Hebrew Morphological Preprocessing for Statistical Machine Translation

Nimesh Singh and **Nizar Habash**
Center for Computational Learning Systems
Columbia University
nks2118@columbia.edu
habash@ccls.columbia.edu

Abstract

This paper presents a range of preprocessing solutions for Hebrew-English statistical machine translation. Our best system, using a morphological analyzer, increases 3.5 BLEU points over a no-tokenization baseline on a blind test set. The next best system uses Morfessor, an unsupervised morphological segmenter, and obtains almost 3.0 BLEU points over the baseline.

1 Introduction

Much research in statistical machine translation (SMT) has shown the importance of morphological preprocessing (aka, tokenization, segmentation) on translation quality. The common wisdom in the field is that such preprocessing helps, especially for morphologically rich languages, such as Arabic, Spanish or Finnish, because it reduces model sparsity and increases source-target symmetry (particularly when the target is morphologically poor, as in English). However, the value of preprocessing generally decreases with added training data, and is highly dependent on the language pair and particular preprocessing approach (Popović and Ney, 2004; Lee, 2004; Goldwater and McClosky, 2005; Habash and Sadat, 2006; Fishel and Kirik, 2010; Al-Haj and Lavie, 2012).

In this paper, we present results from a set of experiments to determine an optimal preprocessing method for Hebrew-English SMT, a language pair with limited previously published work (Lavie et al., 2004; Lembersky et al., 2011). We report on three types of preprocessing techniques using deterministic regular-expressions, unsupervised morphology learning, and morphological analysis and disambiguation. Our results show that using a morphological analyzer helps translation quality the most, followed by using an unsupervised morphological segmenter.

The paper is structured as follows: Section 2 presents relevant related work. Section 3 discusses the linguistic challenges of translating Hebrew to English. Section 4 describes the different preprocessing techniques we study. And Section 5 presents our evaluation results.

2 Related Work

A wide range of preprocessing techniques have been studied for a variety of language pairs requiring different treatments. Nießen and Ney (2004) studied the impact of various types of morpho-syntactic restructuring on German-English SMT and Popović and Ney (2004) studied the effect of splitting words into stems and suffixes on SMT into English from Spanish, Catalan and Serbian. Their results show significant error reduction when stemming is used. Koehn and Knight (2003) compared different methods for compound splitting when translating from German to English. All of their methods improve SMT quality over a no-splitting baseline; however, the methods with the highest accuracy are not the best SMT performers. Lee (2004) investigated the use of automatic alignment of POS tagged English and affix-stem segmented Arabic to determine whether affixes should be kept separate, deleted or reattached to stems. Her results show that morphological preprocessing helps, but only for the smaller corpora sizes she investigated. As size increases, the benefits diminish. Goldwater and McClosky (2005) showed that incorporating various methods for specifying morphological information in Czech-English SMT (e.g., lemmatization and different styles of seg-

mentation) improves translation quality especially when the different methods are combined. Habash and Sadat (2006) compared a variety of what they called tokenization schemes and techniques for Arabic-English SMT. Their work and that of Lee (2004) are especially relevant since Arabic is a Semitic language like Hebrew. This paper is closest in its approach to Habash and Sadat (2006). We refer to their work further below. We do not discuss efforts on translation into morphologically rich languages although similar approaches have been investigated (El Kholy and Habash, 2012a; Al-Haj and Lavie, 2012)

As for the use of unsupervised morphology in SMT, Virpioja et al. (2007) and Fishel and Kirik (2010) presented some experiments with mixed results. They suggested that language pairs different from those they studied (Danish-Finnish-Swedish and Estonian-English, respectively), may benefit from unsupervised morphology. Snyder and Barzilay (2008) presented results on learning Hebrew morphology using parallel and monolingual resources.

Until recently, there has not been much parallel Hebrew-English data (Tsvetkov and Wintner, 2010), and consequently little work on Hebrew-English SMT. Lavie et al. (2004) built a transfer-based translation system for Hebrew-English and so did Shilon et al. (2012) for translation between Hebrew and Arabic. Lembersky et al. (2011), using the above-mentioned parallel corpus, compared the behavior of different SMT systems using training data sets that vary in reference translation directionality.

