Boosting Performance of Weak MT Engines Automatically: Using MT Output to Align Segments & Build Statistical Post-Editors

Clare R. Voss[1], Matthew Aguirre[2], Jeffrey Micher[1],
Richard Chang[3], Jamal Laoudi[3], Reginald Hobbs[1]

[1]Multilingual Computing Branch, Army Research Laboratory, Adelphi, MD 20783
[2]ArtisTech, Inc., 10560 Main St., Suite 105, Fairfax, VA 22030
[3]Advanced Resources Technologies, Inc., 1555 King St., Suite 400 Alexandria, VA 22314
{voss,jmicher,rachang, jlaoudi, hobbs}@arl.army.mil, maguirre@artistech.com

Abstract. This paper addresses the practical challenge of improving existing, operational translation systems with relatively weak, black-box MT engines when higher quality MT engines are not available and only a limited quantity of online resources is available. Recent research results show impressive performance gains in translating between Indo-European languages when chaining mature, existing rule-based MT engines and post-MT editors built automatically with limited amounts of parallel data. We show that this hybrid approach of serially composing or "chaining" an MT engine and automated post-MT editor---when applied to much weaker lexicon-based and rule-based MT engines, translating across the more widely divergent languages of Urdu and English, and given limited amounts of document-parallel only training data---will yield statistically significant boosts in translation quality up to the 50K of parallel segments in training the post-editor, but not necessarily beyond that.

Introduction

In industry and government, MT developers may be asked to improve existing, operational translation systems with relatively weak, black-box MT engines because higher quality MT engines are not available and only a limited quantity of online resources is available. Recent research results show impressive performance gains in translating between Indo-European languages when chaining together mature, existing rule-based MT engines and post-MT editors built automatically with limited amounts of parallel data ([1], [2], [3]). In this paper, we show that this hybrid approach of serially composing an MT engine and automated post-MT editor---when applied to much weaker MT engines, translating across more widely divergent languages, and given only limited amounts of training data---will yield statistically significant boosts in translation quality up to the first 50K of parallel segments in training the post-editor, but not necessarily beyond that.

The key idea behind our approach is to have MT engines do their own translations to boost the performance of the systems in which they are embedded. We document and present results of this "self-help" workflow where (i) the MT engine outputs are used to identify segment-level alignments, (ii) the resulting segment pairs are used to train automated statistical post-editors (APEs), and (iii) the resulting APEs form part of serially chained systems (MT + APE) that outperform the original MT engines.

The paper begins with a brief overview of our approach to post-MT processing. The Alignment Section presents a novel algorithm that we developed for identifying Urdu-English segment-level alignments based on Urdu and English document-aligned files. In the Results and Analyses section that follows we review evaluation results and begin to address questions raised in the Approach section. The paper concludes with a discussion of open issues and notes of future work.

Approach

We know that human translators dislike working with MT output because---with no mechanisms built into the MT system to learn directly from human post-editing corrections--- the same errors appear over and over and the translators must make the same corrections over and over again as well. Our in-house requirement has been to determine how to boost the MT engines we already have and eliminate, where possible, known errors. In this paper, we report on leveraging the SRILM and MOSES tools ([4], [5]) without modifications to rapidly build statistical post-MT editors in just a few months. Our work follows from the insights of [2] and [3] that "post-editors" can be built as monolingual translation engines that convert "raw" target-language (TL) text produced by a baseline MT engine into higher quality TL text by correcting errors in TL word choice and order.

Our approach has been to augment two in-house Urdu-to-English MT engines, one rule-based and one lexicon-based, with automated statistical post-editors built from the same corpus of parallel-aligned data to address several questions:

1. How effective are automated post-editors (APEs) in word re-orderings to boost an Urdu-English MT lexicon-based MT (LBMT) where no re-ordering of the Urdu input occurs? How does this compare to an APE's impact on a rule-based MT (RBMT) where some re-ordering occurs prior to the APE processing by the baseline MT[1]?
2. How much impact does the amount of parallel data for building an APE have on the performance of a LBMT+APE hybrid versus a RBMT+APE hybrid?
3. How effective are the RBMT+APE and LBMT+APE hybrids compared to a standalone statistical MT engine (SMT) built with the same data as the APEs? Are the different engines impacted equally by segment-level alignments of different qualities?

