Can Automatic Post-Editing Make MT More Meaningful?

Kristen Parton[1] **Nizar Habash**[1] **Kathleen McKeown**[1] **Gonzalo Iglesias**[2] **Adrià de Gispert**[2]
[1]Columbia University, NY, USA
{kristen, kathy, habash}@cs.columbia.edu
[2]University of Cambridge, Cambridge, UK
{gi212, ad465}@eng.cam.ac.uk

Abstract

Automatic post-editors (APEs) enable the re-use of black box machine translation (MT) systems for a variety of tasks where different aspects of translation are important. In this paper, we describe APEs that target adequacy errors, a critical problem for tasks such as cross-lingual question-answering, and compare different approaches for post-editing: a rule-based system and a feedback approach that uses a computer in the loop to suggest improvements to the MT system. We test the APEs on two different MT systems and across two different genres. Human evaluation shows that the APEs significantly improve adequacy, regardless of approach, MT system or genre: 30-56% of the post-edited sentences have improved adequacy compared to the original MT.

1 Introduction

Automatic post-editors (APEs) seek to perform the same task as human post-editors: correcting errors in text produced by machine translation (MT) systems. APEs have been used to target a variety of different types of MT errors, from determiner selection (Knight and Chander, 1994) to grammatical agreement (Mareček et al., 2011). There are two main reasons that APEs can improve over decoder output: they can exploit information unavailable to the decoder, and they can carry out deeper text analysis that is too expensive to do in a decoder.

We describe APEs that target three types of adequacy errors: deleted content words, content words that were translated into function words, and mistranslated named entities. These types of errors are common across statistical MT (SMT) systems and can significantly degrade translation *adequacy*, the amount of information preserved during translation. Adequacy is critical to the success of many cross-lingual applications, particularly cross-lingual question answering (CLQA), where adequacy errors can significantly decrease task performance. The APEs utilize word alignments, source- and target-language part-of-speech (POS) tags, and named entities to detect phrase-level errors, and draw on several external resources to find a list of corrections for each error.

Once the APEs have a list of errors with possible corrections, we experiment with different approaches to apply the corrections: an approach that uses phrase-level editing rules, and two techniques for passing the corrections as feedback back to the MT systems. The *rule-based APE* uses word alignments to decide where to insert the top-ranked correction for each error into the target sentence. This approach rewrites the word or phrase where the error was detected, but does not modify the rest of the sentence. We test these MT system-independent rules on two MT systems, MT A and MT B (described in more detail in section **??**).

The *feedback APE* passes multiple suggestions for each correction back to the MT system, and allows the MT decoder to determine whether to correct each error and how to correct each error during re-translation. Many MT systems have a mechanism for "pre-editing," or providing certain translations in advance (e.g., for named entities and numbers). We exploit this mechanism to provide post-editor feedback to the MT systems during a second-pass translation. While post-editing via feedback is a general technique, the mechanism the decoder uses is dependent upon the implementation of each MT system: in our experiments, MT A accepts corpus-level feedback from the APE, while MT B can handle more targeted, phrase-level feedback from the APE.

Our evaluation using human judgments shows that the APEs always improve the overall translation adequacy: across all conditions, whether rule-based or feedback, MT A or MT B, newswire or web genre, adequacy improved in 30-56% of post-edited sentences, and the improved sentences significantly outnumbered sentences that got worse. We also collected judgments on fluency, which highlighted the relative advantages of each APE

approach. The rule-based approach affords more control for error correction, at the expense of fluency. The feedback approach improves adequacy only when it can maintain some level of fluency, which results in more fluent post-edits than the rule-based approach. Due to the fluency constraints, the feedback APEs do not modify as many sentences as the rule-based APE, and therefore improve fewer sentences. Our analysis suggests ways in which feedback may be improved in the future.

2 Motivation

As MT has increased in quality and speed, its usage has gone beyond open-ended translation towards a variety of applications: cross-lingual subjectivity analysis, cross-lingual textual entailment, cross-lingual question-answering, and many others. Open-ended MT systems are task-agnostic, so they seek to balance fluency and adequacy. Depending on the task, however, adequacy may take precedence over fluency (or vice versa). We propose using the framework of automatic post-editing (Knight and Chander, 1994) to detect and correct task-specific MT errors at translation time. (In this paper, we use the term "post-editing" to refer to automatic post-editing only.)

