Overcoming Vocabulary Sparsity in MT Using Lattices

Steve DeNeefe and **Ulf Hermjakob** and **Kevin Knight**
USC Information Sciences Institute
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90292 USA
{sdeneefe,ulf,knight}@isi.edu

Abstract

Source languages with complex word-formation rules present a challenge for statistical machine translation (SMT). In this paper, we take on three facets of this challenge: (1) common stems are fragmented into many different forms in training data, (2) rare and unknown words are frequent in test data, and (3) spelling variation creates additional sparseness problems. We present a novel, lightweight technique for dealing with this fragmentation, based on bilingual data, and we also present a combination of linguistic and statistical techniques for dealing with rare and unknown words. Taking these techniques together, we demonstrate +1.3 and +1.6 BLEU increases on top of strong baselines for Arabic-English machine translation.

1 Introduction

Despite the existence of very large bilingual data sets, statistical machine translation faces a sparseness problem—words, phrases, and lexicalized syntactic patterns do not occur frequently enough for statistics to nail down their behaviors in translation. This problem is most severe for languages with complex word-formation rules. Here we attack three facets of this challenge for complex source languages:

- First, common stems are fragmented into many different forms in training data. Statistics collected for these words are therefore not as robust as they should be. This paper presents a novel, lightweight method for addressing this fragmentation. This method uses the target side of the bilingual corpus to help decide when to break up source-side words. At decoding time, without the target side available, the method suggests multiple ways to break up source words. We demonstrate this technique splitting off the `w-` prefix[1] from Arabic source words, where we obtain a +0.8 BLEU increase from correct handling of this single morpheme.
- Second, rare and unknown words appear frequently in test data. We develop a combination of linguistic and statistical techniques for processing such words at decoding time. To drive this method, we create a set of linguistic rules for dealing with common affixes; these rules are made available to the decoder search.
- Third, spelling variation exacerbates the sparseness we already see in the data. We introduce and evaluate methods for normalizing orthographic variations and correcting misspelled words.

All of these methods' decisions are uncertain ones—their suggestions are not always correct, and blindly following them would introduce many errors into the translations. Therefore, we represent our test sentences as lattices; all the methods give their advice by adding lattice arcs to the source-language string. The new lattice paths represent alternate source-language analyses that the decoder can use. We can then add features to our model to guide the choice of paths. Thus, the lattice represents a kind

[1] Throughout this paper, we use the Buckwalter transliteration of Arabic letters for easier recognition by English readers.

of common structure onto which the new knowledge sources can write their suggestions. When all of the above methods are integrated via lattices, we obtain +1.3 and +1.6 BLEU score improvements on top of strong Arabic-English baselines.

2 Related Work

It has been demonstrated several times, and for several different language pairs, that considering the morphology of a language can improve the quality of statistical MT. For European languages such as Spanish, Catalan, Serbian, German, and Czech, using morphological knowledge to deterministically modify data leads to gains (Nießen and Ney, 2004; Popović and Ney, 2004; Goldwater and McClosky, 2005). Dyer (2007) improves a Czech-English MT system by training multiple models on original and simplified versions of the data, combines them, then represents the same variations in the decoder input using a confusion network.

For Arabic, Lee (2004) demonstrates a gain in SMT quality for smaller training corpora by using automatically aligned parallel corpora to determine the best way to tokenize Arabic to match the parallel English, relying on an English POS tagger and a morphological stemmer. However, the gains did not carry over to larger corpora. Habash and Sadat (2006) compared the use of the BAMA (Buckwalter, 2002) and MADA (Habash and Rambow, 2005) toolkits as well as simple pattern matching to do morphological analysis for Arabic-English SMT, and were able to improve translation for tasks with small or out-of-domain training corpora. The BAMA toolkit provides many analyses based on hand-designed linguistic rules, while the MADA toolkit builds upon that foundation using statistics to determine the proper analysis. Sadat and Habash (2006) also showed that it was possible to combine the use of several variations of morphological analysis both while decoding (combining multiple phrase tables) and rescoring the combined outputs of distinct systems. Recently, Habash (2008) explored techniques for handling unknown source words in Arabic-English SMT including spelling correction and morphological variation by enriching the phrase table, rather than using lattices as we do in this work.