To our knowledge this is the first study comparing different tokenization techniques for Hebrew-English SMT. We successfully show that unsupervised morphology segmentation helps for Hebrew-English SMT, but a more linguistically sophisticated system with a morphological analyzer does best.

3 Hebrew in the Context of SMT

We present in this section some relevant Hebrew linguistic facts. This is followed by an analysis of out-of-vocabulary errors in the baseline system described in Section 5.

3.1 Hebrew Linguistic Facts

Hebrew poses computational processing challenges typical of Semitic languages such as Arabic (Itai and Wintner, 2008; Shilon et al., 2012; Habash, 2010). Similar to Arabic, Hebrew orthography uses optional diacritics and its morphology uses both root-pattern and affixational mechanisms. Hebrew inflects for gender, number, person, state, tense and definiteness. Furthermore, Hebrew has a set of attachable clitics that are typically separate words in English, e.g., conjunctions (such as +ו *w+* 'and'),[1] prepositions (such as +ב *b+* 'in'), the definite article (+ה *h+* 'the'), or pronouns (such as +הם *+hm* 'their'). These issues contribute to a high degree of ambiguity that is a challenge to translation from Hebrew to English or to any other language. Some of these clitics undergo morphotactic transformations that only add to the words' ambiguity. For example, the sequence of the *preposition* + *article* +ה+ב *b+h+* 'in the' results in the deletion of the letter for the article: +ב *b+* 'in the'. This makes the string +ב *b+* ambiguous as 'in a' or 'in the'.[2]

The different clitics appear in a generally strict order around the base word:

conjunction
relativizer
preposition
definite article
base word
pronominal clitic.

The definite article and the pronominal clitics do not co-occur. The conjunction +ו *w+* 'and' and relativizer +ש *š+* 'that/who' can appear with all parts-of-speech (nouns, verbs, prepositions, pronouns, etc.). Prepositions are mostly nominal and the definite article is strictly nominal.[3] Pronominal clitics can attach to nouns and prepositions and infrequently to verbs (archaic).[4] For example, the word בשורה *bšwrh* has the following possible nominal analyses among others: בשורה *bšwrh* 'gospel', ב+שורה *b+šwrh* 'in+(a/the) line', and ב+שור+ה *b+šwr+h* 'in her bull [lit. in+bull+her]'.

[1]The following Hebrew 1-to-1 transliteration is used (in Hebrew lexicographic order): *abgdhwzxTiklmns'pcqršt*. All examples are undiacritized and final forms are not distinguished from non-final forms.

[2]The deleted article survives as a vowel which is written as an optional diacritic.

[3]Infinitive verbs in Hebrew have a prefix +ל *l+* that can be considered a verbal particle 'to'.

[4]Hebrew has an interrogative particle proclitic +ה *h+* 'is it true that ...?' that is now archaic. The subordinating conjunction proclitic +כש *kš+* 'as, when' can also attach to most words. Some prepositions can violate the order described above when they appear before the relativizer +ש *š+*, e.g., +ש+מ *m+š+* 'from that'. We do not handle these cases in our regular expression methods.

In this paper, we focus on the question of morphological segmentation of clitics in Hebrew words to make them easier to translate into English. We do not investigate deeper models of morphology that target lemmatization or inflectional features such as gender, number, and tense (El Kholy and Habash, 2012b).

3.2 Hebrew Out-of-Vocabulary Errors

The Out-of-Vocabulary (OOV) rate in our baseline development set is rather high: 7.0% of all tokens and almost 18% of all types. This is primarily due to the limited size of the parallel text we have access to (Tsvetkov and Wintner, 2010). The resource limitation is a good reason to consider morphological preprocessing given insights from previous published work (Lee, 2004; Habash and Sadat, 2006). We analyzed 10% of all the OOVs, a total of 80 cases from 40 sentences. Verbs are the most frequent part-of-speech (43%) followed by nouns (31%), adjectives (21%) and proper nouns (5%). The definite article +ה *h*+ appears in one-quarter of all cases, and the conjunction +ו *w*+ 'and' in one-fifth. Various prepositional clitics appear a total of 20% and the relativizer +ש *š*+ occurs in one-tenth of all cases. Only one case of a pronominal enclitic was in the sample studied (1.25%). About two-fifths of all cases do not involve any attached clitics (39%), almost one-half have one clitic (47%) and less than one-seventh have two (14%). About 60% of these cases can be potentially addressed by clitic tokenization.