The advantage of post-editing is that the APE can adapt any MT output to the needs of each task without having to re-train or re-tune a specific MT system (Isabelle et al., 2007). Acquiring parallel text, training and maintaining an SMT system is time-consuming and resource-intensive, and therefore not feasible for everyone who wishes to use MT in an application. Ideally, an APE can adapt the output of a black-box MT system to the needs of a specific task in a light-weight and portable manner. Since APEs are not tied to a specific MT system, they also allow application developers flexibility in switching MT systems as better systems become available.

Our focus on adequacy in automatic post-editing is motivated by CLQA with result translation. In this task, even when the correct answer in the source language is retrieved, it may be perceived as irrelevant in the target language if not translated correctly. The MT errors that have the biggest impact on CLQA include missing or mistranslated named entities and missing content words (Parton and McKeown, 2010; Boschee et al., 2010).

Manual error analysis of MT has shown that missing content words produce adequacy errors across different language pairs and different types of SMT systems. Condon et al. (2010) found that 26% of their Arabic-English MT errors were verb, noun or pronoun deletions. Similarly, Vilar et al. (2006) found that 22% of Chinese-English MT errors were content deletion. Popović and Ney (2007) reported that 68% deleted tokens from their Spanish-English MT system were content words. We address these errors via automatic post-editing, with the ultimate goal of improving MT output for adequacy-oriented tasks.

3 Related Work

The goal of APE is to automatically correct translated sentences produced by MT. Adaptive APEs try to learn how to improve the translation output by adapting to the mistakes made by a specific MT system. In contrast, general APEs target specific types of errors, such as English determiner selection (Knight and Chander, 1994), certain types of grammar errors in English (Doyon et al., 2008) and Swedish (Stymne and Ahrenberg, 2010), and complex grammatical agreement in Czech (Mareček et al., 2011). The APEs in this paper are more similar to general APEs, since they target specific kinds of adequacy errors.

APEs may utilize information unavailable to the decoder to improve translation output. Previous task-based MT approaches have used task context to select verb translations in CLQA at query time (Ma and McKeown, 2009) and to identify and correct name translations in CLIR (Parton et al., 2008). The rule-based APE we describe extends those APEs to cover additional types of adequacy errors. The feedback APEs are most similar to (Suzuki, 2011), which uses confidence estimation to select poorly translated sentences and then passes them to an adaptive SMT post-editor. Other work in confidence estimation (Specia et al., 2011) aims to predict translation adequacy at runtime without using reference translations, which is similar to our error detection step.

Many APEs use sentence-level analysis tools to make improvements over decoder output. Since these tools rely on having a fully resolved translation hypothesis (and since they are expensive), they are infeasible to run during decoding. The DepFix post-editor (Mareček et al., 2011) parses translated sentences, and uses the bilingual parses to correct Czech morphology. While syntax-based MT systems use POS and parses, most systems do not use other types of annotations (e.g., information extraction, event detection or sentiment analysis). An alternative approach would be to incorporate these features directly into the MT system; the focus of this paper is on adapting translations to the task without changing the MT system.

4 Post-Editing Techniques

Our APEs carry out three steps: 1) detect errors, 2) suggest and rank corrections for the errors, and 3) apply the suggestions. All the APEs use identical algorithms for steps 1 and 2, and only differ in how they apply the suggestions. The algorithms are language-pair independent, though we carried out all of our experiments on Arabic-English MT.

4.1 Pre-Processing

The Arabic source text was analyzed and tokenized using MADA+TOKAN (Habash et al., 2009). Each MT system used a different tokenization scheme, so the source sentences were processed in two separate pipelines. Separate named entity recognizers (NER) were built for each pipeline using the Stanford NER toolkit (Finkel et al., 2005), by training on CoNLL and ACE data. Each translated English sentence was re-cased using Moses and then analyzed using the Stanford CoreNLP pipeline to get part-of-speech (POS) tags (Toutanova et al., 2003) and NER (Finkel et al., 2005).

4.2 Detecting Errors and Suggesting Corrections

The APEs address specific adequacy errors that we have found to be most detrimental for the CLQA task: content words that are not translated at all, content words that are translated to function words, and mistranslated named entities. In the error detection step, these types of errors are detected via an algorithm from prior work that uses bilingual POS tags and word alignments (Parton and McKeown, 2010). Each flagged error consists of one or more source-language tokens and zero or more target-language tokens. In the error correction step, the source and target sentences and all the flagged errors are passed to the suggestion generator, which uses the following three resources.

