PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 108 JUNE 2017 257–269

Optimizing Tokenization Choice for Machine Translation across Multiple Target Languages

Nasser Zalmout, Nizar Habash

Computational Approaches to Modeling Language Lab
New York University Abu Dhabi, United Arab Emirates

Abstract

Tokenization is very helpful for Statistical Machine Translation (SMT), especially when translating from morphologically rich languages. Typically, a single tokenization scheme is applied to the entire source-language text and regardless of the target language. In this paper, we evaluate the hypothesis that SMT performance may benefit from different tokenization schemes for different words within the same text, and also for different target languages. We apply this approach to Arabic as a source language, with five target languages of varying morphological complexity: English, French, Spanish, Russian and Chinese. Our results show that different target languages indeed require different source-language schemes; and a context-variable tokenization scheme can outperform a context-constant scheme with a statistically significant performance enhancement of about 1.4 BLEU points.

1. Introduction

In Statistical Machine Translation (SMT), words are usually designated as the basic tokens of translation and language modeling. However, especially for morphologically complex languages, using sub-lexical units obtained after morphological preprocessing has been shown to improve the machine translation performance over a word-based system (Popović and Ney, 2004; Habash and Sadat, 2006). For any language, several word tokenization choices, henceforth *tokenization schemes*, can be generated based on the word's in-context morphological analysis. These schemes vary by the intended amount of verbosity for the language and application context, and considered a blueprint for the tokenization process. Tokenization using these schemes is usually

 Corresponding author: nasser.zalmout@nyu.edu
Cite as: Nasser Zalmout, Nizar Habash. Optimizing Tokenization Choice for Machine Translation across Multiple Target Languages. The Prague Bulletin of Mathematical Linguistics No. 108, 2017, pp. 257–269.
doi: 10.1515/pralin-2017-0025.

performed as a preprocessing step to the SMT system, where the choice of the scheme is fixed and predetermined. The limitation of a predetermined single tokenization raises many questions: (a) would the best source language tokenization choice vary given different target languages? (b) would combining the various tokenization options in the training phase enhance the SMT performance? and (c) would considering different tokenization options at decoding time improve SMT performance?

The goal of the approach presented in this paper is to eliminate the fixed predetermined scheme selection that spans the entire text, and target languages, and allow for word-level tokenization scheme selection. This notion of word-level tokenization optimization can be achieved indirectly by combining training tokenization options, directly by lattice decoding of the various tokenization options, or through another indirect approach by learning a classifier on optimal tokenization choices. We apply these techniques on Arabic, where most tokenization contributions for SMT focus on Arabic-English translation, with little investigation of other target languages. We study the Arabic tokenization behavior against five target languages: English, French, Spanish, Russian and Chinese. We also introduce a new tokenization scheme to match some of their linguistic features.

2. Arabic Linguistic Issues

Arabic is a morphologically complex language, with various morphological features that control several inflectional variations, such as gender, number, person and voice, producing a large number of rich word forms. Moreover, clitics in Arabic are written attached to the word and thus increase its ambiguity, making word boundaries harder to detect properly. These morphological structures and attached clitics pose a special challenge for NLP tasks in general. These issues are particularly challenging for the tasks that are highly sensitive to the verbosity of the underlying sentences, like SMT, where each morpheme can be aligned to specific target language word. Figure 1 shows an example of such alignment, where a three-word Arabic sentence is aligned to an eight-word English sentence. Tokenization handles this issue by splitting the different clitics with various levels of verbosity, which helps reducing sparsity, perplexity, and out of vocabulary words.

The tokenization process depends on the morphological structure of the word, to identify the suitable morphemic decomposition. Hence, the first step in the tokenization process is to obtain the various morphological analyses of the given word, and choose the most likely one given the contextual surrounding, through a disambiguation process. The next step is choosing the tokenization scheme that the tokenization tool should use given the disambiguated morphological analysis. These schemes serve as a blueprint for the tokenization process, by controlling the types of clitics to be segmented, hence controlling the level of verbosity of the output texts.

There have been several tokenization schemes proposed in literature for Arabic, some of which include the schemes below, with examples provided at Table 1. An important observation about all these schemes, however, is that their outputs are not mutually exclusive, so multiple schemes might sometimes result in the same tokenization.

