriched forms, the detokenization produces the desired output. In case of reduced Arabic, we consider two alternatives to automatic orthographic enrichment. First, we use MADA to enrich Arabic text after detokenization (MADA-ENR). MADA can predict the correct enriched form of Arabic words at 99.4%.[2] Alternatively, we jointly detokenize and enrich using detokenization tables that map reduced tokenized words to their enriched detokenized form (Joint-DETOK-ENR).

In terms of evaluation, we report our results in both reduced and enriched Arabic forms. We only compare in the matching form, i.e., reduced hypothesis to reduced reference and enriched hypothesis to enriched reference.

[2]Statistics are measured on a devset from the Penn Arabic Treebank (Maamouri et al., 2004).

5. Experimental Results

5.1. Detokenization

We compare the performance of the different detokenization techniques discussed in Section 4. for the ENR and the RED normalization conditions. The performance of the different techniques is measured against the Arabic side of the NIST MT evaluation set for 2004 and 2005 (henceforth, MT04+MT05) which together have 2,409 sentences comprising 64,554 words. We report the results in Table 3 in terms of sentence-level detokenization error rate defined as the percentage of sentences with at least one detokenization error. The best performer across all conditions is the T+R+LM technique. The previously reported best performer was T+R (Badr et al., 2008), which was only compared with D3 and S2 tokenizations only.

As illustrated in the results, the more complex the tokenization scheme, the more prone it is to detokenization errors. Moreover, RED has equal or worse results than ENR under all conditions except for the S detokenization technique with the TB, S2 and D3 schemes. This is a result of the S

detokenization technique not performing any adjustments, which leads to the never-word-internal Alif-Maqsura character appearing incorrectly in word-internal positions in ENR. While for RED, the Alif-Maqsura is reductively normalized to Ya, which is the correct form in some of the cases.

The results for S2 and D3 are identical because these two schemes only superficially differ in whether proclitics are space-separated or not. Similarly, TB results are identical to D3 for the S and R techniques. This can be explained by the fact that the only difference between the D3 and TB schemes is that the definite article is attached to the word (in TB and not D3), a difference that does not produce different results under the deterministic S and R techniques.

We analyze the errors (14 cases) for the T+R+LM technique on D3 scheme and classify them into two categories. The first category comprises 11 cases ($\approx$ 80% of the errors) and is caused by ambiguity resulting from the lack of diacritical marks. Seven (50% overall) of these errors involve the selection of the correct Hamza form before a pronominal enclitic. For example, the tokenized word وأشقاء+ها *w+Âšqa'+hA* 'and+siblings+her' can be detokenized to وأشقاءها *wÂšqa'hA* or وأشقائها *wÂšqAŷhA* or وأشقاؤها *wÂšqAŵhA* depending on the grammatical case of the noun أشقاء *Âšqa'*, which is only expressible as a diacritical mark. The other four cases involve two closed class words, إن *Ăn* and لكن *lkn*, each of which corresponding to two diacritized forms that require different adjustments. For example, the tokenized word إن+ني *Ăn+ny* can be detokenized to إني *Ăny* (إن+ني *Ăin+niy* $\rightarrow$ إنِّي *Ăin∼iy*) or إنني *Ănny* (إنَّ+ني *Ăin∼a+niy* $\rightarrow$ إنَّني *Ăin∼aniy*). In many cases, the n-gram language model is able to select for the correct form, but it is not always successful. The second category of errors compromises 3 cases ($\approx$ 20% of the errors) which involve automatic tokenization failures producing tokens that are impossible to map back to the correct detokenized form.

