Arabic-Segmentation Combination Strategies for Statistical Machine Translation

Saab Mansour and Hermann Ney

Human Language Technology and Pattern Recognition
Computer Science Department
RWTH Aachen University, Aachen, Germany
{mansour,ney}@cs.rwth-aachen.de

Abstract

Arabic segmentation was already applied successfully for the task of statistical machine translation (SMT). Yet, there is no consistent comparison of the effect of different techniques and methods over the final translation quality. In this work, we use existing tools and further re-implement and develop new methods for segmentation. We compare the resulting SMT systems based on the different segmentation methods over the small IWSLT 2010 BTEC and the large NIST 2009 Arabic-to-English translation tasks. Our results show that for both small and large training data, segmentation yields strong improvements, but, the differences between the top ranked segmenters are statistically insignificant.

Due to the different methodologies that we apply for segmentation, we expect a complimentary variation in the results achieved by each method. As done in previous work, we combine several segmentation schemes of the same model but achieve modest improvements. Next, we try a different strategy, where we combine the different segmentation *methods* rather than the different segmentation *schemes*. In this case, we achieve stronger improvements over the best single system. Finally, combining schemes and methods has another slight gain over the best combination strategy.

Keywords: Arabic segmentation, Statistical machine translation, System combination

1. Introduction

Arabic segmentation tools for the task of statistical machine translation (SMT) were already successfully applied in previous work. Such tools get a sequence of words as input and output the corresponding sequence of segments. The methods used to implement these tools vary from rule-based methods (typically encoded as finite state transducer) such as (El Isbihani et al., 2006), to methods which are statistically-based such as (Roth et al., 2008) and (Mansour, 2010). These previous work have shown that segmentation improves machine translation (MT) quality significantly for both small and large scale tasks. Encouraged by these results, we start out by collecting existing tools and further re-implement and develop new methods for segmentation. Having all these methods at hand enabled us to perform consistent evaluation and comparison of the methods over the final MT system quality.

Due to the different methodologies that we apply for segmentation, we expect that there will be complimentary variation in the results achieved by each method. The next step would be to exploit these variations and achieve better results by combining the systems. Combining different segmentations was already applied in (Sadat and Habash, 2006). In their work, they use the same model to perform full morphological disambiguation, and then generate different segmentation schemes from the morphological analyses. A scheme is merely a decision which morphemes to split and treat as stand-alone words and which are kept attached to the stem. After producing several schemes, they combine them by phrase table combination followed by rescoring. We differ from their work three folds: *(i)* not only we compare several schemes of the same model, but we also compare different methods and statistical models for Arabic segmentation; *(ii)* we use a state-of-the-art system combination method similar to (Leusch and Ney, 2010) *(iii)* and we try different combination strategies, combining different schemes, different methods or a mixture of these. Our results show that "methods" combination performs better than "schemes" combination and achieves significant improvements over the best single system.

This paper is organized as follows. In Section 2., we discuss the problems of Arabic SMT, present the solution of segmentation and discuss its advantages. In Section 3., we present the data and resources that will be used to build our segmenters and the SMT system. The different methods for segmentation, including modeling and implementation details, will be presented in Section 4.. Evaluation and discussion of the results of the various segmentation methods and schemes will be presented in Section 5.. In Section 6., we briefly introduce the system combination framework used in this work, and evaluate different strategies for combination. A discussion of the results and further examples including final remarks and future work are given in Section 7..

2. Arabic segmentation

Written Modern Standard Arabic (henceforth *Arabic*) is known for its complex morphology and ambiguous writing system. These complexities are expressed in an SMT system at several levels. The first step in most of state-of-the-art SMT systems, after processing the bilingual corpora, is to generate a word alignment between the source and the corresponding target (translation) sentence. Form these alignments a word lexicon and more importantly a phrase lexicon (usually using heuristics) are extracted. In Arabic,