- Simple Tokenization (D0): Splits off punctuation and numbers, and optionally normalizing some linguistic phenomena.
- D1, D2, and D3: Decliticizations; using different levels of conjugation clitics splits.
- Penn Arabic Treebank (ATB) tokenization: Splits all clitics except the definite article.

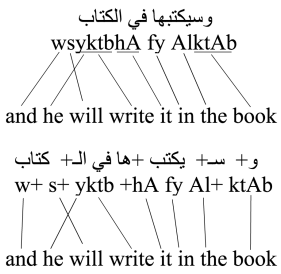

Figure 1. An example of Arabic alignment with English.

Other schemes include the MR (Morphemes); breaks up words into stem and affixival morphemes, and English-like scheme; using lexeme and English-like POS tags.

Selecting the relevant tokenization schemes is predetermined and fixed given the context and application, along with the intended level of verbosity. Moreover, for Arabic SMT, most of the previous contributions on tokenization focus on translating from Arabic to English or vice versa, generalizing tokenization selections to other languages and application domains.

Tokenization Scheme		Example	
D0	no tokenization	wsyktbhA	llTAlb
D1	split CONJ	w+ syktbhA	llTAlb
D2	split CONJ and PART	w+ s+ yktbhA	l+ AlTAlb
ATB	Arabic Treebank	w+ s+ yktb +hA	l+ AlTAlb
D3	split all clitics	w+ s+ yktb +hA	l+ Al+ TAlb

Table 1. Various Arabic tokenization schemes for the sentence wsyktbhA llTAlb 'and he will write it for the student'. Arabic words are presented in Buckwalter transliteration.

3. Background and Related Work

There have been several approaches for Arabic tokenization in literature. Lee et al. (2003) use a look-up table for the various prefixes, stems, and suffixes used in the tokenization process. Habash and Sadat (2006) presented various schemes for tokenizing Arabic text for MT, in addition to the Arabic Treebank tokenization (Maamouri et al., 2004). Diab et al. (2007) presented an SVM-based approach for tokenization. They use a classification based model, where each letter in a word is tagged with a label

indicating its morphological identity. FARASA (Abdelali et al., 2016) uses SVM-rank to rank potential word segmentations. MADAMIRA (Pasha et al., 2014); the current state-of-the-art tool for Arabic morphological analysis and disambiguation, obtains the disambiguated morphological analysis of the word, and feeds it to a tokenization engine. MADAMIRA utilizes MADA (Habash and Rambow, 2005; Roth et al., 2008) for morphological disambiguation. The top morphological analysis is then used for tokenization deterministically through one of the tokenization schemes. We use MADAMIRA to get the various word-level tokenization options, resulting from the various tokenization schemes, then analyze these for the optimal tokenization.

The issues of fixed and text-level selection of tokenization schemes has been previously addressed in literature, for morphologically complex languages in general, and Arabic in particular. Sadat and Habash (2006) presented a technique for maximizing the line-level output BLEU score of the SMT system by combining/consulting outputs of various SMT systems. A "deeper" version of their work that handles tokenization in decoding phase requires a "privilege" scheme, which creates a bias in the system. Moreover, their overall system focuses on optimizing over the SMT output, rather than selecting optimal tokenized inputs. Elming and Habash (2007) used the various tokenization options to build a machine learning model to enhance the quality of word alignments, rather than SMT. Other approaches for unsupervised morphological segmentation includes the work of Mermer (2010) for Turkish-English translation. They use IBM model-1 to formulate the translation objective function as the posterior probability of the training corpus according to a generative segmentation-translation model. Their model, however, didn't exhibit any significant BLEU enhancement. One of the notable contributions within this domain is the work of Dyer et al. (2008) and (Dyer, 2009), where they use a word lattice that encodes the surface forms (unsegmented words) as an option, and the full morphological breakdown of the surface form as another option. In this scope, the lattice is used to model a back-off system for the full morphological segmentation, rather than encoding the various tokenization schemes. Word lattices have also been used for a number of different applications in MT, including the work of Zhang et al. (2007), who use word lattices to model the different chunk-level reordering options.