5.2. Orthographic Enrichment and Detokenization

As previously mentioned, it's desirable for Arabic-generating automatic applications to produce orthographically correct Arabic. As such, reduced tokenized output should be enriched and detokenized to produce proper Arabic. We compare next the two different enrichment techniques discussed in Section 4.: using MADA to enrich detokenized reduced text (MADA-ENR) versus detokenizing and enriching in one joint step (Joint-DETOK-ENR). We consider the effect of applying these two techniques together with the various detokenization techniques when possible. The comparison is presented for D3 in Table 4. D3 has the highest number of tokens per word and it's the hardest to detokenize as shown in Table 3. The MADA-ENR enrichment technique can be applied to the output of all detokenization techniques; however, the Joint-DETOK-ENR enrichment technique can only be used as part of table-based detokenization techniques. The results for basic ENR and RED detokenization are in columns two and three. Columns four and five present the two approaches to enriching the tokenized reduced text. Although the Joint-DETOK-ENR technique does not outperform MADA-ENR for T and T+R, it significantly benefits from the use of the LM extension to these two techniques. In fact, Joint-DETOK-ENR produces the best results overall under T+R+LM, with an error rate that is 20% lower than the best performance by MADA-ENR. Overall, however, enriching and detokenizing RED text yields output that has almost 10 times the error rate compared to detokenizing ENR. This is expected since ENR is far less ambiguous than RED. The best performer across all conditions for detokenization and enrichment is the T+R+LM approach.

All experiments reported so far in this paper start with a perfect pairing between the original and tokenized words. The real challenge is applying the detokenization techniques on automatically produced (noisy) text. The next section discusses the effect of detokenization on SMT output as a case study.

5.3. Tokenization and Detokenization for SMT

In this section we present English-to-Arabic SMT as a case study for the effect of tokenization in improving the quality of translation. Then, we show the performance of the different detokenization techniques on the output and their reflections over the overall performance of the SMT systems.

5.3.1. Experimental Data

All of the training data we use is available from the Linguistic Data Consortium (LDC).[3] We use an English-Arabic parallel corpus of about 142K sentences and 4.4 million words for translation model training data. The parallel text includes Arabic News (LDC2004T17), eTIRR (LDC2004E72), English translation of Arabic Treebank (LDC2005E46), and Ummah (LDC2004T18). Lemma based word alignment is done using GIZA++ (Och and Ney, 2003). For language modeling, we use 200M words from the Arabic Gigaword Corpus (LDC2007T40) together with the Arabic side of our training data. Twelve language models were built for all combinations of normalization and tokenization schemes. We used 5-grams for all LMs unlike (Badr et al., 2008) who used different n-grams sizes for tokenized and untokenized variants. All LMs are implemented using the SRILM toolkit (Stolcke, 2002).

MADA is used to preprocess the Arabic text for translation modeling and language modeling. MADA produced all enriched forms and tokenizations. Due to the fact that the number of tokens per sentence changes from one tokenization scheme to another, we filter the training data so that all experiments are done on the same number of sentences. We use the D3 tokenization scheme as a reference and set the cutoff at 100 D3 tokens. English preprocessing simply included down-casing, separating punctuation from words and splitting off "'s".

All experiments are conducted using the Moses phrase-based SMT system (Koehn et al., 2007). The decoding weight optimization was done using a set of 300 sentences from the 2004 NIST MT evaluation test set (MT04). The

[3] http://www.ldc.upenn.edu

	S		R		T		T+R		T+LM		T+R+LM	
	ENR	RED	ENR	RED	ENR	RED	ENR	RED	ENR	RED	ENR	RED
D1	0.17	0.17	0.17	0.17	0.08	0.08	0.08	0.08	0.08	0.08	**0.08**	**0.08**
D2	22.50	22.50	0.58	0.79	0.37	0.37	0.21	0.21	0.37	0.37	**0.21**	**0.21**
TB	38.36	35.53	1.41	3.03	1.33	1.49	0.75	0.91	1.16	1.25	**0.58**	**0.66**
S2	38.36	35.53	1.41	3.03	1.37	1.54	0.79	0.95	1.20	1.29	**0.62**	**0.71**
D3	38.36	35.53	1.41	3.03	1.37	1.54	0.79	0.95	1.20	1.29	**0.62**	**0.71**

Table 3: Detokenization results in terms of sentence-level detokenization error rate.