one word often corresponds to more than one word in traditional target languages such as English, posing a problem to the traditional IBM alignment models (Brown et al., 1993). Those complex Arabic words are generated from the attachment of a stem to prefix, affix and suffix clitics. Segmenting a word into its corresponding morphemes is already an ambiguous process and relies not only on grammatical rules, but also on the context of the word at hand. Ambiguity is even a harder problem in Arabic, expressed in the lack of short vowels in written Arabic and the high-degree of grammatical inflection. The increase of ambiguity is expressed in the increased number of possible translations per word, but, in addition, it is expressed in the possible segmentations of the word which eventually affects the corresponding translations.
A well studied solution to the problems mentioned above is Arabic word segmentation. Splitting an Arabic word into its corresponding prefixes, stem and suffixes lessens the number of out-of-vocabulary (OOV) words, resolves some of the ambiguous Arabic words and generates more one-to-one correspondences between the Arabic side and the target language side which can be easily captured by the IBM alignment models.
As mentioned in Section 1., some work has been done on Arabic segmentation for SMT. The FST tool presented by (El Isbihani et al., 2006) inherently suffers from ambiguous words which are not segmented in the approach. Another problem of the FST method is that it achieves improved results over a statistical segmenter for a small task, but inferior results for a large task. Another well known segmentation tool for Arabic is the MADA tool. (Sadat and Habash, 2006) perform a comparison between the different segmentation schemes supported by MADA, but a comparison to other techniques is not included.
In this work, we develop few segmentation models for Arabic and consistently evaluate them over the same translation task and training conditions. Furthermore, we apply system combination over the output of the resulting SMT systems and study the best strategies for combination.

3. Experimental setup

3.1. Arabic word segmentation

To train the segmentation methods, we use the Arabic Treebank Part 1 v3.0[1]. The treebank contains 150 000 word tokens and is drawn from the news genre. The Arabic words are segmented according to the so-called ATB scheme. In this scheme, prepositions (excluding the Arabic determiner *Al* and the future marker *s*) and possessive and objective pronouns are split from the Arabic stem.
For some models, we use a lexicon to limit the choice of possible segmentations. For this purpose, we use the Buckwalter Arabic Morphological Analyzer (BAMA) v1.0[2], a rule based analyzer, with 80 000 lexicon entries.
Note that in this work, we do not evaluate segmentation or Part-of-Speech (POS) tagging accuracies of the developed segmenters. We leave such evaluation and direct comparison to the resulting SMT systems quality to future work.

[1]LDC Catalog No. LDC2005T02
[2]LDC Catalog No. LDC2002L49

3.2. MT system

The MT system used in this paper to evaluate the different segmenters is a phrase-based SMT system. The system is an in-house implementation of state-of-the art phrase-based MT system as described in (Zens and Ney, 2008). We use the standard set of models with phrase translation probabilities for source-to-target and target-to-source direction, smoothing with lexical weights, a word and phrase penalty, distance-based and lexicalized reordering and an n-gram target language model.
We compare the resulting SMT systems based on the different segmentation methods over the small IWSLT 2010 BTEC (Paul et al., 2010) and the large NIST 2009 Arabic-to-English [3] translation tasks. We excluded the UN and the ISI data from the NIST corpora. This leaves 300K sentence pairs and 5M running words (about 6% of the whole data). This selection eases building the SMT systems, makes the training and testing genres consistent and the loss in performance is small. The loss is small due to the fact that the test sets are from the newswire genre, while the ISI data is noisy (automatically sentence-aligned) and the UN data is from the parliamentary speech genre and not the newswire one.
English preprocessing includes tokenization and lower casing. Arabic preprocessing includes removal of short vowels and tokenization.

4. Segmentation methods

In this section, we give a brief introduction to each segmentation method applied in this work. We mainly discuss high-level implementation details like the setup and tools used, selected features of the model whenever applicable, and advantages and disadvantages of the method.
The segmenters we experiment with throughout this work can be classified into three categories: rule-based, statistically-based and the later augmented with a lexicon.
Rule-based approaches for segmentation, such as (El Isbihani et al., 2006), employ linguistic knowledge extracted from human professionals. This knowledge is converted into machine readable rules and implemented using a framework capable of representing these rules (for example finite-state-transducer). These approaches are known for the difficulty of constructing the rules and are typically not very robust.
In contrast to rule based approaches, statistically-based approaches can be automatically trained and require very little handcrafted knowledge to be built into the system. Statistical methods could be identified by the probabilistic model they employ and the training criterion and procedure. In this work, we concentrate on comparing different probabilistic models while using the standard training procedures that come along with the software used. The probabilistic models we experiment with in this work are: Support-Vector-Machines (SVM), Hidden-Markov-Models (HMM) and Conditional-Random-Fields (CRF). The models differ in the dependence assumptions they make about their variables, the training procedures that are used to train the models' parameters and the training criterion.