4 Hebrew Preprocessing Techniques

We consider three preprocessing techniques: regular-expressions, unsupervised morphology learning (Creutz and Lagus, 2007), and morphological analysis and disambiguation (Adler, 2009).

4.1 Regular Expression Segmentation

In the first technique, we use simple regular expressions that deterministically segment the Hebrew word. We define four levels of segmentation schemes which we call S1, S2, S3, and S4. S1 splits off the conjunction +ו *w*+ 'and' and the relativizer +ש *š*+ 'that/who'. S2 includes S1, and additionally splits off the preposition clitics +ב *b*+ 'in/on', +כ *k*+ 'like/as', +ל *l*+ 'to/for', and +מ *m*+ 'from'. S3 includes S2, and additionally splits off +ה *h*+ 'the'. Finally, S4 includes S3, and additionally splits off pronominal enclitics (unless the definite article is present). The relative order of the these components, which is discussed in Section 3, is strictly preserved. The clitics' order and form are the only linguistic information utilized in this technique. These segmentation schemes are comparable to the tokenization schemes used by Habash and Sadat (2006) for Arabic: S1≈D1, S2≈D2, and S4≈D3. S3 is in between D2 and D3. To distinguish between schemes and techniques, we use REGEX-*scheme* to designate the regular expression techniques, e.g., REGEX-S1 is the regular expression technique targeting the S1 scheme.

The regular expressions directly apply these rules using no word-context information. As a result, this technique is very fast and is likely to make a lot of errors. Since the phrase-based SMT approach is robust to such segmentation errors (to a limit), we still expect this technique to help over the baseline.

4.2 Morfessor: Unsupervised Morphology

In the second technique, we use Morfessor (MORF), a state-of-the-art tool for unsupervised segmentation of words into morphemes (Creutz and Lagus, 2007). It is language independent, i.e., uses no linguistic knowledge. Instead, it creates a lexicon of morphs, such that the lexicon is both concise and can be used to build any word in the input. The conciseness is measured by combining a cost of the text based on its probability when represented by the morphemes in the lexicon with a cost based on the size of the lexicon. MORF then searches the space of segmentations to minimize that cost. It can be used in one of two modes, either learning a model directly from the input it is segmenting, or learning a model from one training set, and applying that segmentation model to an independent input set. In our experiment, we trained MORF on the word list of the combined training and tuning data sets, then applied that model to each data set, training, tuning, development, and test, in the second mode. We did not use additional monolingual data for training MORF in this paper although this is an interesting idea to study in the future. MORF is fairly quick, but slower than regular expressions. Similar to regular expressions, MORF does not use word-context, i.e., the segmentation is deterministic once a model is built. Furthermore, the produced segmentation is not guaranteed to be a well-defined tokenization scheme or

Base	Gloss	REGEX-S1	REGEX-S2	REGEX-S3	REGEX-S4	MORF	HTAG
להבדיל ✓ *lhbdyl*	to distinguish	להבדיל ✓ *lhbdyl*	ל+הבדיל *l+hbdyl*	ל+ה+בדיל *l+h+bdyl*	ל+ה+בדיל *l+h+bdyl*	להבדיל ✓ *lhbdyl*	להבדיל ✓ *lhbdyl*
שליט ✓ *šlyT*	ruler	ש+ליט *š+lyT*	ש+ל+יט *š+l+yT*	ש+ל+יט *š+l+yT*	ש+ל+יט *š+l+yT*	שליט ✓ *šlyT*	שליט ✓ *šlyT*
השלום *hšlwm*	the peace	השלום *hšlwm*	השלום *hšlwm*	ה+שלום ✓ *h+šlwm*	ה+שלום ✓ *h+šlwm*	ה+שלום ✓ *h+šlwm*	ה+שלום ✓ *h+šlwm*
להלאים ✓ *lhlaym*	to nationalize	להלאים ✓ *lhlaym*	ל+הלאים *l+hlaym*	ל+ה+לאים *l+h+laym*	ל+ה+לאים *l+h+laym*	ל+ה+לא+ים *l+h+la+ym*	להלאים ✓ *lhlaym*
לאור *lawr*	in light of	לאור *lawr*	ל+אור ✓ *l+awr*	ל+אור ✓ *l+awr*	ל+אור ✓ *l+awr*	לאור *lawr*	ל+אור ✓ *l+awr*

Table 1: Word Segmentation Examples. Linguistically valid segmentations that are consistent with the gloss are marked with ✓.