[11] Given that Urdu-English translation has longer distance re-ordering than in prior work of French-English translation where re-orderings are mostly local, we expected that an APE for Urdu-to-English translation would be less effective than an APE for French-to-English translation.

Data Preparation and Alignment

The training data in our study was restricted to the NIST 2008 Open MT Workshop's Urdu "language pack" on DVD that included collections of document-parallel Urdu and English files that NIST provided without standard cleaning, with the stated expectation that workshop participants would modify the files as needed [6]. Our first challenge in using this data was to identify and extract pairs of Urdu and English segments that were translations of each other from separate Urdu and English files aligned only at the document level. The intuition for the aligner algorithm that we developed came from our observations reading the Urdu files after they were run through our two in-house MT engines: even the low-quality raw "English" output of these engines was "good enough" for us to scan and match by content with segments in the corresponding English files. We then wondered if automated evaluation metrics could do this matching for us, by identifying the highest scoring matched pairs of Urdu-translated "English" segments with English-original segments.

As a check on the possibility of segments aligning across document boundaries, we asked an Urdu speaker to examine several pairs of aligned Urdu and English documents to determine whether segments from one Urdu document appeared in the preceding or following documents. This concern arose in part from the fact that aligned documents did not always contain the same number of segments. We discovered that, even though segments did not always have a corresponding partner in their aligned document, the segments did not align across document boundaries. As a result, our alignment algorithm was restricted to comparing segments within aligned documents.

Before starting the alignment, documents were binned into three groups: those containing the same number of segments (Equal), those whose segment counts were off by one (OneOff), and those whose segments counts were more than one off (MoreThanOneOff). We had expected that segment pairs within Equal document set might already be perfectly aligned. On inspection however, we found that many documents in the Equal set were not segment-aligned. The automated evaluation metrics in the algorithm were BLEU [7] and GTM 1.4 [8]. After some initial experimentation, BLEU was set to have an n-gram size of 2, to yield more of a score spread across segments. The translation engines in the algorithm were the in-house LBMT and RBMT engines. The algorithm also included post-MT processing prior to segment-pair scoring to remove annotations intended for the human reader only, to boost segment scores and again create more spread across segments.

Alignment Algorithm

The algorithm steps, necessarily simplifying somewhat from all the details, were:

1. Split the original single files with all of the "aligned" data in it into separate source and reference files based on document ID.
2. Translate all of the Urdu segments to English using both MT engines.
3. For each engine's output and each metric, perform the fully exhaustive (N x M) evaluations of each of the N MTed segments against each of the M reference segments, on

each document. This results in four triplets of {metric score, source segment ID, reference segment ID}, because both MT engines' output were scored with both metrics.

4. All triplets for the two sets with the same metric were ranked, with the more likely aligned segment pairs above those judged less likely based on their metric score. In the event of equivalent scores, the tie was broken by selecting pairs whose difference in source and reference IDs were closest (to each other in the document).
5. An iterative algorithm for selection, deletion, and re-ranking of the triples for "most probable" alignment was then applied to the lists. The highest scoring segment alignment was popped off the list first and saved as a candidate alignment. Then any other segments in the list with the same source ID or the same reference ID as the designated candidate were also discarded. This removed all competing alignments for either segment of the selected candidate, rapidly reducing the number of triples to re-sort and iterate through. During the selection phase, if pairs of possible segments crossed over each other, we removed the "worst-offending" cross-over pair, defined as the pair with the most number of other segments crossed. With the pair removed, the data was re-sorted and checked again for other crossovers.
6. The final lists of candidate alignments for each of the two metrics were then intersected and only alignments found by iterations over both metrics were kept. (Note that we effectively ignored the differences between MT engines by creating two lists of triplets.)