[3]http://www.itl.nist.gov/iad/mig/tests/mt/2009/

The introduction of a lexicon into a statistical-based model was successfully applied for the task of segmentation and POS tagging in several works, among them (Habash and Rambow, 2005), (Bar-haim et al., 2008) and (Manour et al., 2007). In these works it was also shown that the disambiguation system can be aided by a lexicon and achieve better results when applying it.

Due to the different methodologies that we apply for segmentation, we expect that there will be complimentary variation in the results achieved by each method. The next step would be to exploit those variations and achieve better results by system combination. Different combination strategies could be applied over the output of the methods. We focus on two strategies: *(i)* similarly to (Sadat and Habash, 2006), we extract different segmentation *schemes* from the same model, and perform system combination over those schemes, *(ii)* a newly proposed method in this work is to combine the segmentation output of different segmentation *models* applying similar or difference schemes.

In the following subsections, we introduce each segmentation method applied in this work. A summary of the MT performance of the segmenters will be given in Section 5..

4.1. SVM

SVM classification is a supervised learning technique which finds the optimal separating hyperplane between two classes.

The power of SVM classification lies in the flexibility of setting the features vectors, where each entry represents a structure extracted from real world data.

Following (Diab et al., 2004), we use the YAMCHA[4] implementation of support vector machines to implement our SVM model. We use YAMCHA's default training conditions, which means a second degree polynomial kernel function.

Given some Arabic text, we run an SVM classifier to segment the text into segments. SVM segmentation is merely a classification task, where one uses a character based tag set.

The classes we use include: the beginning of the first prefix (B-PRE1), the beginning of a second prefix (B-PRE2), the beginning of a word (B-WRD), in word (I-WRD), the beginning of a suffix (B-SUFF), and in suffix (I-SUFF). The correct segmentation of وببيتهم *wbbythm* 'and in their house' is given in Table 1.

The segmentation features includes 10 characters surrounding the character in focus (-5/+5) and the 5 previous tag decisions, thus making the segmentation process solely character-based. Additionally, we apply feminine marker normalization ($tX \rightarrow p+X$) using a binary SVM model on top of the segmenter output, which proved to be significant for the performance of MT in our experiments.

4.2. CRF

CRF classification was applied successfully for NLP tasks, with several papers reporting favorite results over other training methods. For example, the paper of ((Lafferty et al., 2001), where the CRF model was first introduced and applied successfully for POS tagging of English, and a more recent example in (Trogkanis and Elkan, 2010), where CRF showed superior results on the task of word hyphenation. The first application we are aware of for segmentation was done by (Peng et al., 2004) for Chinese word segmentation, where the task was treated as binary classification.

CRFs offer similar flexibility to SVM, where one can define features that capture various structures in the data. We use similar setup of classifiers and classes as in the SVM model. The features for segmentation are similar to those of the SVM model. The software we use as an implementation of conditional random fields is named CRF++[5]. We adopt the default parameter settings of CRF++, so no development set or tuning set is needed in our work.

4.3. FST

The Finite State Transducer-based (FST) approach for Arabic segmentation, presented in (El Isbihani et al., 2006), is composed of two finite state transducers, one for stripping the prefixes and one for stripping the suffixes. The prefixes that are split include *w,f,k,l,b,Al* and *s*. Suffixes which are handled are pronouns (objective and possessive). Splitting is limited to allowed combinations of the clitics in the grammar of the Arabic language; the resulting stem must be observed in the corpus and the word in hand can not be ambiguous. The main advantage of this method is the segmentation speed it can achieve. In our experiments, we obtained an average speed of 4 500 words per second, which is the fastest among the experimented methods. Still, this method suffers from three main disadvantages:

- the lack of context in the decision procedure results in wrong segmentations
- ambiguous words are not segmented, and Arabic is a highly ambiguous language
- limiting segmentation to seen stems in the corpus causes inconsistencies among different corpora genres

As already reported by its authors, the FST method perform inconsistently among tasks, showing better results than the SVM segmenter on a small task, but inferior results on a large scale task.