	Token Increase	**Similarity to Baseline**	**Accuracy**	
			Gold-S4	**Gold (*Scheme*)**
REGEX-S1	113%	87.4%	70.1%	99.7% (S1)
REGEX-S2	141%	62.2%	65.3%	79.1% (S2)
REGEX-S3	163%	46.3%	68.2%	70.6% (S3)
REGEX-S4	190%	33.8%	54.5%	
MORF	124%	81.6%	72.9%	
HTAG	130%	71.8%	94.0%	
Gold-S4	136%	68.4%		

Table 2: Tokenization system statistics.

to be linguistically correct. These are clearly important limitations given what we know about Hebrew morphology.

4.3 Hebrew Morphological Analysis and Disambiguation

In the third technique, we use a Hebrew morphological tagger (HTAG) (Adler, 2009). The tagger uses a morphological analysis component (or dictionary) together with a disambiguation component trained in an unsupervised manner. The tokenization produced by this tool resembles the S4 scheme discussed above but is context sensitive. This technique is the most linguistically rich of the three techniques used. This results in the most accurate segmentation of words into true morphemes; however, it is the slowest of all the methods. We do not experiment with variations of the schemes based on the tagger's choices as Habash and Sadat (2006) did for Arabic.

4.4 Comparing the Techniques

Table 1 presents some examples of the output of different techniques from our development set. Linguistically correct (at least with regards to the chosen glosses) are indicated.

Table 2 presents three comparison angles contrasting the different techniques presented above. All statistics are computed over a 50-sentence sample consisting of 600 hand-annotated (gold reference) words from the development set. The gold annotations are in a linguistically correct S4 scheme (the maximally verbose scheme). The first column, labeled *Token Increase*, shows the ratio of the number of tokens in a particular scheme to the corresponding number in the baseline system (no tokenization). As expected, the ratio increases as the number of segmentation decisions increases, with REGEX-S4 having the highest ratio. MORF and HTAG have similar numbers and are in between REGEX-S1 and REGEX-S2. The general trends in the full development set are consistent with the studied sample except that the ratios are around 4% lower on average.

The second column presents similarity to the no-tokenization baseline, or in other words, the percentage of unchanged words in the input. As expected REGEX-S1 and REGEX-S4 are the

least and most aggressive techniques, respectively. MORF is not as aggressive as HTAG.

The last two columns list the accuracy of the tokenization techniques against the gold annotation in S4 scheme as well as against a matching scheme converted from the human annotation to match the appropriate less verbose schemes (S1, S2 and S3). REGEX-S1 is highly accurate (99.7%) in its limited decisions. But HTAG has the best accuracy on the most verbose scheme (S4). The worst accuracy is for REGEX-S4. It is hard to judge MORF since it is not necessarily intended to match an S4 scheme, but we provide the number for comparison reasons. In close inspection, MORF seems to make odd decisions: in ≈82% of the time, no tokenization is made, but in the other 18% very wild and excessive decisions take place.

5 Evaluation

5.1 Experimental Settings

We test a total of six systems (REGEX-S1, REGEX-S2, REGEX-S3, REGEX-S4, MORF, HTAG), as well as a no-tokenization baseline. For all of the systems, our data is a Hebrew-English sentence-aligned corpus produced by Tsvetkov and Wintner (2010). We split the data into training, tuning, development, and test sets. The training and tuning data sets are used for training and tuning the translation models. Experiments were initially run on the development data set, and finally run on the test data set when all settings and schemes were finalized. Table 3 presents the data subset details.