Evaluating the Alignment Algorithm

For an initial evaluation, we ran the algorithm on five documents selected from the MoreThanOneOff set, on the assumption these were "noisier" than the other sets and would give us a lower bound on the algorithm's performance. With the assistance of our Urdu speaker, we produced a gold-standard alignment on this set. Precision and recall metrics[2] were calculated on the algorithm-aligned (hypothesized) segments. The per-document pair results indicated that the algorithm would serve our needs for automatically extracting alignment candidates for training a post-editor: precision scores ranged from .67 to .9 and recall from .63 to .88, on documents that differed by 2 to 4 segments in length.

To evaluate the algorithm over the full collection, we built a sample of 13 documents from each of the binned sets (Equal, OneOff, MoreThanOne Off). We again created a gold standard alignment for each document pair and used it to score the sample alignments. Table 1 shows the number of segments in each file of the document pairs selected for the evaluation set. The evaluation results in the two bottom rows indicate that the algorithm performed effectively on the Equal and OneOff pairs, with precisions score at .98 and .93. The large drop in precision for the MoreThanOneOff pairs to .57 pre-empted our use of this data in our builds. Clearly the initial three-way binning of the documents by segment-count differences helped filter and isolate better alignments for MT training. In the next section, we describe the use of the alignments from the Equal and OneOff bins for the different system builds, in effect an extrinsic evaluation of these alignments.

[2] We define precision as # correct hypothesized alignments / total # hypothesized alignments, and recall as # correct hypothesized alignments / total # gold standard alignments.

Table 1. Evaluation of segment alignment algorithm on 13 document pairs from three dataset bins.

Document Bins	Equal #U=#E segs	OneOff {#U segs, #E segs}	MoreThanOneOff {#U segs, #E segs}
Listing of # segments in each document of pair to test for alignment	31, 12, 5, 4, 35, 14, 5, 6, 32, 4, 3, 7, 4	{14,15},{8,9},{13,12}, {30,31},{21,20},{15,16}, {6,5},{7,6},{6,5},{12,11}, {8,7},{10,9},{5,4}	{11,14},{43,13},{11,5}, {10,5},{7,9},{5,3},{8,3}, {15,10},{8,4},{11,8}, {3,5},{10,6},7,4}
Precision	**.98**	**.93**	**.57**
Recall	**.95**	**.86**	**.53**

Results and Analyses

To assess the impact of our two in-house MT engines in creating Urdu-English alignment and statistical post-editors to these engines, we also built, following the identical process, (i) a second set of post-editors using independently-created control alignments created by colleagues from the NIST DVD files with no knowledge of our algorithm[3] and (ii) two sets of standalone statistical MT engines, from our and our colleagues' alignments. All systems were evaluated on the same Urdu dataset of the NIST 2008 Open MT workshop, consisting of 1862 Urdu segments in 132 documents, where each segment was tagged and paired with four English human reference translations. Figure 1 shows the distribution of Urdu test segments by length. The test documents also varied in length from 3 to 79 segments, with slightly over half of the documents (68), having fewer than 10 segments.[4]

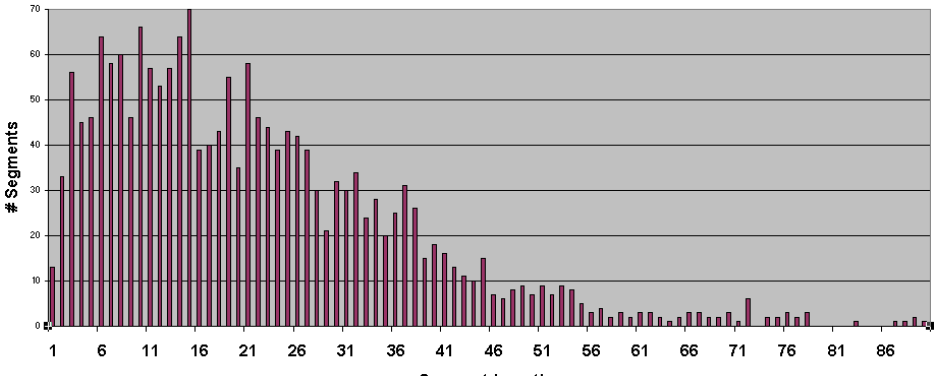

Figure 1. Histogram of #segments by segment length (in tokens) from evaluation dataset

[3] We thank Tim Anderson of AFRL and Wade Shen of MIT Lincoln Labs for sharing their datasets.