4.4. MorphTagger

MorphTagger is a general architecture for Part-Of-Speech (POS) tagging of natural languages. The architecture was first proposed by (Bar-haim et al., 2008) where it was applied for the task of POS tagging of Hebrew. (Manour et al., 2007) adapted the architecture to the Arabic language, and later on for the task of SMT (Mansour, 2010). The architecture is similar to (Habash and Rambow, 2005) where one selects a specific analysis from the output of a morphological analyzer. First, the Arabic input sentence goes through a morphological analyzer, which outputs for each word all possible analyses. Each analysis includes a sequence of pairs of a segment and the corresponding POS tag. A disambiguation component selects the the most probable tagging sequence according to some model. Then,

[4] http://chasen.org/ taku/software/YamCha/

[5] http://crfpp.sourceforge.net/

و	ب	ب	ي	ت	ه	م
w	b	b	y	t	h	m
B-PRE1	B-PRE2	B-WRD	I-WRD	I-WRD	B-SUFF	I-SUFF

Table 1: SVM segmentation classes.

the corresponding segments is inferred from the tagging sequence. MorphTagger also applies several normalization steps which proved to be helpful for SMT. MorphTagger is implemented using BAMA as a morphological analyzer and an HMM model (using the SRILM[6] toolkit) as the disambiguator component.

4.5. MADA

The Morphological Analysis and Disambiguation of Arabic (MADA) system, developed by (Habash and Rambow, 2005), can be seen as an extension of an SVM-based system with the incorporation of a lexicon. The system uses several SVM-classifiers to classify individual morphological attributes, and then selects the best matching analyses proposed by the morphological analyzer by a simple combination scheme (for example, a majority analysis). (Habash and Rambow, 2005) report on improved segmentation and POS tagging results when compared to an SVM-based system as the one suggested by (Diab et al., 2004). For each morphological attribute, the features include the words in a window of size 5 around the current word, plus the previous two classification decisions. As in (Sadat and Habash, 2006), we experiment with different segmentation schemes for each chosen analysis. We use the schemes directly implemented in the MADA version we are using, namely: D1,D2,D3 and the ATB (TB) schemes.

5. Results

The results of the different segmentation methods and schemes are summarized in Table 2 and Table 3 for the IWSLT and NIST tasks correspondingly. For comparison purposes to the proposed segmenters, we include a TOK "segmenter" for Arabic which performs punctuation tokenization only. The development sets used to optimize the scaling factors of the SMT decoder are marked with *dev* within the tables. We include both BLEU and TER to measure the MT systems translation quality.

From the raw results, we observe that segmentation usually helps. In the case of the FST method, the inconsistent segmentations are causing a high rate of OOV words therefore inferior results. The MADA D1 scheme is characterized by a low degree of segmentation (only the conjunction clitics *f* and *w* are split) which proves insufficient, most importantly for the small task at hand. The MADA-D3 scheme is characterized by a high degree of segmentation, which seems to still get improvements on the small IWSLT task, but hurts the performance for the large NIST task.

In addition to the raw automatic results, we perform significance testing over the *IWSLT08* and *nist08* test sets. For both BLEU and TER we perform bootstrap resampling with bounds estimation as described in (Koehn, 2004). We use the 95% confidence threshold to draw significance conclusions. The size of the IWSLT08 test set is about 3 000 words with 16 references on the English side. The 95% level significance bounds were measured at ±3.0 BLEU and ±2.0 TER, thus deeming the differences between the various segmenters statistically insignificant, perhaps due to the small setting.

The results for the NIST task are quite similar to the IWSLT task, where we observe that segmentation helps improving the performance. The size of each of the nist test sets is about 40 000 words with 4 references on the English side. We measured ±1.0 BLEU and ±0.7 TER as significance bounds. Drawing conclusions with statistical significance, we found that for BLEU: *(i)* all systems are better than TOK, *(ii)* CRF,HMM and MADA-* are better than FST, *(iii)* HMM and MADA-TB are better than the other systems except MADA-D2.

	IWSLT05 (dev)		IWSLT08	
Segmentation	BLEU	TER	BLEU	TER
TOK	57.1	30.6	53.7	33.6
FST	57.9	28.8	54.3	32.2
SVM	59.1	28.9	54.7	32.5
CRF	58.5	29.3	54.6	32.8
HMM	59.4	28.2	54.5	32.2
MADA-D1	57.7	30.6	52.8	34.4
MADA-D2	57.4	29.8	53.8	32.9
MADA-D3	58.6	29.1	54.5	32.4
MADA-TB	58.2	29.8	54.6	32.6
MADA-ALL	60.8	27.6	55.7	31.5
SEGS-ALL	61.2	27.2	57.5	30.3
ALL	61.5	26.8	58.1	29.8

Table 2: IWSLT 2010 BTEC Arabic-English results summary.