In the baseline, the Hebrew data is tokenized just to split punctuation. English data is white-space/punctuation tokenized and lowercased. The English MT output is true-cased using the recaser tool that is part of the Moses toolkit (Koehn et al., 2007). The recaser is trained on the English side of the training and tuning sets. For the baseline and all of the experiments, the preprocessing is applied to all data sets - training, tuning, development, and test. After preprocessing, but before training, we filter down to sentences of 100 tokens or less in length. As a result, with more tokenization, there are fewer eligible sentences. The difference is minor, however. We train the translation models and decode with the Moses toolkit (Koehn et al., 2007). We used two English language models, held constant across all experiments: a trigram language model from the English side of the training data and a large 5-gram language model that preexisted this effort from English Gigaword (Graff and Cieri, 2003). Feature weights are tuned to maximize BLEU (Papineni et al., 2002) using Minimum Error Rate Training (Och, 2003) for each system separately.

5.2 Results and Discussion

The results are summarized in Table 4. Results are presented in terms of BLEU (Papineni et al., 2002), NIST (Doddington, 2002) and METEOR (Banerjee and Lavie, 2005).[5]

There is a general trend of improvement in BLEU score going down the table. Each subsequent experiment does better than the last in both the development and test data sets, with the exceptions of REGEX-S3 and REGEX-S4 as compared to REGEX-S2. This is a similar trend to Arabic (Habash and Sadat, 2006). Morphological analysis has a clear impact on translation quality, with both MORF and HTAG scoring higher than the regular expression systems. HTAG also is consistently the best performer in terms of all studied metrics. All differences in BLEU and NIST scores between all systems and the baseline and between MORF and HTAG are statistically significant above the 95% level. The differences between MORF and HTAG and each of REGEX-S1 through REGEX-S4 are also significant. Statistical significance is computed using paired bootstrap resampling with 1000 samples (Koehn, 2004).

The METEOR results support HTAG being the best system; however, the METEOR difference between HTAG and MORF is much bigger than in BLEU; and MORF is not consistently ranked second best.

It's notable that although MORF has no Hebrew-specific linguistic knowledge behind it, it is competitive with the REGEX techniques. This seems to show that linguistic information may not be sufficient to make a non-sophisticated technique perform well, and that unsupervised segmentation can go quite far.

Loosely, OOV levels drop as scores improve, but there are a few exceptions. REGEX-S4 has a lower OOV level than the other regular expression experiments, but its performance varies. A particularly notable exception is HTAG compared to MORF, where MORF has a significantly lower

[5] We used METEOR v1.2 with HTER task mode (Denkowski and Lavie, 2010).

Data Set	Sentences	Tokens	Types	Token OOV	Type OOV
Training	64,155	853,827	83,606		
Tuning	500	7,299	3,762	683 (9.4%)	677 (18.0%)
Development	1,000	11,405	4,386	798 (7.0%)	786 (17.9%)
Test	1,000	14,354	6,249	1311 (9.1%)	1288 (20.6%)

Table 3: Data set statistics.

	Development				Test			
	BLEU %	**NIST**	**METEOR**	**OOV**	**BLEU %**	**NIST**	**METEOR**	**OOV**
Base	20.96	5.3015	42.99	798	19.31	5.4951	44.36	1311
REGEX-S1	21.54	5.3805	44.61	587	20.39	5.6468	45.46	985
REGEX-S2	22.21	5.4491	43.26	401	21.69	5.8082	46.50	671
REGEX-S3	22.38	5.5365	44.33	318	21.61	5.8761	46.60	567
REGEX-S4	21.24	5.4021	42.22	273	21.07	5.8067	46.03	461
MORF	23.06	5.5590	43.16	**28**	22.25	5.9751	46.53	**48**
HTAG	**23.09**	**5.6317**	**44.87**	349	**22.79**	**6.1033**	**48.20**	556
COMBO1	22.69	5.5612	43.47	44	22.72	6.0381	47.20	74
COMBO2	22.68	5.5458	43.78	159	22.69	6.0275	47.17	250

Table 4: Results on development and test sets in multiple MT evaluation metrics. OOVs are presented in absolute (not percentage) counts.