Phrase Table: The phrase table from MT B is used as a phrase dictionary (described in more detail in **??**).

Dictionaries: We also use a translation dictionary extracted from Wikipedia, a bilingual name dictionary extracted from the Buckwalter analyzer (Buckwalter, 2004) and an English synonym dictionary from the CIA World Factbook.[1] They are high precision and low recall: most errors do not have matches in the dictionaries, but when they do, they are often correct, particularly for NEs.

[1]http://www.cia.gov/library/publications/the-world-factbook

Background MT corpus: Since our motivation is CLQA, we also draw on a resource specific to CLQA: a background corpus of about 120,000 Arabic newswire and web documents that have been translated into English by a state-of-the-art industry MT system. Ma and McKeown (2009) were able to exploit a similar pseudo-parallel corpus to correct deleted verbs, since words deleted in one sentence are frequently correctly translated in other sentences.

For each error, the source-language phrase is converted into a query to search all three resources. Then the target-language results are aggregated and ranked by overall confidence scores. The confidence scores are a weighted combination of phrase translation probability, number of dictionary matches and term frequencies in the background corpus. The weights were set manually on a development corpus.

4.3 Rule-Based APE

Table 1 shows examples of sentences post-edited by the different APEs. For each error, the rule-based post-editor applies the top-ranked correction using one of two operations: *replace* or *insert*. An error can be replaced if there is an existing translation, and all of the source- and target-language tokens aligned to the error are flagged as errors. (This is to avoid over-writing a correct partial phrase translation, as in example 2a where the word "their" is not replaced.) If the error cannot be replaced, the new correction is inserted.

During *replace*, all the original target tokens are deleted, and the correction is inserted at the index of the first target token. For *insert*, the algorithm first chooses an insertion index, and then inserts the correction. The insertion index is chosen based on the indices of the target tokens in the error. If there are no target tokens, the insertion index is determined by the alignments of the neighboring source tokens. If they are aligned to neighboring translations, the correction is inserted between them. Or, if only one of them is aligned to a translation, the correction is inserted adjacent to it. If an insertion index cannot be determined via rules, the error is not corrected.

These editing rules are MT system-independent, language-independent and relatively simple. The word order is copied from the original translation or from the source sentence. This simple model worked for (Parton et al., 2008) because they were rewriting mistranslated NEs that were already present in the translation. Similarly, Ma and McKeown (2009) successfully reinserted deleted verbs into English translations us-

	Sentence	Sentence
Reference	Vanunu was released in April, 2004 ...	Why does Aramco donate 8 thousand dollars ...
MT A orig.	And was released in April, 2004 ...	Why ARAMCO to $ thousands ...
Rule-Based	And was vanunu released in April, 2004 ...	He donates why ARAMCO the amount of dollars to $ thousands ...
Corpus-Level	Vanunu was released in April, 2004 ...	Why Aramco donate $ 8 of thousands of dollars ...
	1a) Both APEs re-insert the deleted name, but the rule-based version has poor word order.	*1b) Both APEs re-insert the deleted verb, but the feedback word order is better. $ is incorrectly detected as a function word, and both APEs incorrectly re-insert "dollars". The feedback APE avoids adding the redundant "the amount of".*
Reference	...in proportion to the efforts they make.	...Ministry of Interior Starts to Define Committee's Authority!!
MT B orig.	...commensurate with their.	...The Ministry of Interior started to define the terms of the !
Rule-Based	...commensurate with effort exert their.	...The Ministry of Interior started to define the terms of body !
Phrase-Level	...commensurate with the work they do.	...The Interior Ministry started the authority of the board !
	2a) The rule-based APE makes two separate edits to insert "effort" and "exert." The feedback APE produces a more fluent sentence by handling both at once.	*2b) The original sentence deletes the noun Committee. The rule-based version has the wrong translation and is ungrammatical. The phrase-level feedback selects a better translation, but the verb (define) is now deleted.*

Table 1: Examples of the kinds of edits (both good and bad) made by different APEs.

ing only word alignments, assuming that local Chinese SVO word order would linearly map to English word order.