We use a similar approach to Dyer's (Dyer et al., 2008) for lattice-based decoding of tokenization options, but through encoding all tokenization options at the lattice instead of using it as a backoff model to full morphological breakdown as they use it.

Word lattices and confusion networks are used in NLP mainly to model ambiguity in the input/output, and can be used to represent any finite set of strings. Word lattices, though, have the capability of representing an exponential number of sentences in polynomial space. The words within the lattice represent alternative choices of words in hypothesis, and the edges are used to model the weight or probability score.

4. Approach and Experimental Setup

We first build scheme-specific SMT systems for each language, with six schemes each. We then experiment with a simple scheme combination method, by combining different copies of the training set, each tokenized with a different scheme. Then we apply decoding-time scheme selection, through word lattice decoding of the test set. We finally develop a machine learning tool to learn the optimal tokenizations, as a tradeoff between execution complexity and accuracy.

MT Toolkits and Evaluation We use the Moses toolkit (Koehn et al., 2007) with default parameters to develop the machine translation systems, GIZA++ (Och and Ney, 2003) for alignment, and KenLM (Heafield et al., 2013) to build a 5-gram language model. We use BLEU score (Papineni et al., 2002) and METEOR (Banerjee and Lavie, 2005) for evaluation. Koehn (2004) presents a model for applying statistical significance tests over SMT evaluation metrics. He uses the bootstrap resampling method to measure the p-level statistical confidence. We use this approach for statistical significance tests throughout this paper, with p-value of 0.05.

Data and Preprocessing We use the Multi UN corpus (Eisele and Chen, 2010) throughout the experiments presented in this paper. We chose the Multi UN corpus to study tokenization behavior across several target languages without introducing additional variations. The UN corpus is a good fit as it is parallel for Arabic across five other languages, unlike other commonly used MT corpora.

We use 200,000 lines (circa 5.5 million words) for training, 1,000 lines (circa 25,000 words) for tuning, 3,000 lines (circa 90,000 words) for testing, and 9.5 million lines (circa 280 million words) for language models. The numbers are very similar across all languages we work with. We work with the relatively medium dataset sizes to best capture the tokenization effect, where data sparsity becomes of more relevance. The sparsity issue is particularly important when translating low-resource languages or domains (unlike English for example), which are of interest in this paper.

The preprocessing of the training data includes eliminating the lines beyond the length of 80 words. However, different tokenization schemes will result in different line lengths, which might cause imbalances among the different options. We therefore eliminate the lines across all files whose D3 tokenization exceeds 80 words. Considering D3, the most verbose scheme, as the basis for this elimination guarantees that there won't be any file containing lines exceeding 80 words.

We tokenize the Arabic content using the MADAMIRA toolkit (Pasha et al., 2014), with the alef/yaa normalization, to the various tokenization schemes (D0, D1, D2, ATB, D3). We also use off-the-shelve tools to tokenize the other five languages covered in the paper. We use the available tokenizers at Moses for English, Spanish and Russian, and use the Stanford Word Segmenter from the Stanford NLP Group for Chinese and French (using the TokenizerAnnotator tool).

***D3*: A New Tokenization Scheme** Many languages don't have a clear equivalent of the definite article *"the"*, or *"Al"* in Arabic, like Russian and Chinese. We suggest that removing the definite article in the tokenized source text (Arabic) when translating to these languages might enhance performance. To approach this issue we include a new tokenization scheme in our analysis, by removing the definite article *"Al"* from D3 scheme; which is the only scheme that splits the definite article among the schemes we work with. We designate this new scheme as *D3*.

5. Results and Analysis

We use the same dataset throughout the different experiments, with the same training/tuning/testing splits covered earlier. Each section below presents a different approach into tackling the scheme selection for Arabic tokenization.