Hebrew	החמאס ייהנה מהשפע הזה ויחזק את מעמדו.
Reference	Hamas will benefit from this bonanza.
Base	Hamas ייהנה this מהשפע ויחזק his status.
S1	Hamas ייהנה מהשפע and will the status.
S2	Hamas will benefit from abundance this will his status.
S3	Hamas will benefit from abundance and adds the status.
S4	Hamas will this affect this abundance standing and adds.
MORF	Hamas will be here what plenty and he adds the status.
HTAG	Hamas will benefit from abundance and will his status.
Hebrew	יש לנו קומקום ופלאטה בחדר.
Reference	We have an electric kettle and a hotplate in our room.
Base	We have brought ופלאטה in the room.
S1	We have קומקום and פלאטה in the room.
S2	We have קומקום and פלאטה in the room.
S3	We've got קומקום and פלאטה in the room.
S4	We have kettle and ופלאט room.
MORF	We've got a complete wonder anywhere.
HTAG	We've got kettle and פלאטה in the room.

Table 5: Translation examples.

OOV level, but also lower scores. By looking at the data, it is very clear that MORF's aggressive segmentation is behind the low OOV level, while it seems that HTAG always does the correct level of segmentation. Because of this, MORF's lower OOV level does not necessarily seem to contribute to better MT quality.

The example translations in Table 5 demonstrate some of these points. In the first example, OOV words are a major problem for the baseline system. By REGEX-S2, OOV is no longer a problem. Systems that segment more begin to produce more extraneous words. Finally, HTAG, instead of oversegmenting, produces the same output as REGEX-S2. In the second example, much more segmentation is required to deal with the OOV words. Once again, HTAG closely matches the REGEX-based system with the best output, and manages to successfully translate one of the OOV words. On the other hand, MORF shows its overaggressive segmentation, as it eliminates OOV words, but comes up with completely unrelated words instead.

Preliminary Combination Experiments In a preliminary combination experiment, we considered two simple ideas to combine the power of HTAG with other systems. First, for every sentence in the output of HTAG, if the sentence has an OOV, and MORF does not, we replace the HTAG output with the MORF output (COMBO1 in Table 4). Note that if a MORF sentence has even one OOV word, the corresponding HTAG sentence would not be replaced, even if it had several OOV words. Second, we retranslate the HTAG sentence after we replace each HTAG OOV with a tokenization from one of the other systems that makes the OOV invocabulary in HTAG (COMBO2 in Table 4). This is done with a preference for the most conservative REGEX-S1 down to the least conservative REGEX-S4 and then backing off to MORF. A replacement would not happen if either no method had a tokenization, or the tokenization didn't produce tokens in the phrase table for HTAG. This second scenario was especially likely for MORF tokenizations. The results are not promising, scoring lower than HTAG. These experiments suggest that the OOVs that are unhandled are very hard to address without additional data or more intensive language-specific OOV handling approaches (Habash, 2008). More sophisticated approaches to MT combination can be explored in the future (Rosti et al., 2007).

6 Conclusions and Future Work

We explored a range of preprocessing solutions for Hebrew-English SMT. Our best system, using a morphological analyzer and tagger, increases 3.5 BLEU points over a no-tokenization baseline on a blind test set. The next best result we got (as measured by BLEU) uses Morfessor, an unsupervised morphological segmenter. In the future, we plan to explore combinations of the different tokenization schemes, both pre- and post-translation, perhaps using lattices (Dyer et al., 2008). We also plan to consider Hebrew-specific OOV solutions similar to work by Habash (2008) on Arabic.

Acknowledgments

The work presented here was supported in part by a Google research award. We would like to thank Or Biran, Alon Lavie, Yuval Marton, and Shuly Wintner for helpful feedback and discussions.

References

Adler, Meni. 2009. *Hebrew Morphological Disambiguation: An Unsupervised Stochastic Word-based Approach*. Ph.D. thesis, Ben Gurion University.

Al-Haj, Hassan and Alon Lavie. 2012. The impact of Arabic morphological segmentation on broad-coverage English-to-Arabic statistical machine translation. *Machine Translation*, 26:3–24.

Banerjee, Satanjeev and Alon Lavie. 2005. 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.

Creutz, Mathias and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Trans. Speech Lang. Process.*, 4:3:1–3:34.

