Combination of Statistical Word Alignments Based on Multiple Preprocessing Schemes

Jakob Elming
Center for Comp. Modeling of Language
Copenhagen Business School
je.id@cbs.dk

Nizar Habash
Center for Comp. Learning Systems
Columbia University
habash@cs.columbia.edu

Abstract

We present an approach to using multiple preprocessing schemes to improve statistical word alignments. We show a relative reduction of alignment error rate of about 38%.

1 Introduction

Word alignments over parallel corpora have become an essential supporting technology to a variety of natural language processing (NLP) applications most prominent among which is statistical machine translation (SMT).[1] Although phrase-based approaches to SMT tend to be robust to word-alignment errors (Lopez and Resnik, 2006), improving word-alignment is still useful for other NLP research that is more sensitive to alignment quality, e.g., projection of information across parallel corpora (Yarowsky et al., 2001).

In this paper, we present a novel approach to using and combining multiple preprocessing (tokenization) schemes to improve word alignment. The intuition here is similar to the combination of different preprocessing schemes for a morphologically rich language as part of SMT (Sadat and Habash, 2006) except that the focus is on improving the alignment quality. The language pair we work with is Arabic-English.

In the following two sections, we present related work and Arabic preprocessing schemes. Section 4 and 5 present our approach to alignment preprocessing and combination, respectively. Results are presented in Section 6.

2 Related Work

Recently, several successful attempts have been made at using supervised machine learning for word alignment (Liu et al., 2005; Taskar et al., 2005; Ittycheriah and Roukos, 2005; Fraser and Marcu, 2006). In contrast to generative models, this framework is easier to extend with new features. With the exception of Fraser and Marcu (2006), these previous publications do not entirely discard the generative models in that they integrate IBM model predictions as features. We extend on this approach by including alignment information based on multiple preprocessing schemes in the alignment process.

In other related work, Tillmann et al. (1997) use several preprocessing strategies on both source and target language to make them more alike with regards to sentence length and word order. Lee (2004) only changes the word segmentation of the morphologically complex language (Arabic) to induce morphological and syntactic symmetry between the parallel sentences. We differ from these two in that we do not decide on a certain scheme to make source and target sentences more symmetrical. Instead, it is left to the alignment algorithm to decide under which circumstances alignment information based on a specific scheme is more likely to be correct than information based on other schemes.

3 Arabic Preprocessing Schemes

Arabic is a morphologically complex language with a large set of morphological features. As such, the set of possible preprocessing schemes is rather large (Habash and Sadat, 2006). We focus here on a subset of schemes pertaining to Arabic attachable clitics. There are three degrees of cliticization that apply to a word BASE: `([CONJ+ [PART+ [Al+ BASE +PRON]]])`. At the deepest level, the BASE can have a definite article ال+ (Al+ *the*)[2] or a member of the

[1] The second author was supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-06-C-0023. Any opinions, findings and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of DARPA. We thank Necip Ayan, Mona Diab, Bonnie Dorr, Abe Ittycheriah, Martin Jansche and Owen Rambow for helpful discussions.

[2] Arabic is transliterated in Buckwalter's transliteration scheme.

Proceedings of NAACL HLT 2007, Companion Volume, pages 25–28,
Rochester, NY, April 2007. ©2007 Association for Computational Linguistics

Table 1: Arabic preprocessing scheme variants for وسيكتبها 'and he will write it'

Preprocessing Scheme		Example	
AR	simple	وسيكتبها	wsyktbhA
$D1$	split CONJ	و+ سيكتبها	w+ syktbhA
$D2$	split CONJ, PART	و+ سـ+ يكتبها	w+ s+ yktbhA
TB	Arabic Treebank	و+ سيكتب +ها	w+ syktb +hA
$D3$	split all clitics	و+ سـ+ يكتب +ها	w+ s+ yktb +hA

class of pronominal clitics, +PRON, (e.g., ها+ +hA *her/it/its*). Next comes the class of particles (PART+), (e.g., +سـ s+ *will [future]*). Most shallow is the class of conjunctions (CONJ+), (e.g., +و w+ *and*). We use the following five schemes: AR, $D1$, $D2$, $D3$ and TB. Definitions and contrastive examples of these schemes are presented in Table 1. To create these schemes, we use MADA, an off-the-shelf resource for Arabic morphological disambiguation (Habash and Rambow, 2005), and TOKAN, a general Arabic tokenizer (Habash and Sadat, 2006).