5.1. Scheme Specific SMT Systems

The first set of experiments study the various tokenization schemes in isolation. We develop a total of 30 machine translation systems in this part, each corresponding to a specific tokenization scheme for each of the five target languages. Table 2

	English		French		Russian		Spanish		Chinese	
	BLEU	METEOR	BLEU	METEOR	BLEU	METEOR	BLEU	METEOR	BLEU	METEOR
D0	39.80	0.3736	26.79	0.4478	28.56	0.4659	40.70	0.6019	32.04	0.4815
D1	41.25	0.3805	27.71	0.4586	29.47	0.4827	40.92	0.6096	33.23	0.4954
D2	41.62	0.3839	27.89	0.4627	29.85	0.4880	**41.85**	0.6134	33.30	0.4971
D3	41.85	0.3807	27.89	0.4618	29.49	0.4881	41.47	**0.6153**	31.73	0.4848
ATB	41.91	0.3837	**27.91**	**0.4644**	30.38	0.4938	41.61	0.6140	33.30	0.4975
D3*	**41.94**	**0.3846**	27.76	0.4626	**30.55**	**0.4964**	41.66	0.6148	**33.51**	**0.4986**

Table 2. Scheme-specific SMT systems - baselines

presents the BLEU and METEOR for the 30 machine translation systems developed for analyzing the effect of varying tokenization schemes.

The character-level Chinese system outperforms word-level evaluation significantly, matching the results of Habash and Hu (2009). Both sets of results are directly correlated, however, so we present the character-level scores only.

In the Arabic-English systems both ATB and D3* perform closely. This behavior is consistent with the generally used tokenization scheme for Arabic with English as target language in literature, mostly working with ATB. French, on the other hand, shows consistent behavior favoring the ATB scheme for machine translation. The re-

sults for Spanish show that D2 and D3 outperform ATB and D3*. D3* performs the best in both BLEU and METEOR for Chinese and Russian, so our hypothesis proved right.

5.2. Training on Combined Schemes

The first schemes combination method we try is based on simple concatenation of the source training dataset copies (copies of the same previous training dataset), having each tokenized with a different tokenization scheme. The tokenization options resulting from the tokenization schemes are not mutually exclusive, so multiple schemes might result in the same tokenization in certain cases.

The dataset itself is copied and concatenated, so this doesn't constitute a bigger training set, it is rather a richer representation of the same set with additional tokenization options. For sanity check regarding the data duplication, we conducted side experiments by training the MT systems based on the individual schemes, having the training dataset duplicated six times. This did not result in any improvement, so we confirm that any overall improvement is not the result of duplicating the training data. We also duplicate each target language to match the source language (Arabic).

We then perform 30 additional experiments to test each individual tokenization scheme against this combined corpus. We tokenize the testing dataset for each language using each of the six schemes, and use it as a separate testing set for the system trained on the combined corpus.

Table 3 provides the results for the various experiments. The results show a noticeable improvement across all languages and for both BLEU and METEOR. This shows that providing more tokenization options at the training phase enhances the overall MT system performance. The results also show that ATB performs better than the other schemes across English, French, and Chinese, beating the scores for the D3*, even for Chinese where it showed considerable improvement earlier. A potential analysis is that concatenating the training files might have created a bias in the phrase-table model towards phrases that include the definite article, since all other schemes include the article within the tokenization (whether attached or segmented).

Russian and Spanish remain consistent in favoring D3* and D2 respectively, since Russian performs quite closely for D2, ATB and D3* at around 31 BLEU points.

5.3. Word Lattice Input

The word lattice decoding follows the noisier channel model (Dyer et al., 2008). Word lattices are primarily used to model ambiguity in NLP systems, this ambiguity can be referred to by an observed ambiguity signal, which produces a set of source-language strings $f' \in F(s)$. The objective function within this scope would be: $\hat{e} = \mathrm{argmax}_e \mathrm{max}_{f' \in F(s)} \Pr(e)\Pr(f'|e)\Pr(s|f')$. The different probabilities within the formula include: $\Pr(e)$, the language model; $\Pr(f'|e)$, the translation model, and $\Pr(s|f')$,the tokenization model.