Denkowski, Michael and Alon Lavie. 2010. Extending the meteor machine translation evaluation metric to the phrase level. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 250–253, Los Angeles, California.

Doddington, George. 2002. Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurrence Statistics. In *Human Language Technology*, pages 128–132, San Diego.

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

El Kholy, Ahmed and Nizar Habash. 2012a. Orthographic and morphological processing for English–Arabic statistical machine translation. *Machine Translation*, 26:25–45.

El Kholy, Ahmed and Nizar Habash. 2012b. Translate, Predict or Generate: Modeling Rich Morphology in Statistical Machine Translation. In *Proceedings of EAMT 2012*, Trento, Italy.

Fishel, Mark and Harri Kirik. 2010. Linguistically motivated unsupervised segmentation for machine translation. In *Proceedings of the seventh International Conference on Language Resources and Evaluation (LREC)*, Valletta, Malta.

Goldwater, Sharon and David McClosky. 2005. Improving Statistical MT Through Morphological Analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 676–683, Vancouver, Canada.

Graff, David and Christopher Cieri. 2003. English Gigaword, LDC Catalog No.: LDC2003T05. Linguistic Data Consortium, University of Pennsylvania.

Habash, Nizar and Fatiha Sadat. 2006. Arabic preprocessing schemes for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, NAACL-Short '06, pages 49–52, Stroudsburg, PA.

Habash, Nizar. 2008. Four Techniques for Online Handling of Out-of-Vocabulary Words in Arabic-English Statistical Machine Translation. In *Proceedings of ACL-08: HLT, Short Papers*, pages 57–60, Columbus, Ohio.

Habash, Nizar. 2010. *Introduction to Arabic Natural Language Processing*. Morgan & Claypool Publishers.

Itai, Alon and Shuly Wintner. 2008. Language resources for Hebrew. *Language Resources and Evaluation*, 42:75–98.

Koehn, P. and K. Knight. 2003. Empirical methods for compound splitting. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 1*, pages 187–193.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Christopher Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Christopher 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.

Koehn, Philipp. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of the Empirical Methods in Natural Language Processing Conference (EMNLP'04)*, Barcelona, Spain.

Lavie, A., S. Wintner, Y. Eytani, E. Peterson, and K. Probst. 2004. Rapid prototyping of a transfer-based Hebrew-to-English machine translation system. In *10th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI)*.

Lee, Y.S. 2004. Morphological analysis for statistical machine translation. In *Proceedings of HLT-NAACL 2004: Short Papers*, pages 57–60.

Lembersky, Gennadi, Noam Ordan, and Shuly Wintner. 2011. Language Models for Machine Translation: Original vs. Translated Texts. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 363–374, Edinburgh, Scotland, UK.

Nießen, Sonja and Hermann Ney. 2004. Statistical Machine Translation with Scarce Resources using Morpho-syntactic Information. *Computational Linguistics*, 30(2).

Och, Franz Josef. 2003. Minimum Error Rate Training for Statistical Machine Translation. In *Proceedings of the 41st Annual Conference of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan.

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

Popović, Maja and Hermann Ney. 2004. 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.

Rosti, Antti-Veikko, Necip Fazil Ayan, Bing Xiang, Spyros Matsoukas, Richard Schwartz, and Bonnie Dorr. 2007. Combining outputs from multiple machine translation systems. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 228–235, Rochester, New York.

Shilon, Reshef, Nizar Habash, Alon Lavie, and Shuly Wintner. 2012. Machine translation between Hebrew and Arabic. *Machine Translation*, 26:177–195.

Snyder, Benjamin and Regina Barzilay. 2008. Unsupervised Multilingual Learning for Morphological Segmentation. In *Proceedings of ACL-08: HLT*, pages 737–745, Columbus, Ohio.

Tsvetkov, Y. and S. Wintner. 2010. Automatic acquisition of parallel corpora from websites with dynamic content. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, pages 3389–3392.

Virpioja, Sami, Jaakko J. Vyrynen, Mathias Creutz, and Markus Sadeniemi. 2007. Morphology-aware statistical machine translation based on morphs induced in an unsupervised manner. In *Proceedings of Machine Translation Summit*, Copenhagen, Denmark.