4 Preprocessing Schemes for Alignment

Using a preprocessing scheme for word alignment breaks the process of applying Giza++ (Och and Ney, 2003) on some parallel text into three steps: preprocessing, alignment and remapping. In preprocessing, the words are tokenized into smaller units. Then, they are passed along to Giza++ for alignment (default settings). Finally, the Giza++ alignments are mapped back (remapped) to the original word form which is AR tokens in this work. For instance, take the first word in Table 1, *wsyktbhA*; if the $D3$ preprocesssing scheme is applied to it before alignment, it is turned into four tokens (*w+ s+ yktb +hA*). Giza++ will link these tokens to different words on the English side. In the remapping step, the union of these links is assigned to the original word *wsyktbhA*. We refer to such alignments as remappings.

5 Alignment Combination

After creating the multiple remappings, we pass them as features into an alignment combiner. The combiner is also given a variety of additional features, which we discuss later in this section. The combiner is simply a binary classifier that determines for each source-target pair whether they are linked or not. Given the large size of the data used, we use a simplifying heuristic that allows us to minimize the number of source-target pairs used in training. Only links evidenced by at least one of the initial alignments and their immediate neighbors are included. All other links are considered non-existent. The combiner we use here is implemented using a rule-based classifier, Ripper (Cohen, 1996). The reasons we use Ripper as opposed other machine learning approaches are: (a) Ripper produces human readable rules that allow better understanding of the kind of decisions being made; and (b) Ripper is relatively fast compared to other machine learning approaches we examined given the very large nature of the training data we use. The combiner is trained using supervised data (human annotated alignments), which we discuss in Section 6.1.

In the rest of this section we describe the different machine learning features given to the combiner. We break the combination features in two types: word/sentence level and remapping features.

Word/Sentence Features:
• **Word Form**: The source and target word forms.
• **POS**: The source and target part-of-speech tags.
• **Location**: The source and target *relative* sentence position (the ratio of absolute position to sentence length). We also use the difference between these values for both source and target.
• **Frequency**: The source and target word frequency computed as the number of occurrences of the word form in training data. We also use the ratio of source to target frequency.
Similarity: This feature is motivated by the fact that proper nouns in different languages often resemble each other, e.g. صدام حسين 'SdAm Hsyn' and 'saddam hussein'. We use the equivalence classes proposed by Freeman et al. (2006) to normalize Arabic and English word forms. Then, we employ the longest common substring as a similarity measure.

Remapping Features:
• **Link**: for each source-target link, we include (a) a binary value indicating whether the link exists according to each remapping; (b) a cumulative sum of the different remappings supporting this link; and (c) co-occurrence information for this link. This last value is calculated for each source-target word pair as a weighted average of the product of the relative frequency of co-occurrence in both directions for each remapping. The weight assigned to each

remapping is computed empirically.[3]

- **Neighbor**: The same information as Link, but for each of the immediate neighbors of the current link.
- **Cross**: These include (a) the number of source words linked to the current target word, the same for target to source, and the number of words linked to either of the current words; and (b) the ratio of the co-occurrence mass placed in this link to the total mass assigned to the source word, the same for the target word and the union of both.

6 Evaluation

6.1 Experimental Data and Metrics

The gold standard alignments we use here are part of the IBM Arabic-English aligned corpus (IBMAC)[4] (Ittycheriah and Roukos, 2005). We only use 8.8K sentences from IBMAC because the rest (smaller portion) of the corpus uses different normalizations for numerals that make the two sets incompatible. We break this data into 6.6K sentences for training and 2.2K sentences for development. As for test data, we use the IBMAC's test set: NIST MTEval 2003 (663 Arabic sentences each human aligned to four English references).