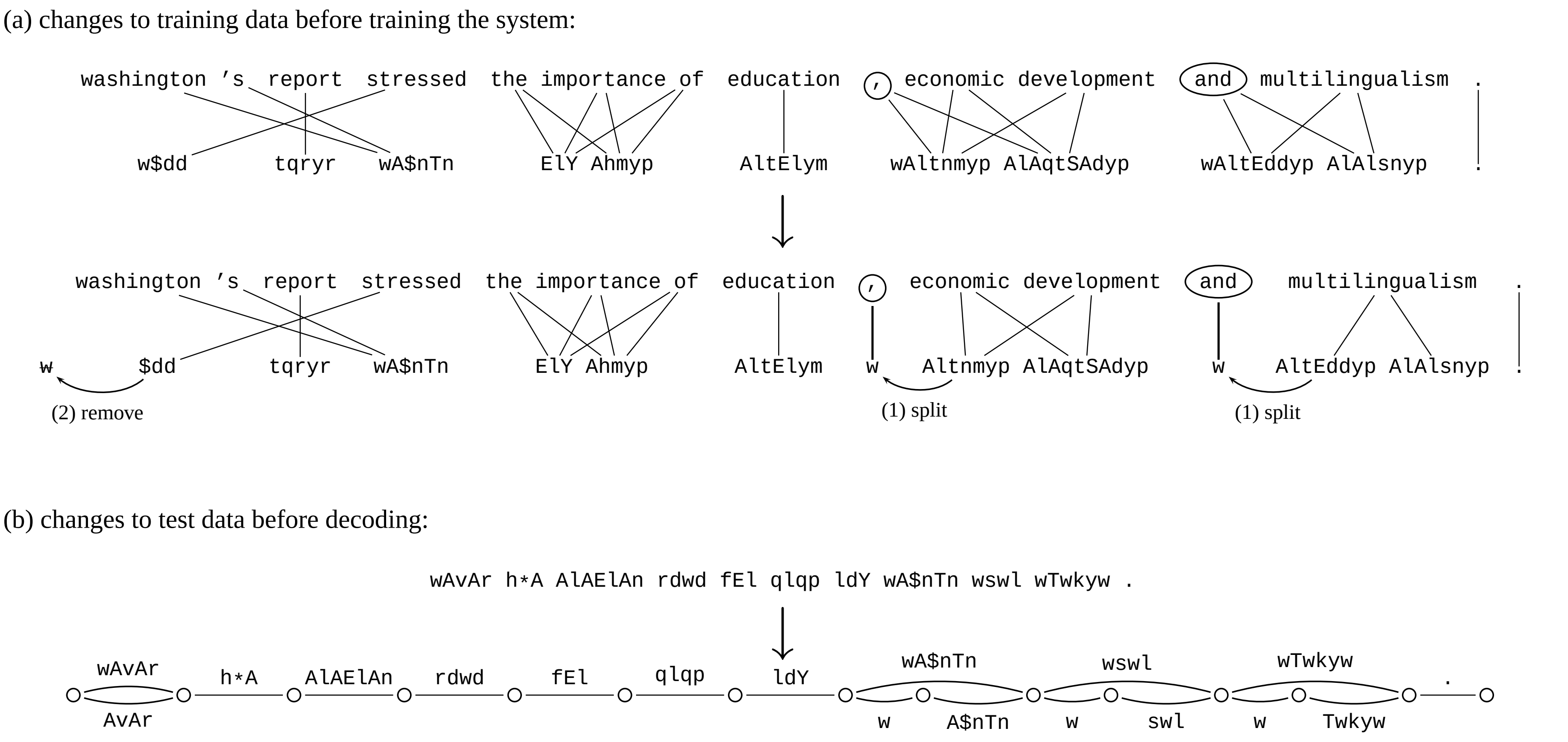
Lattices have been used for NLP tasks for some time, especially in the speech community. Decoding a lattice containing the output from an ASR system, rather than the single best analysis of spoken word, is a widely-used and proven technique (Ney, 1999; Saleem et al., 2004; Matusov et al., 2005, etc.). Wu (1996) allowed for multiple Chinese segmentations using a technique that is equivalent to a fully connected lattice. Recently, Dyer et al. (2008) present lattices as a useful generalization for text-based MT, applying them to source language alternatives such as Chinese segmentation variations and Arabic morphological variations. The Arabic morphological analysis used the BAMA toolkit to segment the source text, and a unigram LM to disambiguate between alternatives.