	English		French		Russian		Spanish		Chinese	
	BLEU	METEOR	BLEU	METEOR	BLEU	METEOR	BLEU	METEOR	BLEU	METEOR
D0	42.11	0.3740	27.82	0.4535	29.80	0.4918	41.33	0.6133	32.53	0.4866
D1	42.71	0.3815	28.18	0.4620	30.47	0.4950	41.84	0.6151	33.53	0.4963
D2	42.90	0.3861	28.25	0.4676	**31.01**	0.4994	**42.18**	**0.6165**	33.76	0.4988
D3	41.01	0.3816	27.96	0.4658	30.47	0.4958	41.75	0.6147	32.53	0.4902
ATB	**43.11**	**0.3880**	**28.26**	**0.4690**	31.00	0.5001	42.03	0.6156	**33.93**	**0.5013**
D3*	42.29	0.3849	28.02	0.4615	31.00	**0.5007**	41.45	0.6147	33.73	0.5004

Table 3. SMT results for systems trained on combined schemes

We use the lattice decoding functionality at Moses (Koehn et al., 2007), which uses an approximate variation of this model through maximum entropy. Moses uses Python Lattice Format (PLF) to represent the lattice input. When Moses translates input encoded as a word lattice, the translation it chooses maximizes the translation probability along any path in the input. In the case of confusion networks, however, this means maximizing the translation probability along all distinct tokenization options for each surface form. We build the lattice out of the testing set tokenized with the six tokenization schemes. We use a customized version of the tools used at the (Salloum and Habash, 2012) paper (acquired through personal communication), to encode the lattice in the PLF format.

The results, presented at Table 4, show statistically significant improvement relative to the baselines of the scheme-specific systems, and a statistically significant improvement also relative to the simple combined schemes approach. To better under-

	English		French		Russian		Spanish		Chinese	
	BLEU	METEOR	BLEU	METEOR	BLEU	METEOR	BLEU	METEOR	BLEU	METEOR
Lattice Input	**43.33**	0.3860	**28.59**	**0.470**	**31.28**	**0.5033**	**42.31**	**0.6185**	**34.03**	**0.5016**

Table 4. SMT results for lattice based testing input

stand the resulting optimal tokenization choices, we calculate their similarity against all schemes. We observe that circa 92% of the selected optimal tokenizations are similar to D2, for all five languages. ATB is also very similar to the selected optimal tokens, with average of 91%. The next most similar scheme is D1 (around 90%) then D0 (around 84.5%) and finally D3 (around 69%).

D0	EDw	AlmHkmp	AldA}mp	lltHkym	,	lAhAy
ATB	EDw	AlmHkmp	AldA}mp	l+ AltHkym	,	lAhAy
D3	EDw	Al+ mHkmp	Al+ dA}mp	l+ Al+ tHkym	,	lAhAy
Lattice	EDw	Al+ mHkmp	AldA}mp	l+ AltHkym	,	lAhAy
English	Member of	the permanent court		of arbitration	,	the Hague

Table 5. An example of the resulting lattice tokenization

Table 5 shows an example of the lattice-based tokenization, compared to various other tokenization schemes. The lattice output maintains the definite article Al+ "the" with the words AldA}mp "permanent" and AltHkym "arbitration", while segmenting the article for the word Al+ mHkmp "the court", matching the pattern regarding the definite article "the" at the English sentence.

Error Analysis: Definite Article Behavior The ratio of the definite article Al "the", which is tokenized only at the D3 scheme, for the lattice tokenization relative to the D3 tokenization is 11.7% only.This can be the actual optimal behavior statistically (baseline systems show that D3 performs lower than ATB, the closest scheme in verbosity). This behavior can also be attributed to biases in the combined-schemes training corpus against D3-specific tokens.

6. Learning Optimal Tokenization

The models presented thus far show a significant performance improvement, whether for the combined-schemes approach or for the lattice approach, with about 1.4 BLEU points. For any interestingly large datasets, however, these approaches have limitations to their extent of applicability.

6.1. Motivation and Approach

Despite the successful SMT performance boost for the presented approaches, the execution time for the various involved processes make these models relatively challenging for interestingly large corpora. Some of these processes are executed offline, like training. However, other computationally expensive online processes, like the lattice decoding, hinders the application of the lattice approach severely.

The intuition here is to push these computationally-heavy processes offline. Since the lattice decoding is one of the most demanding processes computationally in the presented pipeline, we propose a model that learns the optimal tokenization choices generated from the lattice decoding process. This model can then be used independently to generate the most relevant word-level tokenization choices. The learning

process is based on the best-paths generated from the lattice, so in effect, the learning process will be unsupervised for there is no need for manually tokenized gold data.