To get initial Giza++ alignments, we use a larger parallel corpus together with the annotated set. The Arabic-English parallel corpus has about 5 million words.[5] The Arabic text in IBMAC is preprocessed in the AR preprocessing scheme with some additional character normalizations. We match the preprocessing and normalizations on our additional data to that of IBMAC's Arabic and English preprocessing (Ittycheriah and Roukos, 2005).

The standard evaluation metric within word alignment is the Alignment Error Rate (AER) (Och and Ney, 2000), which requires gold alignments that are marked as 'sure' or 'probable'. Since the IBMAC gold alignments we use are not marked as such, AER reduces to 1 - F-score (Ittycheriah and Roukos, 2005):

$$Pr = \frac{|A \cap S|}{|A|} \quad Rc = \frac{|A \cap S|}{|S|} \quad AER = 1 - \frac{2PrRc}{Pr+Rc}$$

where A links are proposed and S links are gold. NULL links are not included in the evaluation (Ayan, 2005; Ittycheriah and Roukos, 2005).

6.2 Results

We conducted three experiments on our development data: (a) to assess the contribution of alignment remapping, (b) to assess the contribution of combination features for a single alignment (i.e., independent of the combination task) and (c) to determine the best performing combination of alignment remappings. Experiments (b) and (c) used only 2.2K of the gold alignment training data to minimize computation time. As for our test data experiment, we use our best system with all of the available data. We also present an error analysis of our best system. The baseline we measure against in all of these experiments is the state-of-the-art grow-diag-final (*gdf*) alignment refinement heuristic commonly used in phrase-based SMT (Koehn et al., 2003). This heuristic adds links to the intersection of two asymmetrical statistical alignments in an attempt to assign every word a link. The AER of this baseline is 24.77%.

The Contribution of Alignment Remapping We experimented with five alignment remappings in two directions: *dir* (Ar-En) and *inv* (En-Ar). We also constructed their corresponding *gdf* alignment. The more verbose a preprocessing scheme, the lower the AER for either direction and for *gdf* of the corresponding remapping. The order of the schemes from worst to best is AR, $D1$, $D2$, TB and $D3$. The best result we obtained through remapping is that of $D3_{gdf}$ which had a 20.45% AER (17.4% relative decrease from the baseline).

The Contribution of Combination Features For each of the basic ten (non *gdf*) alignment remappings, we trained a version of the combiner that uses all the relevant features but has access to one alignment at a time. We saw a substantial improvement for all alignment remappings averaging 29.9% relative decrease in AER against the basic remapped version. The range of AER values is from 14.5% ($D3_{dir}$) to 20.79% (AR_{inv}).

Alignment Combination Experiments To determine the best subset of alignment remappings to combine, we ordered the alignments given their AER performance in the last experiment described (using combination features). Starting with the best performer ($D3_{dir}$), we continued adding alignments in the order of their performance so long the com-

[3] We use the AER on the development data normalized so all weights sum to one. See Section 6.2.

[4] We thank IBM for making their hand aligned data available to the research community.

[5] All of the training data we use is available from the Linguistic Data Consortium (LDC). The parallel text includes Arabic News, eTIRR, English translation of Arabic Treebank, and Ummah.

Table 2: Combining the Alignment Remappings

Alignment Remapping combination	**AER**
$D3_{dir}$	14.50
$D3_{dir}D2_{dir}$	14.12
$D3_{dir}D2_{dir}D3_{inv}$	12.81
$D3_{dir}D2_{dir}D3_{inv}D1_{dir}$	12.75
$D3_{dir}D2_{dir}D3_{inv}D1_{dir}AR_{inv}$	12.69

bination's AER score is decreased. Our best combination results are listed in Table 2. All additional alignments not listed in this table caused an increase in AER. The best alignment combination used alignments from four different schemes which confirms our intuition that such combination is useful.

