What Matters Most in Morphologically Segmented SMT Models?

Mohammad Salameh[†] **Colin Cherry**[‡] **Grzegorz Kondrak**[†]

[†]Department of Computing Science
University of Alberta
Edmonton, AB, T6G 2E8, Canada
{msalameh,gkondrak}@ualberta.ca

[‡]National Research Council Canada
1200 Montreal Road
Ottawa, ON, K1A 0R6, Canada
Colin.Cherry@nrc-cnrc.gc.ca

Abstract

Morphological segmentation is an effective strategy for addressing difficulties caused by morphological complexity. In this study, we use an English-to-Arabic test bed to determine what steps and components of a phrase-based statistical machine translation pipeline benefit the most from segmenting the target language. We test several scenarios that differ primarily in when desegmentation is applied, showing that the most important criterion for success in segmentation is to allow the system to build target words from morphemes that span phrase boundaries. We also investigate the impact of segmented and unsegmented target language models (LMs) on translation quality. We show that an unsegmented LM is helpful according to BLEU score, but also leads to a drop in the overall usage of compositional morphology, bringing it to well below the amount observed in human references.

1 Introduction

It is well known that morphological segmentation can improve statistical machine translation (SMT). By splitting relevant morphological affixes into independent tokens, segmentation has repeatedly been shown to improve translation into or out of morphologically complex languages. Segmentation as a preprocessing step brings several benefits to translation:

- **Correspondence** with morphologically simple languages, such as English is improved. In Figure 1, segmenting *bsyArth* allows one-to-one links for "with", "his" and "car".

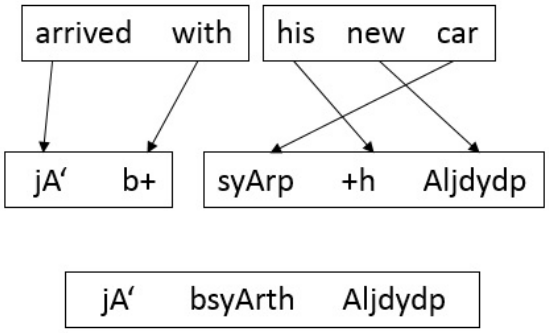

Figure 1: An illustration of one-to-one correspondence between Arabic morphemes and English words. Arabic text is segmented using the PATB tokenization scheme, and shown in Buckwalter transliteration.

- By building models over morphemes, rather than words, **data sparsity** is reduced.
- By allowing morphemes with clear syntactic roles to be translated independently, we increase our **expressive power** by creating new lexical translations. For example, using the two phrase-pairs in Figure 1 results in a new word after desegmentation (b+ syArp +h $\Rightarrow$ bsyArth), which might not have existed in the training data.

However, there is also a price to be paid. While morpheme-level models are more resistant to data sparsity, they account for less context than word-level models, make stronger independence assumptions, and they are less efficient statistically, in that they devote probability mass to sequences containing illegal words. Furthermore, when segmentation is applied to the target language, the process must be

Proceedings of SSST-9, Ninth Workshop on Syntax, Semantics and Structure in Statistical Translation, pages 65–73,
Denver, Colorado, June 4, 2015.

reversed at the end of the pipeline to present the output in a readable format. This **desegmentation** step complicates our pipeline, and can introduce errors.

Our work is inspired by two recent contributions that attempt to combine the advantages of word- and morpheme-based models. Luong et al. (2010) combine word and morpheme views in a desegmented phrase table, allowing morphemes to reduce sparsity while words expand context, and eliminating the need for a separate desegmentation step. Their word-boundary-aware morpheme-level phrase extraction technique restricts phrase boundaries so that no target phrase can begin with a suffix or end with a prefix. This allows them to desegment each target phrase independently, enabling the use of both word- and morpheme-level language models during decoding. However, this phrase-table desegmentation approach lacks the expressive power that comes from translating morphemes independently.

More recently, Salameh et al. (2014) propose a lattice desegmentation approach, which comes close to combining all the advantages of word and morpheme views. By desegmenting a lattice that compactly represents many translation options, and rescoring it with a word-level language model, they avoid restricting the phrase table. However, by delaying desegmentation until rescoring, the approach loses Luong et al. (2010)'s advantage of full decoder integration.