6.2. Machine Learning Process

The machine learning model is intended to provide the optimal tokenization for each word in the testing set, having the model trained on the data generated from the lattice decoding. We approach this problem by learning the optimal tokenization scheme tag for each word, rather than the actual lexical tokenization, from the lattice results. We then apply this model to the testing words. The resulting tags are then used to get the corresponding actual tokenization through a lookup table. The input to the lookup table is the scheme tag and surface form, while the output is the corresponding actual tokenization. We used Conditional Random Fields (CRF) for the learning algorithm, with each line as input sequence. The features we use include the surface form, lemma, part of speech tag (POS), and a boolean mask indicating the presence of the different types of proclitics and enclitics (question, conjunction, preposition, article, among others).

The tokenization options for each surface word are not mutually exclusive, that is, the resulting tokenizations from the different schemes for the same surface form might be similar. The tokenization options for the word "AlmHkmp" mentioned previously are the same for D0, D1, D2, and ATB, which is the same as the surface word. The only different tokenization option is for D3; by splitting the definite article: "Al+ mHkmp". Moreover, as covered at the Arabic tokenization schemes section, the tokenization schemes vary by verbosity as follows (increasing verbosity):

D0 <D1 <D2 <ATB <D3

Since the tokenization options might be similar across several tokenization schemes, we consider the verbosity of the selected scheme label in case the surface word has similar tokenization options to other schemes. The system can assign the most/least verbose scheme, which will be analyzed and discussed at the next section.

6.3. Experiments and Analysis

We apply the CRF approach on the Arabic-English system. We use a dataset of 50K lines (around 1.3M words) to train the system. We apply the lattice pipeline discussed earlier, and obtain the best paths resulting from the lattice decoding through Moses, and use these as the training set. Instead of using the actual training labels for the system evaluation, which might be prone to biases due to different tokenization schemes having similar outputs, we use the actual generated tokenized words, through simple accuracy scores. We then input the resulting tokenized content to the MT system, and use the BLEU score as another evaluation metric.

We use ATB as the baseline for our analysis, since it's the most widely used tokenization scheme for Arabic in literature, and it had the best performance in our baseline systems (along with D3*). Table 6 shows the evaluation scores for the machine

learning system. The system shows a clear improvement over the baseline. We further conducted another experiment regarding the verbosity ordering of the tokenization schemes. The result shows a clear improvement for the decreasing verbosity order (at 93.8%) relative to increasing verbosity (at 90.9%). The execution time for the learning

Evaluation Metric	Score
ATB baseline accuracy	91.73%
ATB baseline MT BLEU score (English)	41.91
CRF accuracy	93.80%
CRF MT BLEU score (English)	42.84

Table 6. The performance of the learnt tokenizer

approach is around 4X less than that of the lattice approach, considering the shared processes with the lattice approach as part of the offline tasks. The resulting BLEU score is 42.84; about 0.9 higher than the ATB baseline; a statistically significant boost, and 0.5 BLEU points lower than the lattice approach. These numbers make the case for using the learnt tokenizer, given the complexity of the lattice approach.

7. Conclusion

We presented several tokenization models that enhance the overall Statistical Machine Translation performance. We applied these models to Arabic and were able to conclude that combining different tokenization options at the training phase of the SMT system enhances the overall performance. We were also able to prove that considering all tokenization options at the decoding phase of the testing set further enhances the performance. We didn't see a significant behavior shift across the different languages when it comes to the schemes combination methods, but the scheme we suggested, D3*, proved efficient for Russian and Chinese. We finally presented a learning approach to model the optimal tokenization options based on the lattice decoding, to facilitate a more practical tokenization process.

Bibliography

Abdelali, Ahmed, Kareem Darwish, Nadir Durrani, and Hamdy Mubarak. Farasa: A Fast and Furious Segmenter for Arabic. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 11–16, San Diego, California, 2016. URL `http://www.aclweb.org/anthology/N16-3003`.

Banerjee, Satanjeev and Alon Lavie. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages

65–72, Ann Arbor, Michigan, June 2005. URL http://www.aclweb.org/anthology/W/W05/W05-0909.