However, our APEs need to deal with a much wider range of error types, including phrases that were mistranslated, partially translated or never translated; and content words of any POS, not just NEs or verbs. Since Arabic word order differs from English, these rules often produce poorly ordered words: verbs may appear before their subjects, and adjectives may appear after their nouns. In this case, we are explicitly trading off fluency for adequacy, under the assumption that the end task is adequacy-oriented. In example 1a, the subject comes after the auxiliary verb, but the sentence can still be understood. On the other hand, since adequacy and fluency are not independent, degrading the fluency of a sentence can often negatively impact the adequacy as well.

Even when the error detection and correction steps work correctly, not all errors can be fixed with these simple operations. The original MT may be too garbled to correct, or may have no place to insert the corrected translation so that it carries the appropriate meaning.

4.4 Feedback APEs

To mitigate the problems of the rule-based APE, we developed an approach that is more powerful and flexible. The feedback APEs take as input the same list of errors and corrections as the rule-based APE, and then convert the corrections into feedback for the MT system. Sentences with detected errors are decoded a second time with feedback. Passing feedback to the MT system is a general technique: many MT systems allow users to specify certain fixed translations ahead of time, such as numbers, dates and named entities. The underlying implementation of how these fixed translations are handled by the decoder is MT system-specific, and we describe two such implementations in section 4.5: corpus-level feedback and phrase-level feedback.

The difference between pre-editing and post-editing in this case is that the post-editor is *reactive* to the first-pass translation. The APE only passes suggestions to the MT system when it detects an error in the first-pass translation, and has some confidence that it can provide a reasonable correction. Since the post-editing is actually done by the decoder, the effectiveness of the feedback APE will vary across different MT systems.

This is similar to the error correction approach described in (Parton and McKeown, 2010), where sentences with detected errors are re-translated using a much better (but slower) MT system. They found that the second-pass translations were much better than the first-pass translations, but most of the detected errors were still present. The feedback post-editor allows us to pass specific information about which errors to correct and how to correct them to the original MT system. Unlike adaptive post-editors, where the second translation step translates from "bad" target-language text to "good" target-language text, the feedback APEs re-translate from the source text, and only one MT system is needed.

The biggest advantage the feedback APEs have over the rule-based APE is that the MT system can modify the whole sentence during re-translation, while taking the feedback into account, rather than just replacing or inserting a single phrase at a time. The decoder will not permit local disfluencies that might occur from a simple insertion (e.g., "they goes" or "a impact"), and will often prefer the correct word order, as in example 1a in Table 1. Furthermore, the decoder can take all of the feedback into account at once, whereas the rule-based ap-

proach makes each correction in the sentence separately, as in example 2a. Finally, the rule-based approach always picks the top-ranked correction for each error, and almost always edits every error. The feedback APEs can pass multiple corrections to the MT system, often along with probabilities, which proves helpful in example 2b. One drawback of the feedback APEs is that they are slower than the rule-based APE since they require a second-pass decoding. Also, the decoder may ultimately decide not to use any of the corrections, which may be an advantage if low-confidence suggestions are discarded, or could be a disadvantage, since fewer errors will get corrected.

4.5 Corpus-Level vs. Phrase-Level Feedback

Each of our MT systems has a different mechanisms for accepting feedback on-the-fly, and handles the feedback differently. MT A allows *corpus-level feedback* without translation probabilities. In other words, the APE passes all of the translation suggestions for the entire corpus back to the MT system during re-translation. MT B allows *phrase-level feedback* with translation probabilities. Each source phrase flagged as an error is annotated with the list of possible corrections and their translation probabilities. Both MT systems allow multiple corrections for each detected error, unlike the rule-based APE. Both also allow the post-edited corrections to compete with existing translations in the system, so the re-translation may not use the suggested translations. Note that both forms of feedback are used in an online manner by the SMT systems; no re-training or re-tuning is done.

Overall, the phrase-level feedback mechanism is more fine-grained because corrections are targeted at specific errors. On the other hand, the coarser, corpus-level feedback could result in unexpected improvements in sentences where errors were not detected, since the translation corrections can be used in any re-translated sentence.

5 Experiments

We tested our APEs on two different MT systems using the NIST MT08 newswire (nw) and web (wb) testsets, which had 813 and 547 sentences, respectively. The translations were evaluated with multiple automatic metrics as well as crowd-sourced human adequacy judgments.