The focus of this paper is handling common sources of vocabulary sparseness in Arabic-English MT, not morphology per se. Many of the above works use morphological toolkits, while in this work we explore lightweight techniques that use the parallel data as the main source of information. We are able to combine both linguistic and statistical sources knowledge and then train the system to select which information it will use at decoding time. And unlike much previous work, we are able to show an improvement for large training data conditions.

3 Using Alignments to Aid Morphological Analysis of Common Words

The rich morphology of Arabic can often interfere with the collection of statistics over training data in a statistical MT system. One English word or phrase will coincide with multiple variations of the same Arabic root word with different affixes, thus fragmenting the phrase table and co-occurrence statistics. In some cases, an affix is equivalent to an English function word and can be split off into its own word and separately aligned. In other cases, the affix is superfluous for the purposes of English translation and can be removed. In this section we describe a lightweight technique for statistical morphological analysis of common words and affixes.

Figure 1 shows the overall technique: first modify the training data using the aligned English as a guide before training a system, then represent the same possible modifications non-deterministically in the test data using lattices.


(a) changes to training data before training the system:

(b) changes to test data before decoding:

washington ’s report stressed the importance of education , economic development and multilingualism .
w$dd tqryr wA$nTn ElY Ahmyp AltElym wAltnmyp AlAqtSAdyp wAltEddyp AlAlsnyp .
washington ’s report stressed the importance of education , economic development and multilingualism .
~~w~~ $dd tqryr wA$nTn ElY Ahmyp AltElym w Altnmyp AlAqtSAdyp w AltEddyp AlAlsnyp .
(2) remove
(1) split
(1) split
wAvAr h*A AlAElAn rdwd fEl qlqp ldY wA$nTn wswl wTwkyw .
wAvAr
AvAr
h*A
AlAElAn
rdwd
fEl
qlqp
ldY
wA$nTn
w
A$nTn
wswl
w
swl
wTwkyw
w
Twkyw
.

Figure 1: (a) Starting with the training data set as shown at the top, we modify the training data as follows: (1) split off w- prefix when motivated by the aligned English words (shown circled), and (2) remove sentence-initial w- prefix based on corpus statistics. The resulting modified training data (as shown below the arrow) is used to train the MT system. (b) We transform the input data into a lattice containing all possible variants of morphological processing for the w- prefix. Shown here, the top arcs contain the original input words, while the bottom arcs represent the modifications.

and
as well as
, (comma)
; (semicolon)

Table 1: English words, phrases, or punctuation that motivate splitting the Arabic w- prefix

	tokens	types
total # of Arabic words in training	37,544,015	392,328
# of words starting with w	2,264,896	76,110
# of w- prefixes split	857,149	49,481
# of w- prefixes removed	584,453	11,459
total # of words modified	1,441,602	51,631

Table 2: Effect of w handling on training data: words starting with w are very common in the data (6% of all tokens), and nearly two-thirds of those are split or removed by our procedure

3.1 Modifying the Training Data

When working with parallel training data, decisions about Arabic morphological analysis for MT should be informed by the English side of the data. In Figure 1(a) there are four Arabic words that start with the Arabic letter w. Consider the last ones first: the third and fourth w-words—wAltnmyp and wAltEddyp—both begin phrasal alignments, where the aligned English phrase starts with a comma or the word "and". Splitting off the w into a separate word and aligning it solely with the motivating English word leads to a finer-grained alignment and a reduction in sparsity. See Table 1 for a list of motivating English words used to find w splits.