In this paper, we present an experimental study of English-to-Arabic translation that is designed to better understand the impact of various trade-offs when translating into a morphologically segmented target language, and to identify what aspects of segmentation are most beneficial to translation. The benefits of segmentation can impact several components in the SMT pipeline: the alignment model, the translation table, and the various language and translation models. Throughout this study, we investigate the effect of varying the point in the SMT pipeline where the segmentation is reversed. In addition, we attempt to combine word- and morpheme-level models within the decoder as much as possible.

Our experimental study provides three novel insights. First, we present strong evidence indicating that the ability to build target words across phrase boundaries is the most important property of target language segmentation. This implies that phrase table desegmentation, the only published desegmentation technique that has been fully integrated into decoding, gives up segmentation's primary advantage. Second, we draw a previously unobserved connection between the use of an unsegmented LM and the decoder's overall use of compositional morphology; we show that although unsegmented LMs tend to increase BLEU score, they also reduce the system's use of morphological affixes to well below that of a human. Finally, we present the first direct comparison between phrase table desegmentation (Luong et al., 2010) and lattice desegmentation (Salameh et al., 2014).

2 Background

Our work builds on earlier studies of automatic morphological segmentation and its impact on SMT. There are many ways to segment syntactically relevant affixes from stems. Supervised techniques may either pass through an intermediate morphological analysis (Habash et al., 2009), or directly segment the character stream (Green and DeNero, 2012); recent work on supervised Arabic segmentation focuses primarily on adaptation to dialects (Habash et al., 2013; Monroe et al., 2014). There are also a host of unsupervised techniques (Creutz and Lagus, 2005; Lee et al., 2011; Sirts and Goldwater, 2013), which provide valuable language portability, but which generally fall behind supervised methods when labeled data is available.

There is a large body of work studying the best form of segmentation when translating from a morphologically complex source language (Sadat and Habash, 2006; Stallard et al., 2012), where the segmentation can be used as a simple preprocessing step, or to create an input lattice (Dyer et al., 2008). Recently, there has been a growing interest in segmentation on the target side (Oflazer and Durgar El-Kahlout, 2007), which introduces a question of how to perform proper desegmentation (Badr et al., 2008). El Kholy and Habash (2012) have conducted a thorough exploration of the various segmentation and desegmentation options for English to Arabic translation, and we follow their work when designing our test bed.

Method	Unsegmented	Alignment Deseg.	Phrase Table Deseg.	One-best Deseg.	Lattice Deseg.
Desegment before:	Never segment	Phrase extraction	Decoding	Evaluation	Evaluation
Alignment model	Word	Morph	Morph	Morph	Morph
Lexical weights	Word	Word	Morph	Morph	Morph
Language model	Word	Word	Word	Morph	Morph + Word
Tuning	Word	Word	Word	Morph	Morph then Word
Flexible boundaries?	No	No	No	Yes	Yes

Table 1: Desegmentation scenarios and their effect on the components of a typical SMT system.

3 Methods

When translating into a segmented target language, such as Arabic, the segmentation will need to eventually be reversed for the output to be readable. The key insight driving our experiments is that by varying the point in the SMT pipeline where this reversal occurs, we can alter which models are based on morphemes and which are based on words, and thereby determine which components most benefit from segmentation. We assume a phrase-based SMT architecture similar to that of Moses (Koehn et al., 2007), but most of our observations hold for hierarchical and tree-based models. In all of our approaches, we desegment using a mapping table that counts the segmentations performed on the target side of our training data. The table uses counts of word-segmentation pairs to map each morpheme sequence back to its most likely unsegmented word form. We back off to manually crafted rules in cases where the segmented form does not exist in the mapping table (El Kholy and Habash, 2012).

Table 1 summarizes the effect of the desegmentation point on the components of a typical SMT system, indicating which components are built using morphemes and which are built using words. Most components should be familiar, but the last row introduces **flexible boundaries**, a concept that will be central to our study. This property of the phrase table indicates whether phrases can have unattached affixes at their left or right boundaries. Systems without flexible boundaries cannot combine morphemes across phrases to create translations that were not already seen in the parallel text; as such, this property has a large impact on a system's expressive power.

We describe our comparison systems in turn, each corresponding to a column in Table 1. We also describe a segmented language model feature, which can be added to any system that uses a word-level phrase table.

3.1 Baselines

We rely on two main baselines to evaluate what matters most in segmented models. An **unsegmented** system leaves the Arabic target unsegmented and uses an unsegmented language model. This model suffers from data sparsity and poor English-Arabic word correspondence. The decoder always outputs morphologically correct Arabic words, as it does not require a desegmentation step.