6. System combination

System combination is used to produce consensus translations from multiple hypotheses generated with different systems built upon various segmentations of the Arabic text. The pipeline of the system combination is based on the pipeline described in (Leusch and Ney, 2010), which was used in the WMT 2010 evaluation and achieved state-of-the-art results.

Figure 1 gives an overview of the system combination architecture. After preprocessing the MT hypotheses, pairwise alignments (using GIZA++) between the hypotheses are calculated. The hypotheses are then reordered to match the word order of a selected primary or skeleton hypothesis. From this, a lattice is created which is then rescored using system prior weights and a language model. The single best

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

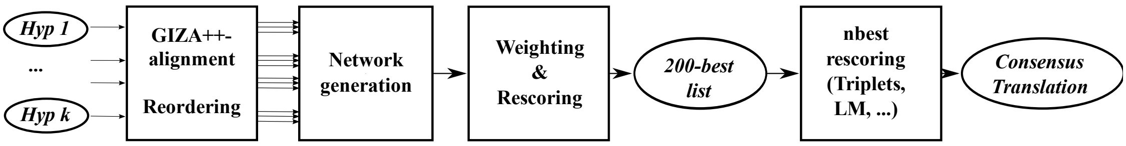

Figure 1: The system combination architecture.

Segmentation	nist04		nist05		nist06 (dev)		nist08	
	BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER
TOK	49.1	44.7	51.4	44.1	41.8	50.5	38.1	52.5
FST	52.6	42.0	54.2	41.6	42.7	49.2	39.2	51.4
SVM	50.7	43.2	53.2	41.9	43.4	48.9	39.9	51.3
CRF	51.0	43.0	53.0	42.0	43.5	48.8	40.5	50.7
HMM	52.3	42.0	53.8	41.5	44.0	48.3	41.5	49.9
MADA-D1	51.0	43.2	52.5	43.0	43.7	49.0	40.5	51.1
MADA-D2	51.7	42.8	53.7	42.3	44.6	48.3	41.4	50.5
MADA-D3	50.8	42.7	53.3	41.5	43.7	47.7	40.8	50.0
MADA-TB	52.7	41.5	54.6	40.9	45.1	47.4	41.8	49.4
MADA-ALL	53.3	41.1	55.1	40.6	45.5	47.0	42.7	48.7
SEGS-ALL	54.0	40.6	55.4	40.2	46.1	46.4	42.6	48.6
ALL	54.2	40.4	55.9	39.8	46.3	46.3	42.9	48.5

Table 3: NIST 2009 Arabic-English results summary.

path in this confusion network then constitutes the consensus translation which is outputted by this system. The consensus translation is then true cased and postprocessed.

The results of the different system combination variations were summarized in Table 2 and Table 3. We compared two combination strategies. Similar to (Sadat and Habash, 2006), we combine the different schemes of the MADA segmenter (*MADA-ALL*). In this case, we achieve improvements of +1.0% BLEU and -1.0% TER over the best single system for IWSLT, and +0.9% BLEU and -0.7% TER for NIST over the *nist08* test set.

Next, we try the combination of the different methods outputs (*SEGS-ALL*). For the IWSLT task, this gives an improvement of +2.8% BLEU and -2.2% TER, a significant (for BLEU at the 90% confidence level) improvement over the best single system. Furthermore, we tried a combination of all the schemes and methods (*ALL*), which gave a slight improvement of +0.6% BLEU and -0.5% TER over the methods combination.

Similar behavior is observed for the NIST task, excluding *nist08*, *SEGS-ALL* is performing better than *MADA-ALL* and achieves significant improvements in both BLEU and TER over the best single system, whereas *MADA-ALL* fails to achieve that. *ALL* combination gives another slight gain in performance. For the *nist08* test set, *MADA-ALL* and *SEGS-ALL* achieve comparable results.

7. Conclusions and outlook

In this work, we compared several available and self-developed Arabic segmentation methods for the task of SMT. Supporting the outcome of previous work, we found that high-degree of segmentation performs better than simple tokenization on a small scale Arabic to English translation task. Nevertheless, the differences between the high-degree segmentation methods proved to be statistically insignificant.

Next, we experimented with exploiting the advantages of the different segmentation-based SMT systems through system combination. We start out by combining several segmentation schemes of the same model. By this strategy, we achieve improvements over the best single system but the improvements proved to be statistically insignificant (except the *nist08* test set). Next, we tried a different strategy, where we combined the different segmentation methods rather than different segmentation schemes. In this case, we obtained significant improvement over the best single system. We also observed improvements over the schemes combination strategy. Finally, combining schemes and methods had another slight gain over the methods combination.