[4] Another forty documents, slightly under a third, had 10 to 19 segments. Twenty-one documents had 20 to 46 segments. Three others were much longer: 54, 76, 79 segments.

System-level Evaluation

The lexicon-based MT (LBMT) engine, when run standalone, produced a very weak BLEU-4 score (see baseline column in Table 2). This was not unexpected: given that this MT preserves Urdu's SOV and head-final phrase-internal word order because it does no reordering of translated words, its output scores points mostly for single word matches.

Table 2. System-level BLEU-4 scores on lexicon-based MT with automated post-editors trained on same 25, 50, 75, and 105K pairs of aligned segments as for RBMT APE & SMT in Tables 3 and 4.

	LBMT	LBMT+APEs (alignment set size)			
	baseline	25K	50K	75K	105K
alignment 1	0.064	0.150	0.172	0.180	0.185
alignment 2	0.064	0.161	0.182		

When the LBMT was chained with an automated post-editor (APE) built with only 25K of parallel segments, whether aligned by our colleagues (alignment 1) or our own (alignment 2), the hybrid score was more than double the baseline MT score. The hybrid score increases however fell off dramatically beyond that initial set: with a second 25K parallel segments to train the APE, the hybrid score increased by about one-eighth, and then with a third 25K, the hybrid score increased only by one-twentieth.

Table 3. System-level BLEU-4 scores on rule-based MT with automated post-editors trained on same sets of 25, 50, 75, 105K aligned segment pairs as for LBMT APE & SMT in Tables 2 and 4.

	RBMT	RBMT +APE			
	Baseline	+25K	+50K	+75K	+105K
alignment 1	0.127	0.180	0.195	0.202	0.206
alignment 2	0.127	0.185	0.203		

In contrast, the rule-based MT engine, when run standalone, scored significantly higher than the LBMT, at roughly double the BLEU points (see baseline column in Table 3). When chained with an APE built on 25K of parallel segments (alignment 2), the hybrid score increased roughly by one-half. While this was not as dramatic a gain as with the LBMT+APE combination, the increase was statistically significant nonetheless. The RBMT+APE score increases beyond that initial 25K dataset---as occurred with the LBMT+APE---fell off dramatically: with a second 25K parallel segments to train the APE, the score increasing by only about one-tenth, and then with a third 25K (in alignment 1), the score increased only by one-twentieth.

These results suggested the first 25K training datasets contained the critical mass of new in-genre, in-domain vocabulary and short phrases needed to translate the evaluation dataset, while the subsequent 25K datasets drawn from this same set of source texts contained much less new content and so only contributed to boosting the translation coverage in a more limited fashion.

To test for this possibility, we used only the two alignment sets to train a series of new statistical MT (SMT)[5] and the results came out consistent with this possibility. The BLEU scores on the SMT trained on our 25K and 50K alignments were statistically indistinguishable from the BLEU scores for the LBMT+APE engines trained on these same alignments (see Table 4). If, at the finer-grained document and segment levels, we were to see that SMT output does outscore LBMT+APE output some of the time, then it would be fair to ask the fundamental question for this hybrid approach: for another much larger, high quality alignment set, will the SMT systematically match or will it instead outscore the LBMT+APE, on the same amount of training data?[6]

Table 4. System-level BLEU-4 scores for statistical MTs trained on same alignment datasets of the APE engines in Tables 2 and 3 (The 14K* training set was built from Equal bin alignments only)

	SMT				
	14K*	25K	50K	75K	105K
alignment 1		0.144	0.163	0.175	0.180
alignment 2	0.144	0.166	0.186		

Document-level and Segment-level Evaluation

As a first step in addressing this question, Tables 5 and 6 show, at document- and segment-level evaluations respectively, how frequently the 50K APEs with our alignments boost their baseline MT engines. The RBMT+APE hybrid showed individual documents decreased in score from their baseline RBMT translation, but only 2 scores were statistically significantly lower. By contrast, at the segment level in Table 6, both hybrids show statistically significant drops in segment scores.