Meanwhile, **one-best desegmentation** segments the Arabic target language before training begins, and the decoder's output is generated in segmented form. As a post-processing step, the one-best output is desegmented using a mapping table and desegmentation rules. All of the component models used during decoding are based on morphemes instead of words. The segmented models are intended to help alleviate data sparsity and improve token correspondence. Unlike the unsegmented system, this system requires a desegmentation step, which can produce morphologically incorrect words.

3.2 Alignment Desegmentation

Our unsupervised alignment models (Brown et al., 1993; Och and Ney, 2003) are sensitive both to poor word-to-word correspondence and to data sparsity issues. They are also at the very start of the SMT pipeline; they impact nearly all other downstream models. Therefore, it would be reasonable to suspect that the primary benefit of segmentation could come from improved word alignment. Alignment desegmentation allows us to test this theory by desegmenting immediately after alignment.

More specifically, we segment the target side as pre-processing. After word alignment, we replace

the segmented Arabic training data with its unsegmented form. Note that this desegmentation is perfect, as we can always refer to the original sentence to resolve any ambiguities. This is accompanied by desegmenting alignment links by replacing each morpheme index with the index of the unsegmented word that now contains the morpheme. As one would expect, this leads to an increase in the number of one-to-many alignments. Training is then resumed with these links and the unsegmented target. Other than having its alignment model benefit from segmentation, this system has the same properties of an unsegmented system: all remaining component models are based on words. Since all morphemes are desegmented well before decoding begins, it clearly cannot use flexible boundaries to build new words.

3.3 Phrase Table Desegmentation

Our next desegmentation point is after phrase extraction, resulting in a system where we segment the text, align the morphemes, perform phrase extraction over morphemes, and then desegment the resulting tables. Following Luong et al. (2010), we first remove all phrases that have target sides with flexible boundaries, which allows us to desegment each remaining target phrase independently. The result is a desegmented phase table. Note that we leave the various scores associated with each phrase-pair unchanged.

This model is similar to alignment desegmentation described in the previous section in that all remaining components and operations are based on words. However, there are two key differences. First, the lexical weights of each phrase are calculated over morphemes rather than words. Second, the phrase-length limit is applied at the morpheme level rather than at the word level. We use this scenario to test the utility of morpheme-level lexical weights.

This system is related to, but not identical to the work of Luong et al. (2010). Their system actually merges tables from an unsegmented model with those from phrase table desegmentation; they investigate a number of methods to combine the scores across tables. In addition, they incorporate both segmented and unsegmented language models, which is a difference that we address in the next section.

3.4 Segmented LM Scoring in Desegmented Models

Both alignment desegmentation and phrase table desegmentation rely on an unsegmented language model, as they naturally decode directly into a desegmented target language. We experiment with augmenting both of these systems with an extra feature: a segmented language model. For each Arabic target word, we add its segmented form to the phrase table as an extra factor (Koehn and Hoang, 2007). We insert this factor after phrase extraction, so it has no impact on alignment or the calculation of translation model scores. The factor merely gives us access to the segmented morphemes during decoding. The decoder uses this factor to apply a segmented language model during each hypothesis extension.

Although the segmented language model spans a shorter context, its scores benefit from the reduced data sparsity that comes from modeling morphemes. In particular, it can unveil whether attaching two hypotheses is grammatical. For example, the unsegmented language model score for the consecutive target phrases [kl m$AklnA] *"all our problems"* [wxlAfAtnA] *"and conflicts"* is relatively low. Scoring their segmented representation [kl m$Akl +nA] [w+ xlAfAt +nA] leads to a more optimistic score, as the segmented language model assesses the morpheme sequence using 4-grams and trigrams, while the unsegmented model scores the word sequence with unigrams and bigrams.

3.5 Lattice Desegmentation

We re-implement the Lattice Desegmentation technique proposed by Salameh et al. (2014), and place it in Table 1 for reference. A system built entirely over morphemes outputs a pruned lattice that compactly represents its hypothesis space. This lattice is then desegmented by composing it with a finite state transducer that maps morpheme sequences into words. By rescoring the desegmented lattice with new features, the system benefits from having both a segmented and desegmented view of the search space. The added features include discontiguity features, as well as an unsegmented language model. The discontiguity features indicate whether a desegmented word came from one contiguous morpheme sequence, two discontiguous sequences, or more.