Detokenization	ENR	RED		
	ENR	RED	MADA-ENR	Joint-DETOK-ENR
S	38.36	35.53	39.73	
R	1.41	3.03	10.59	
T	1.37	1.54	8.92	9.46
T+R	0.79	0.95	8.68	9.22
T+LM	1.20	1.29	9.34	6.23
T+R+LM	**0.62**	**0.71**	**7.39**	**5.89**

Table 4: Detokenization and enrichment results for D3 tokenization scheme in terms of sentence-level detokenization error rate.

tuning is based on the tokenized Arabic without detokenization. We use a maximum phrase length of size 8 for all experiments. We report results on the 2005 NIST MT evaluation set (MT05). These test sets were created for Arabic-English MT and have 4 English references. We use only one Arabic reference in reverse direction for both tuning and testing. We evaluate using BLEU-4 (Papineni et al., 2002) although we are aware of its caveats (Callison-Burch et al., 2006).

5.3.2. Tokenization Experiments

System	ENR		RED	
Evaluation	ENR	RED	ENR	RED
D0	24.63	24.67	24.66	24.71
D1	25.92	25.99	26.06	26.12
D2	26.41	26.49	26.06	26.15
TB	**26.46**	**26.51**	**26.73**	**26.80**
S2	25.71	25.76	26.11	26.19
D3	25.68	25.75	25.03	25.10

Table 5: Comparing different tokenization schemes for statistical MT in BLEU scores over detokenized Arabic (using T+R+LM technique)

We compare the performance of the different tokenization schemes and normalization conditions. The results are presented in Table 5 using T+R+LM detokenization technique. The best performer across all conditions is the TB scheme. The previously reported best performer was S2 (Badr et al., 2008), which was only compared against D0 and D3 tokenizations. Our results are consistent with Badr et al. (2008)'s results regarding D0 and D3. However, our TB result outperforms S2. The differences between TB and all other conditions are statistically significant above the 95% level. Statistical significance is computed using paired bootstrap resampling (Koehn, 2004). Training over RED Arabic then enriching its output sometimes yields better results than training on ENR directly which is the case with the TB tokenization scheme. However, sometimes the opposite is true as demonstrated in the D3 results. This is due to the tradeoff between the quality of translation and the quality of detokenization which is discussed in the next section.

5.3.3. Detokenization Experiments

We measure the performance of the different detokenization techniques discussed in Section 4. against the SMT output for the TB tokenization scheme. We report results in terms of BLEU scores in Table 6. The results for basic ENR and RED detokenization are in columns two and three. Column four presents the results for the Joint-DETOK-ENR approach to joint enriching and detokenization of tokenized reduced output discussed in Section 4.

When comparing Table 6 (in BLEU scores) with the corresponding cells in Table 4 (in sentence-level detokenization error rate), we observe that the wide range of performance in Table 4 is not reflected in BLEU scores in Table 6. This is expected given the different natures of the tasks and metrics used. Although the various detokenization techniques do not preserve their relative order completely, the S technique remains the worst performer and T+R+LM remains the best in both tables. However, the R and T+LM techniques perform relatively much better with MT output than they do with naturally occurring text. The most interesting observation is perhaps that under the best performing T+R+LM technique, joint detokenization and enrichment (Joint-DETOK-ENR) outperforms ENR detokenization despite the fact that Joint-DETOK-ENR has over nine times the error rate in Table 4. This shows that improved MT quality using RED training data out-weighs the lower quality of automatic enrichment.

Detokenization	ENR	RED	
	ENR	RED	Joint-DETOK-ENR
S	25.57	26.04	N/A
R	26.45	26.78	N/A
T	26.40	26.78	22.44
T+R	26.40	26.78	22.44
T+LM	26.46	26.80	26.73
T+R+LM	**26.46**	26.80	**26.73**

Table 6: BLEU scores for SMT outputs with different detokenization techniques over TB tokenization scheme

5.3.4. SMT Detokenization Error Analysis

Since we do not have a gold detokenization reference for our MT output, we automatically identify detokenization errors resulting in non-words (i.e., invalid words). We analyze the SMT output for the D3 tokenization scheme and T+R+LM detokenization technique using the morphological analyzer component in the MADA toolkit,[4] which provides all possible morphological analyses for a given word and identifies words with no analysis. We find 94 cases of words with no analysis out of 27,151 words (0.34%), appearing in 84 sentences out of 1,056 (7.9%). Most of the errors come from producing incompatible sequences of clitics, such as having a definite article with a pronominal clitic. For instance, the tokenized word ال+علاقة+نا *Al+ςlAqħ+nA* 'the+relation+our' is detokenized to العلاقتنا *AlςlAqtnA* which is grammatically incorrect. This is not a detokenization problem per se but rather an MT error. Such errors could still be addressed with specific detokenization extensions such as removing either the definite article or the pronominal clitic.