Next consider the second word wA$nTn. It is aligned only to its English transliteration "washington" and the possessive token "'s", so there is no English word to motivate a split.

The first word is also an important case, because Arabic sentences often start with the letter w. This w is often used at the beginning of an Arabic sentence, but only by convention and not conveying any information[2]. It is usually more appropriate to ignore it in an English translation, rather than translate it as "and". Therefore, we want to remove it as shown in the figure. However, as seen in the case of "washington" above, some Arabic words start with a non-prefix w, and these can also occur at the beginning of a sentence. To determine if the w at the beginning of a sentence-initial word should be removed, we create a list of words likely to be prefixed by w. Over the entire training data, we count the English words aligned to our sentence-initial Arabic word, and separately count the English words aligned to the form of the word without the starting w. If we detect a non-trivial overlap between the aligned English words, then we consider this w to be a grammatical prefix and remove it, otherwise we leave it in place. We consider an English word to be a trivial overlap if it is a stop word (e.g., to, of, or, the), or if the number of occurrences is below a certain threshold. For the most common words on the list, some manual analysis was required to differentiate between trivial and non-trivial overlap, for example, when the Arabic words themselves were stop words.

Table 2 describes the effect of both of these methods on training data. Note that nearly 4% of the foreign word tokens were modified.

3.2 Modifying the Test Data

Now that the foreign text and alignments of the training data have been modified to better match the English side of the training data, we also need to modify the test data similarly. However, there are two important differences. First, since our previous morphological analysis of the foreign text depends on the English side, we cannot repeat the same process with unseen test data. Also, we want our test data to have flexibility in case of errors in our training data modifications. To be robust, we provide all possible alternatives to the decoding process in the form of a lattice: both the original input words as well as the split or clipped variants. All relevant input words are processed in this way, not just rare or unknown words.

Figure 1(b) shows the lattice for a typical test sentence. Notice that for the first word variations with and without the w- prefix are given, and for every other word starting with w we provide both one word and two word paths. We do this even when we would not split the training data, such as for wA$nTn, the transliteration of "washington".

We give positive empirical results for this method

[2] Similar to "So…" in spoken English.

in Section 6.

4 Handling morphological variation of rare words

Morphology not only fragments our training data, but it also causes many rare and unknown words to appear in test data. The two major sources of such morphological fragmentation are (1) prefixes and suffixes and (2) general inflection, e.g., different forms of adjectives based on number, gender, and definiteness. We automatically generate more frequent variations of unknown or rare words by searching our training data for related forms.

4.1 Affixes

Consider that we have detected a particular word as rare[3] or unknown to our translation model and in addition we recognize possible Arabic affixes to split off. For example, if the word `llbw*A` was unknown, we recognize that it could be an `l-` prefixed version of `Albw*A` or a `l-` + `Al-` double-prefixed version of `bw*A`. We then look up the stemmed versions in our training data to see if they are more frequently occurring than our original word. If so, we provide alternate lattice paths for each stemmed version of the word along with its affix. For example `llbw*A` becomes three paths: first it is split into two words `l-` and `Albw*A`. Then `Albw*A` is further be broken up into `Al-` and `bw*A`.

Because prefixes such as `l-` are not words found in the training data, the system needs additional help to translate them. To this end, we manually create a set of translation rules for each affix. Since these rules were generated artificially, we also provide a feature to govern their use in our translation model. Figure 2 shows the rules provided for the `l-` prefix. We provide a total of 193 rules covering 13 different affixes, all similar in spirit to those shown.

Our system does not always preserve the Arabic order of the components of a token. Consider for example `mqAmhm` (“their place”), which ends in the common pronominal suffix `-hm`. This suffix can serve as both as an object pronoun (“them”) and a possessive pronoun (“their”). In such cases, we add to the lattice both `mqAm` + `-hm` as well as `hm-` + `mqAm`. We create object pronoun rules for the `-hm` form, and possessive pronoun rules for the `hm-` form. The English-based word order makes creating these rules and the subsequent decoding much simpler.