5.1 MT Systems

We used state-of-the art Arabic-English MT systems with widely different implementations. MT A was built using HiFST (de Gispert et al., 2010), a hierarchical phrase-based SMT system implemented using finite state transducers. It is trained on all the parallel corpora in the NIST MT08 Arabic Constrained Data track (5.9M parallel sentences, 150M words per language). The first-pass 4-gram language model (LM) is trained on the English side of the parallel text and a subset of Gigaword 3. The second-pass 5-gram LM is a zero-cutoff stupid-backoff (Brants et al., 2007) estimated using 6.6B words of English newswire text.

MT B was built using Moses (Koehn et al., 2007), and is a non-hierarchical phrase-based system. It is trained on 3.2M sentences of parallel text (65M words on the English side) using several LDC corpora including some available only through the GALE program (e.g., LDC2004T17, LDC2004E72, LDC2005E46 and LDC2004T18). The data includes some sentences from the ISI corpus (LDC2007T08) and UN corpus (LDC2004E13) selected to specifically add vocabulary absent in the other resources. The Arabic text is tokenized and lemmatized using the MADA+TOKAN system (Habash et al., 2009). Lemmas are used for Giza++ alignment only. The tokenization scheme used is the Penn Arabic Treebank scheme (Habash, 2010; Sadat and Habash, 2006). The system uses a 5-gram LM that was trained on Gigaword 4. Both systems are tuned for BLEU score using MERT.

5.2 Automatic and Human Evaluation

We ran several automatic metrics on the baseline MT output and the post-edited MT output: BLEU (Papineni et al., 2002), Meteor-a (Denkowski and Lavie, 2011) and TERp-a (Snover et al., 2009). BLEU is based on n-gram precision, while Meteor takes both precision and recall into account. TERp also implicitly takes precision and recall into account, since it is similar to edit distance. Both Meteor and TERp allow more flexible n-gram matching than BLEU, since they allow matching across stems, synonyms and paraphrases. Meteor-a and TERp-a are both tuned to have high correlation with human adequacy judgments.

In contrast to automatic system-level metrics, human judgments can give a nuanced sentence-level view of particular aspects of the MT. In order to compare adequacy across APEs, we used human annotations crowd-sourced from CrowdFlower.[2] Since our annotators are not MT experts, we used a head-to-head comparison rather than a 5-point scale. Adequacy scales have been shown

[2] http://www.crowdflower.com

MT	set	APE	sents w/err.	sents mod.
A	nw	rule-based	48%	41%
		corpus feed.	48%	40%
	wb	rule-based	69%	64%
		corpus feed.	69%	62%
B	nw	rule-based	24%	24%
		phrase feed.	24%	15%
	wb	rule-based	34%	34%
		phrase feed.	34%	25%

Table 2: The percentage of all sentences with errors detected, and the percentage of all sentences modified by each APE.

		Δ BLEU			Δ TERp-adeq			Δ Meteor-adeq		
MT	set	base MT	rule based	feed back	base MT	rule based	feed back	base MT	rule based	feed back
A	nw	51.32	−0.91	−0.41	37.49	−0.54	−0.74	69.48	+0.15	+0.32
	wb	36.15	−1.41	+0.03	60.66	−1.34	−2.69	55.24	+0.15	+0.88
B	nw	51.23	−0.49	+0.05	35.31	−0.22	−0.26	70.38	+0.00	+0.17
	wb	37.60	−0.50	−0.12	55.97	−0.26	−0.23	57.06	−0.07	+0.13

Table 3: The effect of APEs on automatic metric scores. Base columns show the score for the original MT and the other columns show the difference between the post-edited MT and the original MT. The rule-based APE is the same for both systems, and the feedback APE is corpus-level for MT A and phrase-level for MT B.

to have low inter-annotator agreement (Callison-Burch et al., 2007). Each annotator was asked to select which of two sentences matched the meaning of one reference sentence the best, or to select "about the same." The tokens that differed between the translations were automatically highlighted, and their order was randomized. The instructions explicitly said to ignore minor grammatical errors and focus only on how the meaning of each translation matched the reference, and included a number of example judgments.

We compared each post-edited sentence to the baseline MT. For each comparison, we collected five "trusted" judgments (as defined by CrowdFlower) according to how well they did on our gold-standard questions. For clarity, we are reporting results using macro aggregation, in other words, the number of times overall that a particular APE was voted better than, worse than, or about the same as the original MT.