4 Experimental Setup

We train our English-to-Arabic system using 1.49 million sentence pairs drawn from the NIST 2012 training set, excluding the UN data. This training set contains about 40 million Arabic tokens before segmentation, and 47 million after segmentation. We tune on the NIST 2004 evaluation set (1353 sentences) and evaluate on NIST 2005 (1056 sentences). We also report a second test, which tunes on the NIST 2006 evaluation set (1664 sentences) and evaluates on NIST 2008 (1360 sentences) and 2009 (1313 sentences). NIST 2004 and 2005 datasets have sentences from newswire, while NIST 2006/2008/2009 have sentences drawn from newswire and the web. These evaluation sets are intended for Arabic-to-English translation, and therefore have multiple English references. As we are translating into Arabic, we select the first English reference to use as our source text, and use the Arabic source as our single reference translation.

4.1 Segmentation

For Arabic, morphological segmentation is performed by MADA 3.2 (Habash et al., 2009), using the Penn Arabic Treebank (PATB) segmentation scheme as recommended by El Kholy and Habash (2012). For both segmented and unsegmented Arabic, we further normalize the script by converting different forms of Alif and Ya to bare Alif and dotless Ya. In order to generate the desegmentation table, we analyze the MADA segmentations from the Arabic side of the parallel training data to collect mappings from morpheme sequences to surface forms.

4.2 Systems

We align the parallel data with GIZA++ (Och et al., 2003) and decode using Moses (Koehn et al., 2007). The decoder's log-linear model includes a standard feature set. Four translation model features encode phrase translation probabilities and lexical weights in both directions. Seven distortion features encode a standard distortion penalty as well as a bidirectional lexicalized reordering model. A KN-smoothed 5-gram language model is trained on the target side of the parallel data with SRILM (Stolcke, 2002). Finally, we include word and phrase penalties. The decoder uses Moses' default search parameters, except that the maximum phrase length is set to 8. The decoder's log-linear model is tuned with MERT (Och, 2003). Following Salameh et al. (2014), the tuning of the re-ranking models for lattice desegmentation is performed using a lattice variant of hope-fear MIRA (Cherry and Foster, 2012); lattices are pruned to a density of 50 edges per word before re-ranking. We evaluate our system using BLEU (Papineni et al., 2002).

5 Results

Table 2 shows the results of our translation quality experiments. In previous sections, we mentioned several factors that might contribute to the quality improvements found with segmented models. Beyond the raw ranking of systems, we can use the commonalities and differences between these systems to draw some broad conclusions of what aspects of a segmented system are most important.

5.1 Decoder Integration

Lattice Desegmentation performs best overall, which is not entirely surprising, as it has access to all of the information present in the other systems. Notably, it outperforms Phrase Table Desegmentation; this is the first time to our knowledge that the two have been compared directly.

The main disadvantage of Lattice Deseg, which is not present in Alignment and Phrase Table Deseg, is the lack of decoder integration of its unsegmented view of the target; instead, it is handled by re-ranking a lattice in post-processing. In fact, the top two systems, Lattice Deseg and 1-Best Deseg, are also the only two systems without access to unsegmented information in the decoder. This suggests that the benefits of decoder integration are not sufficient to overcome the trade-offs currently demanded by integration.

5.2 Flexible Boundaries

What is perhaps more surprising is that neither Alignment Deseg nor Phrase Table Deseg are able to match the 1-best Deseg scenario. With the benefit of added segmented language models, both of these systems have access to almost all 1-best Deseg's information and more, yet they fail to match

Model	mt05	mt08	mt09
Unsegmented	32.8	15.0	19.0
Alignment Deseg.	33.4	15.4	19.1
with Segmented LM	33.7	15.4	19.4
Phrase Table Deseg.	33.4	15.5	19.3
with Segmented LM	33.6	15.6	19.7
1-best Deseg.	33.7	15.7	20.2
without flexible boundaries	32.9	15.4	19.4
Lattice Deseg.	34.3	16.4	20.5

Table 2: BLEU scores on on each of the methods described in section 3 . MT05 results are tuned using NIST MT04. Results on NIST MT08 and MT09 datasets are tuned on MT06 dataset.

its translation quality in every test. What both systems lack with respect to 1-best Deseg is flexible phrase boundaries, which allow the creation of new translations across phrases. To confirm the importance of flexible boundaries, we created a new version of 1-best Deseg by pruning all phrases with flexible boundaries from the phrase table, and then re-tuning. The resulting system loses 0.6 BLEU on average, which is more than half of the 0.9 difference between Unsegmented and 1-best Deseg. We conclude that flexible boundaries are one of the most important aspects of a segmentation scenario.