6. Conclusions and Future Work

We presented experiments studying six detokenization techniques to produce orthographically correct and enriched Arabic text. We presented results on naturally occurring Arabic text and MT output against different tokenization schemes. The best technique under all conditions is T+R+LM for both naturally occurring Arabic text and MT output. Regarding enrichment, joint enrichment with detokenization gives better results than performing the two tasks in two separate steps. Moreover, the best setup for MT is training on RED text and then enriching and detokenizing the output using the joint technique.

In the future, we plan to investigate the creation of mappers trained on seen examples in our tables to produce ranked detokenized alternatives for unseen tokenized word forms. In addition, we plan to examine language modeling approaches that target Arabic's complex morphology such as factored LMs (Bilmes and Kirchhoff, 2003). We also plan to explore ways to make detokenization robust to MT errors.

[4]This component uses the databases of the Buckwalter Arabic Morphological Analyzer (Buckwalter, 2004).

7. Acknowledgement

The work presented here was funded by a Google research award. We would like to thank Ioannis Tsochantaridis, Marine Carpuat, Alon Lavie, Hassan Al-Haj and Ibrahim Badr for helpful discussions.

8. References

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.

Jeff A. Bilmes and Katrin Kirchhoff. 2003. Factored language models and generalized parallel backoff. In *Proceedings of the Human Language Technology Conference/North American Chapter of Association for Computational Linguistics (HLT/NAACL-03)*, pages 4–6, Edmonton, Canada.

T. Buckwalter. 2004. Buckwalter Arabic Morphological Analyzer Version 2.0. Linguistic Data Consortium, University of Pennsylvania, 2002. LDC Cat alog No.: LDC2004L02, ISBN 1-58563-324-0.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the Role of BLEU in Machine Translation Research. In *Proceedings of the 11th conference of the European Chapter of the Association for Computational Linguistics (EACL'06)*, pages 249–256, Trento, Italy.

F. Diehl, M.J.F. Gales, M. Tomalin, and P.C. Woodland. 2009. Morphological Analysis and Decomposition for Arabic Speech-to-Text Systems. In *Proceedings of InterSpeech*.

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)*, pages 573–580, Ann Arbor, Michigan, June. Association for Computational Linguistics.

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.

Nizar Habash, Abdelhadi Soudi, and Tim Buckwalter. 2007. On Arabic Transliteration. In A. van den Bosch and A. Soudi, editors, *Arabic Computational Morphology: Knowledge-based and Empirical Methods*. Springer.

Nizar Habash. 2007. Arabic Morphological Representations for Machine Translation. In A. van den Bosch and A. Soudi, editors, *Arabic Computational Morphology: Knowledge-based and Empirical Methods*. Springer.

Ilana Heintz. 2008. Arabic language modeling with finite state transducers. In *Proceedings of the ACL-08: HLT Student Research Workshop*, pages 37–42, Columbus, Ohio, June. Association for Computational Linguistics.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June.

Philipp Koehn. 2004. Statistical significance tests formachine translation evaluation. In *Proceedings of the Empirical Methods in Natural Language Processing Conference (EMNLP'04)*, Barcelona, Spain.

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 Linguistics/Human Language Technologies Conference (HLT-NAACL04)*, pages 57–60, Boston, MA.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. 2004. The Penn Arabic Treebank : Building a Large-Scale Annotated Arabic Corpus. In *NEMLAR Conference on Arabic Language Resources and Tools*, pages 102–109, Cairo, Egypt.

Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–52.

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

Ruhi Sarikaya and Yonggang Deng. 2007. Joint morphological-lexical language modeling for machine translation. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 145–148, Rochester, New York, April. Association for Computational Linguistics.

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

Andreas Zollmann, Ashish Venugopal, and Stephan Vogel. 2006. Bridging the inflection morphology gap for arabic statistical machine translation. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 201–204, New York City, USA. Association for Computational Linguistics.