4.2 General inflection

An adjective such as `wAlqrsywn` (“and coercive”, plural, masc.) might be out of vocabulary, even after the prefixes `w-` and `Al-` are split off. However, the inflected forms `Alqrsyp` (“coercive”, sing., fem.) and `AlqrsY` (“coercive”, sing., masc.) might occur thousands of times in the training corpus. Our system therefore generates such alternative adjective forms for the lattice. The decoder can then directly use these alternative adjective forms, relying on the fact that the English translations for these adjective forms are the same, because English does not inflect adjectives based on number, gender, or definiteness. The same applies to dual/plural nouns and verbal participles, which also share the same morphological forms in English.

Our system handles tokens that are formed by a combination of prefixes, a suffix, and some general inflection.

5 Correcting Typos

A misspelled word often results in a token never or very rarely seen in a training bitext, rendering it untranslatable by a basic SMT system.

We found that the most common kinds of typos are missing or spurious spaces, missing or spurious letters, transposed letters, replacement of similar-looking letters, and attachment of junk characters.

To generate likely spelling corrections for words in a test sentence, we consider each word without any context. If that word occurs at most once in the training corpus, we generate spelling-corrected candidates by applying the reverse of each of these typo operations, keeping only results that occur more often in the training corpus than the original word. As in previous sections, we add the resulting spelling corrections to the test sentence lattice, rather than replacing the original word entirely.

Table 3 shows the kinds of spelling correction our system performs along with some examples. The counts indicate how frequently a token (or sequence

[3] occurring in the training corpus up to 10 times

translations of l- in context	**alternate syntactic structure**	**rules that delete l-**
PP(IN(for) x_0:NP-C) ↔ l- x_0	PP(IN(for) NP-C(x_0:NPB)) ↔ l- x_0	NP-C(x_0:NP-C) ↔ l- x_0
PP(TO(to) x_0:NP-C) ↔ l- x_0	PP(TO(to) NP-C(x_0:NPB)) ↔ l- x_0	NP-C(x_0:NPB) ↔ l- x_0
PP(IN(by) x_0:NP-C) ↔ l- x_0	PP(IN(by) NP-C(x_0:NPB)) ↔ l- x_0	
PP(IN(towards) x_0:NP-C) ↔ l- x_0	PP(IN(towards) NP-C(x_0:NPB)) ↔ l- x_0	

Figure 2: Syntax-based rules to translate the l- prefix in context, derived from an Arabic-to-English dictionary. Similar rules are created for the morphemes Al-, w-, f-, s-, b-, k-, -h, -hA, -hm, -hmA, -nA, and -km. These rules follow the style of Galley et al. (2004) for syntax-based translation rules.

Operation	Original	Count	Spelling-corrected	Count	English
Add missing space	Alm&AmrpAlHqyqyp	0	Alm&Amrp AlHqyqyp	638; 6,034	real conspiracy
- ASCII/non-ASCII	ywmY14	0	ywmY 14	8,085; 84,881	daily 14
Drop spurious space	n wfmbr	18,346; 5	nwfmbr	73,147	November
Add missing letter	Al$AEAt	1	AlA$AEAt	310	rumors
Drop spurious letter	ms&wwl	1	ms&wl	24,794	official
Replace similar letter	mHAdtAt (محادتات)	0	mHAdvAt (محادثات)	28,899	talks
Swap transposed letters	wsylqtY	1	wsyltqY	656	and will meet
Remove junk characters	/ElY	0	ElY	2,445,781	on

Table 3: Types of typos handled by spelling correction, with examples from the training corpus before and after correction. Differences are underlined (except for spacing). Count indicates how frequently a token occurs in the training data.

of tokens) occurs in the training corpus.

Strictly speaking, we do not distinguish between actual typos and rare but correct words. This operation does not actually detect typos, but rather finds very rare or unseen tokens and produces more common alternatives. In practice, however, this often produces the spelling corrections desired.