6 Results

Table 2 shows the percentage of sentences with detected errors for which the correction algorithm found a suggested solution. These sentences were passed to each APE, which could then decide to modify the sentence or leave it unchanged. The percentage of all sentences that were changed by each APE is also shown in Table 2.

The web genre has more errors than the newswire genre, likely because informal text is more difficult for both MT systems to translate. MT A has twice as many sentences with detected errors as MT B. This is not a reflection of relative MT quality (both systems have comparable BLEU scores), but rather a limitation of the error detecting algorithm. When MT A deletes a word, it is frequently dropped as a single token, which is simple to detect as a null alignment. Missing words in MT B are frequently deleted as part of a phrase, so they are more difficult to detect (e.g., mistranslating "white house" as "white" does not get flagged).

The impact of the APEs also varies depending on how many sentences with detected errors were actually changed by the APE. The rule-based APE almost always applies the edits. The corpus-level APE also modified most of the sentences, since all of the corrections were applied to all of the re-translated sentences. However, the phrase-level feedback APE frequently retained the original translation.

Both of these factors mean that the potential improvement from post-editing varies significantly by experimental setting, from only 15% of the sentences by the phrase-based feedback (MT B) on the news corpus, up to 64% of the corpus by the rule-based APE for MT A on the web corpus.

6.1 Automatic Metric Results

Table 3 shows the automatic metric scores for both MT systems, across both datasets. For the baseline MT output, the raw score is shown, and for the APEs, the change in score between the post-edited MT and the baseline MT is shown. (Since post-editing only changes a fraction of sentences in the corpus, the score changes are generally small.)

All APEs improve the TERp-a score across all conditions[3], with the feedback APEs often outperforming the rule-based APE. The feedback APEs also improve the Meteor-a score across all conditions, while the rule-based APE has mixed Meteor results. None of the APEs improve the BLEU score: the rule-based APE is always significantly worse than the original MT, while the feedback APEs have either a negative or negligible impact.

The positive improvements in TERp-a and Meteor-a suggest that the APEs are improving adequacy. In general, the feedback APEs improve the automatic scores more than the rule-based APE, although the rule-based APE actually edits more sentences in the corpus than the feedback APEs.

[3] Since TERp is an error metric, smaller scores are better.

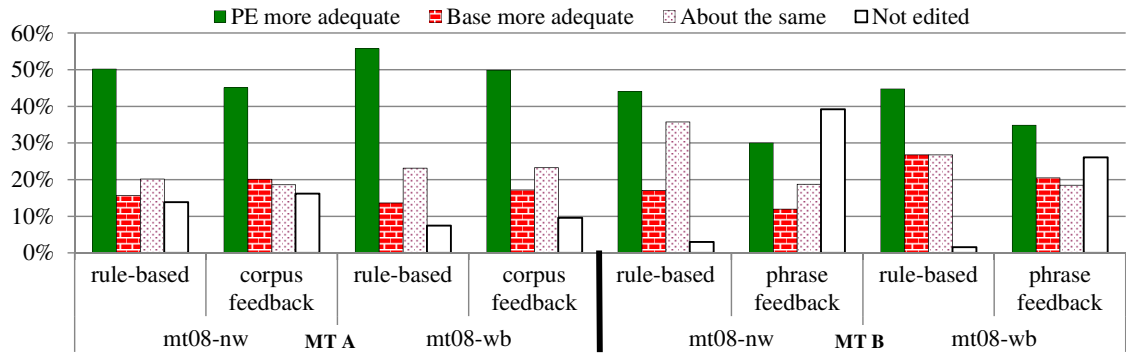

Figure 1: Percentage of post-edited sentences that were judged more adequate, less adequate or about the same as the original MT. "Not edited" is the percentage of sentences with errors that the APE decided not to modify.

The feedback APEs also always have better BLEU scores than the rule-based APE. The negative impact of APEs on BLEU score is not surprising, since they work by adding content to the translations, which is more likely to improve translation recall than precision.

6.2 Human-Annotated Adequacy Results

Figure 1 shows the percentage of post-edited sentences that were judged more adequate, less adequate or the same as the original MT, and the percentage of sentences with errors that the APE did not edit. Of the sentences that were post-edited, the APEs improved adequacy 30-56% of the time. Across both MT systems and both datasets, post-editing improved adequacy much more often than it degraded it: the ratio of improved sentences to degraded sentences varied from 1.7 to 4.1. For both MT systems, the APEs had a larger impact on the web corpus than the newswire corpus, both because more errors were detected in the web corpus and because the APEs edited errors more often in the web corpus.