Diab, Mona, Kadri Hacioglu, and Daniel Jurafsky. *Automatic Processing of Modern Standard Arabic Text*, pages 159–179. Springer Netherlands, Dordrecht, 2007. ISBN 978-1-4020-6046-5. doi: 10.1007/978-1-4020-6046-5_9. URL http://dx.doi.org/10.1007/978-1-4020-6046-5_9.

Dyer, Chris. Using a Maximum Entropy Model to Build Segmentation Lattices for MT. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 406–414, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-41-1. URL http://dl.acm.org/citation.cfm?id=1620754.1620814.

Dyer, Christopher, Smaranda Muresan, and Philip Resnik. Generalizing Word Lattice Translation. In *Proceedings of ACL-08: HLT*, Columbus, Ohio, 2008.

Eisele, Andreas and Yu Chen. MultiUN: A Multilingual Corpus from United Nation Documents. In Tapias, Daniel, Mike Rosner, Stelios Piperidis, Jan Odjik, Joseph Mariani, Bente Maegaard, Khalid Choukri, and Nicoletta Calzolari (Conference Chair), editors, *Proceedings of the Seventh conference on International Language Resources and Evaluation*, pages 2868–2872. European Language Resources Association (ELRA), 5 2010.

Elming, Jakob and Nizar Habash. Combination of Statistical Word Alignments Based on Multiple Preprocessing Schemes. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 25–28, Rochester, New York, April 2007. URL http://www.aclweb.org/anthology/N/N07/N07-2007.

Habash, Nizar and Jun Hu. Improving Arabic-Chinese statistical machine translation using English as pivot language. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 173–181. Association for Computational Linguistics, 2009.

Habash, Nizar and Owen Rambow. 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, 2005. URL http://www.aclweb.org/anthology/P/P05/P05-1071.

Habash, Nizar and Fatiha Sadat. Arabic Preprocessing Schemes for Statistical Machine Translation. pages 49–52, New York, NY, 2006.

Heafield, Kenneth, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. Scalable Modified Kneser-Ney Language Model Estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 690–696, Sofia, Bulgaria, August 2013. URL http://kheafield.com/professional/edinburgh/estimate_paper.pdf.

Koehn, Philipp. Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain, July 2004. Association for Computational Linguistics.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of ACL, demo session*, Prague, Czech Republic, 2007.

Lee, Young-Suk, Kishore Papineni, Salim Roukos, Ossama Emam, and Hany Hassan. Language Model Based Arabic Word Segmentation. pages 399–406, Sapporo, Japan, 2003.

Maamouri, Mohamed, Ann Bies, and Tim Buckwalter. The Penn Arabic Treebank : Building a Largescale Annotated Arabic Vorpus. In *Conference on Arabic Language Resources and Tools*. NEMLAR, 2004.

Mermer, Coşkun. Unsupervised Search for the Optimal Segmentation for Statistical Machine Translation. In *Proceedings of the ACL 2010 Student Research Workshop*, ACLstudent '10, pages 31–36, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1858913.1858919`.

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

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 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, 2002.

Pasha, Arfath, Mohamed Al-Badrashiny, Ahmed El Kholy, Ramy Eskander, Mona Diab, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic. In *In Proceedings of LREC*, Reykjavik, Iceland, 2014.

Popović, Maja and Hermann Ney. Towards the Use of Word Stems and Suffixes for Statistical Machine Translation. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)*, pages 1585–1588, Lisbon, Portugal, May 2004.

Roth, Ryan, Owen Rambow, Nizar Habash, Mona Diab, and Cynthia Rudin. 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, 2008. URL `http://www.aclweb.org/anthology/P/P08/P08-2030`.

Sadat, Fatiha and Nizar Habash. Combination of Arabic Preprocessing Schemes for Statistical Machine Translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 1–8, Sydney, Australia, July 2006. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P/P06/P06-1001`.

Salloum, Wael and Nizar Habash. Elissa: A Dialectal to Standard Arabic Machine Translation System. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012): Demonstration Papers*, pages 385–392, Mumbai, India, 2012.

Zhang, Yuqi, Richard Zens, and Hermann Ney. Improved chunk-level reordering for statistical machine translation. In *IWSLT*, pages 21–28, 2007.

Address for correspondence:
Nasser Zalmout
`nasser.zalmout@nyu.edu`
New York University Abu Dhabi, United Arab Emirates