Note that we do not try to fix multiple typos in a single word. However, if a specific misspelling occurs often enough in the training data, the general SMT framework can produce proper alignments, rules, and translations to English for such a common misspelling without a special-purpose typo-correction module. This allows our SMT system to correct words that contain both a common and a rare typo.

Consider, for example, the Arabic token mHmwEbAs ("Mahmouabbas" instead of "Mahmoud Abbas"), which lacks both the Arabic letter d and the space. The token does not occur in the training corpus, but mHmwdEbAs, which includes the missing d (but still lacks the missing space) happens to occur 4 times in the training corpus. Our spelling correction adds the missing d, which then in turn enables the decoder to correctly translate the partially spell-corrected Arabic token to "Mahmoud Abbas".

We do not add spelling-correction alternatives for tokens that are already covered by other rare word handling techniques such as those that translate quantities and rare proper names.

It is worth noting a few technical details:

- The junk characters we consider for removal include control characters, punctuation, letters from alphabets other than Arabic and extended Latin, as well as the Arabic *tatweel* character.
- Adding missing characters could lead to a potentially large number of candidates, as the missing character could be any character anywhere in the word. We therefore optimize the process of adding missing characters by creating a reverse index mapping misspelled tokens to correctly spelled tokens—those occurring at least 10 times in the training corpus.

6 Experiments and Evaluation

We evaluate these techniques, both separately and jointly, using the statistical syntax-based MT system described by Galley et al. (2006) and DeNeefe et al. (2007). Syntax-based rules translate a string into an English parse tree via a CKY decoder. This decoder is extended to handle input lattices using the basic technique of van Noord (1995).

Dataset	# of sentences	# of Arabic words	# of English words
training	2,033,696	37,544,015	44,225,727
newswire development set	1,385	37,681	n/a
web development set	2,516	51,776	n/a
newswire test set	1,500	40,287	n/a
web test set	2,530	50,365	n/a

Table 4: Description of datasets used in end-to-end MT experiments

Experiment	Development BLEU		Test BLEU	
	newswire	web data	newswire	web data
baseline	54.6	21.5	51.9	19.2
common word morphology	55.1		52.7	
rare word morphology	54.6		52.3	
typo correction	54.4		51.9	
all combined	**55.5**	**23.0**	**53.2**	**20.8**

Table 5: Individually, the common and rare word morphology handling techniques achieve gains of +0.8 and +0.4 on the newswire test set, while typo correction had no significant effect on BLEU. When combined, all three techniques bring a gain of +1.3 to the newswire test set and +1.6 BLEU to the web data test set.

We used the standard feature functions in decoding, and in addition we add one feature to our rare morphological stem rules and one feature to our spelling-correction lattice arcs. The feature weights are tuned separately for each experiment using minimum error rate training (Och, 2003).

Table 4 describes the datasets used for this evaluation—note that they are larger than those used in many other morphology experiments cited in the related work. Our data was aligned using the LEAF alignment method (Fraser and Marcu, 2007).

We measured the individual contribution of each technique separately, as well as the effect of combining all techniques. Table 6 shows our empirical results in terms of case-insensitive BLEU. Note that the morphology-related techniques provided a gain on their own, but the limited changes from spelling correction did not. On the blind test set, the total improvement for newswire was +1.3 BLEU, while for web text it was +1.6 BLEU. Both these improvements are statistically significant according to paired bootstrap resampling at the 99% confidence level.

Table 6 shows the empirical results in terms of the modifications to the development set, and how many of these modifications were used during decoding.

7 Conclusions and Future Work

In this paper, we addressed challenges raised by source languages with complex word-formation. We developed several methods and integrated them at decoding time via source-language lattices, obtaining good improvements in end-to-end translation.

There are possible extensions to this work. The `w-` prefix is not the only common affix in Arabic that often corresponds to a separate word in English. We believe many of the affixes we use in the rare word handling of Section 4 (e.g., `b-`, `l-`, `Al-`, `k-`) could also be handled during training using similar techniques to those described in Section 3. Also, the spelling correction could be improved by looking at more context around the rare or unknown word, for example, using a bigram or trigram model. In addition, our analysis of spelling errors was done mostly on newswire data. More analysis could be done in other genres.