Table 5. Document-level Bleu-4 score changes from LBMT to LBMT+APE runs and from RBMT to RBMT+APE runs (all APEs built with 50K alignmt 2)

Document score changes Bleu-4	from LBMT to LBMT+APE	from RBMT to RBMT+APE
# increased / unchanged / decreased	132 / 0 / 0	124 / 0 / 8

It is especially intriguing that both hybrids show proportionately more decreases relative to their baseline MT systems in Bleu-1 scores at the segment-level (Table 6) than in BLEU-4 scores. This indicates that particular word or punctuation changes made by the APEs are "worse", i.e., with fewer 1-gram matches, even though on balance the APEs are increasing the higher-order n-gram matches that boost BLEU-4 scores, which could be a result of APE substitutions or re-orderings yielding longer matches. While the APE "advantages" with the first 50K training data are only partly a matter of increased vocabulary

[5] The English language model was built with only the English side of the parallel data.

[6] Since ramping up and maintaining a LBMT may be easier and less expensive than retraining an SMT or APE, the answer to this question has practical ramifications as well.

coverage that comes with more training data, the segment-level evaluation suggests that the lack of stronger lexical analysis in the APE to increase 1-gram matches is a limiting factor in boosting the overall performance of the hybrids.

Table 6. Segment-level Bleu score differences from LBMT to LBMT+APE runs and from RBMT to RBMT+APE runs (all APEs built with 50K alignmt 2)

Segment score changes	from LBMT to LBMT+APE	from RBMT to RBMT+APE
Bleu-4		
# increased / unchanged / decreased	1543 /66 / 252	1252 / 473 / 136
Bleu-1		
# increased / unchanged / decreased	1436 / 108 / 317	1198 / 197 / 466

In addition to looking at the added-value and limiting factors from the APEs themselves, we return to the question raised earlier about the impact of the baseline MTs on the system performance. Table 7 suggests that, using scores at the document level, there is consistent evidence in the score differences to rank order the LBMT + APE below SMT, with 71 out of 132 SMT documents outscoring the LBMT+APE. One explanation might be that an LBMT-specific APE faces more challenges with re-ordering edits to make on LBMT output than the SMT does on Urdu text: the APE must deal with noisy LBMT-induced English without the benefit of linguistic content and redundancy (such as morphological and syntactic information) from Urdu that has been lost. In contrast, the SMT is "free" to detect and make use of that Urdu linguistic knowledge for the re-ordering for translation into English.

Table 7. Document-level score differences between LBMT+APE and SMT engines, and between SMT and RBMT+APE engines (50K alignmt 2)

Document score changes Bleu-4	between LBMT+APE and SMT	between SMT and RBMT+APE
# increased / unchanged / decreased	71 / 59 / 2	90 / 3 / 39

Table 7 also shows that with document-level scores, there is some evidence to rank the SMT below the RBMT+APE builds on 50K alignments.[7] With a carefully constructed test set, it would be possible to determine whether this RBMT provides to its APE a parsing analysis and re-ordering advantage that the SMT we have built in its current form lacks (for example, no factored translation model [9], because we lacked Urdu resources to annotate our data for lemmas, part-of-speech, morphology, word class).

Though we have presented an evaluation of the alignments and hybrid builds in terms of Bleu scores, we recognize that this is but a first step in reaching a deeper understanding of the impact and effectiveness of APEs when chained with LBMT and RBMT engines,

[7] We apply paired t-tests of statistical significance over document scores, rather than using BLEU's automated confidence intervals without system to system paired comparisons.

especially given the well-recognized limitations of literal matching for assessing translation quality [10]. At this stage in our work, we have begun manually assessing segment outputs across translation engines by aligning them with multiple reference translations (RTs), as shown in Table 8. The APEs for both engines made three identical substitutions while also making other distinct changes. The LBMT+APE and StatMT outputs are strikingly similar, while the RBMT+APE output quite distinct re-ordering of the Urdu original word order (compare with LBMT output).

Table 8. Five MT system outputs (LBMT, LBMT+APE, RBMT, RBMT+APE, StatMT) on same input segment and four Reference Translations (RT1-4), manually aligned for presentation.