5.3 Language Models

Both Align Deseg and Phrase Table Deseg show consistent, albeit small, improvements from the addition of a segmented LM. In order to assess the importance of the unsegmented LM, we consider 1-best Deseg without flexible boundaries, and Phrase Table Deseg with Segmented LM. These two systems have exactly the same output space, as their respective phrase tables are constructed from morpheme-level phrase extraction followed by pruning flexible boundaries. Furthermore, both systems use a segmented LM and lexical weights built over morphemes. Their only differences are that Phrase Table Deseg uses an unsegmented LM and unsegmented tuning, resulting in BLEU scores that are higher by 0.4 on average. Similarly, a unsegmented LM is one of the main differences between Lattice Deseg and 1-best Deseg, with the others being unsegmented tuning and discontiguity features. Although we have not isolated the unsegmented LM perfectly, these results indicate that it is valuable.

5.4 Lexical Weights

The primary difference between Alignment Deseg and Phrase Table Deseg is that the latter uses morpheme-level lexical weights.[1] Without a segmented LM, we see a 0.1 average BLEU advantage for Phrase Table Deseg, increasing to 0.2 when a segmented LM is included. Unfortunately, these improvements are not consistent across test sets. This suggests that there may be an advantage from morpheme-based lexical weights, but it is certainly not large.

6 Analysis

Our translation quality comparison indicates that flexible boundaries are the most important property of a target segmentation scenario, so we examined them in greater detail. Phrase pairs with flexible boundaries account for roughly 12% of phrases used in the final output of our 1-Best Deseg system.

We performed a detailed analysis to see if the flexible boundaries were used to produce novel words; that is, words that were not seen in the target side of the training data. Roughly 3% of the desegmented types generated by the 1-best-desegmentation system are novel. We randomly selected 40 novel words from each test set to analyze manually. First, none of these desegmented words appear in the reference, and therefore, they have no positive impact on BLEU. Furthermore, 64 of the 120 selected words violate the morphological rules of Arabic. Looking instead at the novel words in the reference, only 115

[1]The other difference is the calculation of the phrase length limit, which favors Alignment Deseg, as its word-based limit allows more phrases overall.

Model	mt05	mt08	mt09
Reference	15.9%	18.1%	18.9%
Unsegmented	12.0%	12.2%	12.6%
Alignment Deseg.	11.6%	11.0%	11.8%
with Segmented LM	11.7%	11.2%	12.0%
Phrase Table Deseg.	11.3%	10.1%	11.2%
with Segmented LM	11.6%	10.5%	11.4%
1-best Deseg.	16.1%	18.2%	19.2%
without flexible boundaries	14.2%	14.7%	15.4%
Lattice Deseg.	10.0%	11.5%	12.2%

Table 3: Percentage of words in the SMT output that have non-identity morphological segmentations.

reference words could not be found in the Arabic side of our training data. Of these, only 37 could be could be constructed from morphemes found in our training set. This means that there is only a small number of opportunities to better match the reference by producing a novel word. Together, these two pieces of analysis strongly suggest that the advantage of flexible boundaries comes from creating new translation options for a given source sequence, rather than from creating novel words.

We were able to compute statistics on flexible boundaries for only two of our systems, because the other three disallow them entirely. In order to characterize all five systems, along with the human references, we measured overall affix usage by counting decomposable words. Table 3 shows the percentage of words in the Arabic translations that have non-identity morphological segmentations when processed by MADA. In terms of affix usage, the 1-best Deseg method tracks the Reference very closely, while all remaining scenarios show a substantial drop in usage of decomposable words. Most surprisingly, Lattice Deseg is included in this group, even though its BLEU scores are higher than 1-best Deseg. Since 1-Best Deseg's most prominent characteristic is its lack of an unsegmented LM, this suggests that unsegmented LMs may dramatically impact affix usage. Note that flexible boundaries do not (fully) account for the gap in affix usage, as the 1-best Deseg still has noticeably higher usage of decomposable words, even with flexible boundaries removed. This implies that Lattice Deseg and the various fully integrated desegmentations could be improved by attempting to directly manipulate their usage of decomposable words, perhaps through a specialized feature.