As future work, we plan to investigate the connection between segmentation and POS tagging accuracy to the final SMT system output. One interesting result that we obtained was the insignificant difference between lexicon based and non-lexicon based statistical models. This is in contradiction to previous results reported by (Chang et al., 2008), where it was shown that incorporating lexicon features into a CRF-based segmenter improves the segmentation accuracy, but more importantly it improves the consistency of the segmentations produced, thus leading to better translation performance (though they did not perform significance testing). We hypothesize that the difference to their work might be due the fact that Arabic segmentation accuracies are peaking the 99% accuracy level (Habash and Rambow, 2005) compared to the 95% accuracy level reported for Chinese (Peng et al., 2004). Another difference to their work is that they use a subsample of the GALE Year 2 training data for MT, which contains 40 million running words on the English side. Perhaps a lexicon and the consistency and accuracy of the segmenter will prove crucial for larger tasks. We leave the validation of this hypothesis to future work.

8. Acknowledgements

This material was partly realized under the Quaero Programme, funded by OSEO, French State agency for innovation.

9. References

Roy Bar-haim, Khalil Sima'an, and Yoad Winter. 2008. Part-of-speech tagging of modern hebrew text. *Nat. Lang. Eng.*, 14(2):223–251.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311, June.

Pi-Chuan Chang, Michel Galley, and Christopher D. Manning. 2008. Optimizing chinese word segmentation for machine translation performance. In *StatMT '08: Proceedings of the Third Workshop on Statistical Machine Translation*, pages 224–232, Morristown, NJ, USA. Association for Computational Linguistics.

Mona Diab, Kadri Hacioglu, and Daniel Jurafsky. 2004. Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks. In Daniel Marcu Susan Dumais and Salim Roukos, editors, *HLT-NAACL 2004: Short Papers*, pages 149–152, Boston, Massachusetts, USA, May 2 - May 7. Association for Computational Linguistics.

Anas El Isbihani, Shahram Khadivi, Oliver Bender, and Hermann Ney. 2006. Morpho-syntactic Arabic Preprocessing for Arabic to English Statistical Machine Translation. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 15–22, New York City, June. Association for Computational Linguistics.

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

Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *Proc. of the Conf. on Empirical Methods for Natural Language Processing (EMNLP)*, pages 388–395, Barcelona, Spain, July.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*, pages 282–289.

Gregor Leusch and Hermann Ney. 2010. The rwth system combination system for wmt 2010. In *ACL 2010 Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR*, Uppsala, Sweden, July.

Saib Manour, Khalil Sima'an, and Yoad Winter. 2007. Smoothing a lexicon-based pos tagger for arabic and hebrew. In *Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources*, pages 97–103, Prague, Czech Republic, June. Association for Computational Linguistics.

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

Michael Paul, Marcello Federico, and Sebastian Stücker. 2010. Overview of the IWSLT 2010 Evaluation Campaign. In Marcello Federico, Ian Lane, Michael Paul, and François Yvon, editors, *Proceedings of the seventh International Workshop on Spoken Language Translation (IWSLT)*, pages 3–27.

Fuchun Peng, Fangfang Feng, and Andrew McCallum. 2004. Chinese segmentation and new word detection using conditional random fields. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 562, Morristown, NJ, USA. Association for Computational Linguistics.

Ryan Roth, Owen Rambow, Nizar Habash, Mona Diab, and Cynthia Rudin. 2008. Arabic morphological tagging, diacritization, and lemmatization using lexeme models and feature ranking. In *Proceedings of ACL-08: HLT, Short Papers*, pages 117–120, Columbus, Ohio, June. Association for Computational Linguistics.

Fatiha Sadat and Nizar Habash. 2006. Combination of Preprocessing Schemes for Statistical MT. In *Proc. of the 44th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1–8, Sydney, Australia, July.

Nikolaos Trogkanis and Charles Elkan. 2010. Conditional random fields for word hyphenation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 366–374, Uppsala, Sweden, July. Association for Computational Linguistics.

Richard Zens and Hermann Ney. 2008. Improvements in Dynamic Programming Beam Search for Phrase-based Statistical Machine Translation. In *International Workshop on Spoken Language Translation*, pages 195–205, Honolulu, Hawaii, October.