	Segment Translation				
LBMT	P S O	privatisation	,	injunction	issuing
LBMT+APE	pso	privatization	,	stay order	issued
RBMT	Injunction	ongoing	P S O	privatisation	,
RBMT+APE	stay order	the	pso	privatization	.
StatMT	pso	privatization	,	the stay order	issued
RT 1	PSO	Privatization	,	Stray Order	Issued
RT 2	PSO	Privatization	,	Stay Order	Issue
RT 3	Stay on	PSO	Privatization		
RT 4	Stay on	PSO	Privatization		

Conclusion and Future Work

In this paper, we have reported statistically significant performance improvements in (i) translating between Urdu and English, languages more divergent in word order than previously tested Indo-European pairs, by (ii) composing existing, but weak lexicon substitution-based and rule-based MT engines with statistical post-editors. The post-editors were trained on segment-level alignments generated with a novel, iterative re-ranking algorithm that selects most likely alignment pairs from automatically scored outputs of these two engines. We also examined document-level performance of the lexicon-based and rule-based hybrids for clues to limits we observed on their post-editors' improvements after 50K of training data.

The most striking result of using the MT engines' own outputs was the enormous gain in performance with the serial composition of the LBMT+APE system based on only 25K alignments. This suggests, for time-critical, rapid ramp-up of MT engines for very low-resource languages, that the first step is to find or build a translation lexicon and an

LBMT while immediately working in tandem to obtain document-parallel or comparable datasets that can boost the LBMT with progressively stronger APEs built with that engine. Longer term, however, given that (i) larger, in-domain training corpora can be constructedand (ii) SMTs outperform LBMT-based hybrids but underperform RBMT+APEs when trained on the same small quantities of data, we expect that RBMT-based hybrids, like our RBMT+APE or new automated RBMT hybrid types [11], will outperform SMTs on widely syntax-divergent language pairs[8]

References

1. Elming, J. "Transformation-based correction of rule-based MT." In Proceedings of the 11th Annual Conference of the European Association for Machine Translation, Oslo, Norway (2006)
2. Dugast, L., Senellart, J., Koehn, P. "Statistical Post-Editing on SYSTRAN's Rule-Based Translation System." In Proceedings of the Second ACL Workshop on Statistical Machine Translation, Prague, Czech Republic (2007)
3. Simard, M., Ueffing, N., Isabelle, P., Kuhn, R. "Rule-Ba.sed Translation with Statistical Phrase-Based Post-Editing," In Proceedings of the Second ACL Workshop on Statistical Machine Translation, Prague, Czech Republic (2007)
4. Stolke, A. "SRILM - an extensible language modeling toolkit." In Proceedings of the International Conference on Spoken Language Processing (2002)
5. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. "Moses: Open source toolkit for statistical machine translation." In Proceedings of the Annual Meeting of the ACL, Demonstration and poster session, Prague, Czech Republic (2007).
6. NIST 2008 Open MT Workshop with Urdu Resource DVD (R116_1_1), http://www.nist.gov/speech/tests/mt/2008/doc/MT08_EvalPlan.v2.4.pdf, www.nist.gov/speech/tests/mt/2008/doc/2008_NIST_MTOpenEval_Agmnt_StandardV3.pdf
7. Papineni, K., Roukos, S., Ward, T., Zhu, W. "BLEU: a method for automatic evaluation of MT." In Proceedings of the ACL, Philadelphia, PA (2002)
8. Melamed, I. Dan, Green, R., Turian, J. "Precision and recall of machine translation" In Proceedings of the HLT-NAACL, Edmonton Canada (2003)
9. Koehn, P., Hoang, H. "Factored Translation Models" EMNLP-CoNLL-2007: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Prague, Czech Republic (2007)
10. Callison-Burch, C., M. Osborne, P. Koehn "Re-evaluating the role of BLEU in machine translation research" In the Proceedings of EACL, Trento, Italy (2006)
11. Font Llitjós, A, Vogel, W. "A walk on the other side: adding statistical components to a transfer-based translation system" SSST, NAACL-HLT-2007 Workshop on Syntax and Structure in Statistical Translation, Rochester, NY (2007)

[8] [3] also discussed the relation of their RBMT+APE and SMT, projecting that their SMT would surpass the RBMT + APE only with a massive amount of training data.