As a final piece of analysis, we also investigated the impact of different n-gram orders for segmented LMs. Most of the scenarios proposed here add an unsegmented LM to a segmented system, and the most obvious advantage of an unsegmented LM is that it accounts for more context than a segmented LM. However, this only holds if we force both LMs to have the same n-gram order. To see if higher order segmented LMs would improve translation, we experimented with different n-gram orders for our 1-best Deseg system. As we increased the segmented n-gram order from 5 to 8, we saw no improvement over the 5-gram LM used throughout this paper. In fact, BLEU score began to drop after $n = 6$. This suggests that the advantage of adding an unsegmented LM cannot be emulated by increasing the order of the segmented LM.

7 Conclusion

We have presented an experimental study on translation into segmented target languages by creating models that apply desegmentation at different points in the translation pipeline. We have provided evidence that access to phrases with flexible boundaries is a crucial property for a successful segmentation approach. We have also examined the impact of unsegmented LMs, showing that although they are helpful according to BLEU, they also hinder the generation of morphologically-complex words. This suggests that current methods could be improved by attempting to increase their use of morphological affixes.

References

Ibrahim Badr, Rabih Zbib, and James Glass. 2008. Segmentation for English-to-Arabic statistical machine translation. In *Proceedings of ACL*, pages 153–156.

Peter F Brown, Vincent J Della Pietra, Stephen A Della Pietra, and Robert L Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of HLT-NAACL*, Montreal, Canada, June.

Mathias Creutz and Krista Lagus. 2005. Inducing the morphological lexicon of a natural language from unannotated text. In *In Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR05*, pages 106–113.

Christopher Dyer, Smaranda Muresan, and Philip Resnik. 2008. Generalizing word lattice translation. In *Proceedings of ACL-08: HLT*, pages 1012–1020, Columbus, Ohio, June.

Ahmed El Kholy and Nizar Habash. 2012. Orthographic and morphological processing for English—Arabic statistical machine translation. *Machine Translation*, 26(1-2):25–45, March.

Spence Green and John DeNero. 2012. A class-based agreement model for generating accurately inflected translations. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 146–155, Jeju Island, Korea, July.

Nizar Habash, Owen Rambow, and Ryan Roth. 2009. Mada+tokan: A toolkit for Arabic tokenization, diacritization, morphological disambiguation, POS tagging, stemming and lemmatization. In Khalid Choukri and Bente Maegaard, editors, *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, Cairo, Egypt, April. The MEDAR Consortium.

Nizar Habash, Ryan Roth, Owen Rambow, Ramy Eskander, and Nadi Tomeh. 2013. Morphological analysis and disambiguation for dialectal arabic. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 426–432, Atlanta, Georgia, June.

Philipp Koehn and Hieu Hoang. 2007. Factored translation models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 868–876, Prague, Czech Republic, June.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan 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.

Yoong Keok Lee, Aria Haghighi, and Regina Barzilay. 2011. Modeling syntactic context improves morphological segmentation. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 1–9, Portland, Oregon, USA, June.

Minh-Thang Luong, Preslav Nakov, and Min-Yen Kan. 2010. A hybrid morpheme-word representation for machine translation of morphologically rich languages. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 148–157, Cambridge, MA, October.

Will Monroe, Spence Green, and Christopher D. Manning. 2014. Word segmentation of informal arabic with domain adaptation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 206–211, Baltimore, Maryland, June.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

Franz Josef Och, Hermann Ney, Franz Josef, and Och Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29.

Franz Joseph Och. 2003. Minimum error rate training for statistical machine translation. In *Proceedings of ACL*, pages 160–167.

Kemal Oflazer and Ilknur Durgar El-Kahlout. 2007. Exploring different representational units in English-to-Turkish statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 25–32, Prague, Czech Republic, June.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Fatiha Sadat and Nizar Habash. 2006. Combination of Arabic preprocessing schemes for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 1–8.

Mohammad Salameh, Colin Cherry, and Grzegorz Kondrak. 2014. Lattice desegmentation for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 100–110.

Kairit Sirts and Sharon Goldwater. 2013. Minimally-supervised morphological segmentation using adaptor grammars. *TACL*, 1:255–266.

David Stallard, Jacob Devlin, Michael Kayser, Yoong Keok Lee, and Regina Barzilay. 2012. Unsupervised morphology rivals supervised morphology for arabic mt. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ACL '12, pages 322–327, Stroudsburg, PA, USA.

Andreas Stolcke. 2002. Srilm - an extensible language modeling toolkit. In *Intl. Conf. Spoken Language Processing*, pages 901–904.