Acknowledgments

We thank Jason Riesa for early discussion and suggestions on this topic, Michael Pust and Jens-Sönke Vöckler for developing parts of the experimental framework, and other colleagues at ISI for their helpful input. We also thank the anonymous review-

	# occurring	**# used in decoding**
total # of Arabic words in newswire development set	37,681	n/a
# of words starting with w	3,036	n/a
# of w- prefixes split	2,244	1,591
# of w- prefixes removed	803	769
# of rare affixes split	735	251
# of spelling errors detected	104	n/a
# of spelling corrections proposed	163	66

Table 6: Quantitative evaluation of modifications to newswire development data and their use at decoding time: again, words starting with w are very frequent (8% of all tokens). For rare affixes and spelling corrections, often several mutually exclusive options are proposed, but only one can be chosen during decoding time. Note that spelling correction had the smallest effect (correcting only 66 words).

ers for useful comments. This research is financially supported under DARPA Contract No. HR0011-06-C-0022, BBN subcontract 9500008412.

References

Tim Buckwalter. 2002. Buckwalter Arabic morphological analyzer version 1.0. Linguistic Data Consortium, University of Pennsylvania, 2002. LDC Catalog No.: LDC2002L49.

Steve DeNeefe, Kevin Knight, Wei Wang, and Daniel Marcu. 2007. What can syntax-based MT learn from phrase-based MT? In *Proc. EMNLP-CoNLL 2007*.

Christopher Dyer, Smaranda Muresan, and Philip Resnik. 2008. Generalizing word lattice translation. In *Proc. ACL 2008*.

Christopher Dyer. 2007. The “noisier channel”: Translation from morphologically complex languages. In *Proc. WMT 2007*.

Alexander Fraser and Daniel Marcu. 2007. Getting the structure right for word alignment: LEAF. In *Proc. EMNLP-CoNLL 2007*.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What’s in a translation rule? In *Proc. HLT-NAACL 2004*.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proc. ACL 2006*.

Sharon Goldwater and David McClosky. 2005. Improving statistical MT through morphological analysis. In *Proc. HLT-EMNLP 2005*.

Nizar Habash and Owen Rambow. 2005. Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In *Proc. ACL 2005*.

Nizar Habash and Fatiha Sadat. 2006. Arabic preprocessing schemes for statistical machine translation. In *Proc. NAACL 2006, Companion Volume: Short Papers*.

Nizar Habash. 2008. Four techniques for online handling of out-of-vocabulary words in Arabic-English statistical machine translation. In *Proc. ACL 2008: HLT, Short Papers*.

Young-Suk Lee. 2004. Morphological analysis for statistical machine translation. In *Proc. HLT-NAACL 2004: Short Papers*.

Evgeny Matusov, Stephan Kanthak, and Hermann Ney. 2005. On the integration of speech recognition and statistical machine translation. In *Proc. InterSpeech 2005*.

Hermann Ney. 1999. Speech translation: Coupling of recognition and translation. In *Proc. IEEE ICASSP 1999*.

Sonja Nießen and Hermann Ney. 2004. Statistical machine translation with scarce resources using morpho-syntactic information. *Computational Linguistics*, 30(2).

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. ACL 2003*.

Maja Popović and Hermann Ney. 2004. Towards the use of word stems and suffixes for statistical machine translation. In *Proc. LREC 2004*.

Fatiha Sadat and Nizar Habash. 2006. Combination of Arabic preprocessing schemes for statistical machine translation. In *Proc. ACL 2006*.

S. Saleem, S.-C. Joi, S. Vogel, and T. Schulz. 2004. Using word lattice information for a tighter coupling in speech translation systems. In *Proc. ICSLP 2004*.

Gertjan van Noord. 1995. The intersection of finite state automata and definite clause grammars. In *Proc. ACL 1995*.

Dekai Wu. 1996. A polynomial-time algorithm for statistical machine translation. In *Proc. ACL 1996*.