Test Set Evaluation We ran our best system trained on all of the IBMAC data (training & development), on all the unseen IBMAC test set. On this data we achieve a substantial relative improvement of 38.3% from an AER of 22.99 to 14.19.

Ittycheriah and Roukos (2005) used only the top 50 sentences in IBMAC test data. Our best AER result on their test set is 14.02% (baseline is 22.48%) which is higher than their reported result (12.2% with 20.5% baseline (unrefined GIZA++)). The two results are not comparable because: (a) Ittycheriah and Roukos (2005) used additional gold aligned data that was not released and (b) they use an additional 500K sentences from the LDC UN corpus for Giza training that was created by adapting to the source side of the test set – the details of such adaptation were not provided and thus it is not clear how to replicate them to compare fairly. Clearly this additional data is helpful since even their baseline is higher than ours.[6]

Error Analysis We conducted error analysis on 50 sentences from our development set. The majority of the errors involved high frequency closed-class words (54%) and complex phrases (non-compositional or divergent translations) (23%). Both kinds of errors could be partly addressed by introducing phrasal constraints which are currently lacking in our system. Orthogonally, about 18% of all errors involved gold-standard inconsistencies and errors. These gold errors are split equally between closed-class and complex-phrase errors.

7 Conclusion and Future Plans

We have presented an approach for using and combining multiple alignments created using different preprocessing schemes. We have shown a relative reduction of AER of about 38% on a blind test set. In the future, we plan to extend our system with additional models at the phrase and multi-word levels for both alignment and alignment combination improvement. We plan to use more sophisticated machine learning models such as support vector machines for combination and make use of more available parallel data. We also plan to evaluate the influence of our alignment improvement on MT quality.

References

N. Ayan. 2005. *Combining Linguistic and Machine Learning Techniques for Word Alignment Improvement*. Ph.D. thesis, University of Maryland, College Park.

W. Cohen. 1996. Learning trees and rules with set-valued features. In *Fourteenth Conference of the American Association of Artificial Intelligence*. AAAI.

A. Fraser and D. Marcu. 2006. Semi-supervised training for statistical word alignment. In *ACL-06*.

A. Freeman, S. Condon, and C. Ackerman. 2006. Cross linguistic name matching in English and Arabic. In *HLT-NAACL-06*.

N. Habash and O. Rambow. 2005. Arabic Tokenization, Part-of-Speech Tagging and Morphological Disambiguation in One Fell Swoop. In *ACL-05*.

N. Habash and F. Sadat. 2006. Arabic Preprocessing Schemes for Statistical Machine Translation. In *HLT-NAACL-06*.

A. Ittycheriah and S. Roukos. 2005. A maximum entropy word aligner for arabic-english machine translation. In *EMNLP-05*.

P. Koehn, F. Och, and D. Marcu. 2003. Statistical Phrase-based Translation. In *HLT-NAACL-03*.

Y. Lee. 2004. Morphological Analysis for Statistical Machine Translation. In *HLT-NAACL-04*.

Y. Liu, Q. Liu, and S. Lin. 2005. Log-linear models for word alignment. In *ACL-05*.

A. Lopez and P. Resnik. 2006. Word-based alignment, phrase-based translation: what's the link? In *AMTA-06*.

F. Och and H. Ney. 2000. Improved statistical alignment models. In *ACL-2000*.

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

F. Sadat and N. Habash. 2006. Combination of Arabic Preprocessing Schemes for Statistical Machine Translation. In *ACL-06*.

B. Taskar, S. Lacoste-Julien, and D. Klein. 2005. A discriminative matching approach to word alignment. In *EMNLP-05*.

C. Tillmann, S. Vogel, H. Ney, and A. Zubiaga. 1997. A DP-based search using monotone alignments in statistical translation. In *ACL-97*.

D. Yarowsky, G. Ngai, and R. Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *HLT-01*.

[6] Abraham Ittycheriah, personal communication.