We were surprised to find that the rule-based APE improved adequacy more often than the feedback APEs, across both MT systems and genres, especially given that the automatic metrics favored the feedback APEs. To understand the results better, we did another crowd-sourced evaluation, comparing the fluency of the rule-based and feedback post-edited sentences (when both APEs made changes). The sentences produced by the feedback APEs were judged more fluent than the rule-based APE sentences across all conditions.

The fluency evaluation shows the relative advantages of the different approaches. The rule-based APE does introduce new, correct information into the translations, but at the expense of fluency. With extra effort, the meaning of these sentences can usually be inferred, especially when the rest of the sentence is fluent (as in example 1a). On the other hand, the feedback APEs try to balance the post-editor's request to include more information in the sentence against the goal of the decoder to produce fluent output. But the need for fluency also led to fewer modified sentences, particularly for phrase-level feedback. In cases where both APE approaches improve the adequacy, the feedback approach is better because it produces more fluent sentences. But in cases where the feedback approach does not modify the sentence, the rule-based approach can often still improve the adequacy of the translation at the expense of fluency.

7 Conclusions and Future Work

We described several APE techniques: rule-based in addition to corpus-level and phrase-level feedback. Whereas previous APEs focused primarily on translation fluency and grammaticality, our APEs targeted adequacy errors. Manual analysis showed that post-editing was effective in improving the adequacy of the original MT output 30-56% of the time, across two MT systems and two text genres. The APEs had a larger impact on the web text than the newswire, indicating that they are particularly useful for hard-to-translate genres.

Manual evaluation of the APEs revealed a trade-off between fluency and control. The rule-based APE allowed control over which errors to correct and exactly how to correct them, but was limited to two basic edit operations that often led to disfluent sentences. The feedback APEs produced sentences that were more fluent, but they relied on MT decoders that might or might not carry out the corrections. The corpus-level feedback APE was the least targeted, because suggestions passed to the MT system could affect any re-translated sentence, even those where the phrase was translated correctly. Surprisingly, it was still able to improve adequacy. The phrase-level feedback APE allowed more targeted error correction, yet had the least

impact because it often ignored the corrections.

In future work, we plan to improve the error detection module to handle additional types of adequacy errors, in order to detect more of the adequacy errors made by MT B. We would also like to encourage the phrase-level APE to carry out our corrections more often. Another direction for research is including syntactic information in the rule-based APE, for more fluent translations.

The APEs were motivated by the CLQA task, where adequacy errors can make correct answers appear incorrect after translation. We believe that APE is particularly suitable for task-oriented MT, where black box MT systems must be adapted to the needs of a specific task. We plan to do a task-based evaluation of the adequacy-oriented APEs, to measure their impact on CLQA relevance.

Acknowledgments

This material is based upon work supported by DARPA under Contract Nos. HR0011-12-C-0016 and HR0011-12-C-0014. Any opinions, findings, and conclusions expressed in this material do not necessarily reflect the views of DARPA. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7-ICT-2009-4) under grant agreement number 247762.

References

Boschee, Elizabeth, Marjorie Freedman, Roger Bock, John Graettinger, and Ralph Weischedel. 2010. Error analysis and future directions for distillation. In *Handbook of Natural Language Processing and Machine Translation*.

Brants, Thorsten, Ashok C. Popat, Peng Xu, Franz J. Och, and Jeffrey Dean. 2007. Large language models in machine translation. In *EMNLP-CoNLL*, pp. 858–867.

Buckwalter, Tim. 2004. Buckwalter arabic morphological analyzer version 2.0. *LDC2004L02, ISBN 1-58563-324-0*.

Callison-Burch, Chris, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2007. (meta-) evaluation of machine translation. In *StatMT '07: Proc. of the Second WMT*, pp. 136–158.

Carpuat, Marine, Yuval Marton, and Nizar Habash. 2012. Improved arabic-to-english statistical machine translation by reordering post-verbal subjects for word alignment. *Machine Translation*, 26:105–120.

Condon, Sherri L., Dan Parvaz, John S. Aberdeen, Christy Doran, Andrew Freeman, and Marwan Awad. 2010. Evaluation of machine translation errors in English and Iraqi Arabic. In *LREC*.

de Gispert, Adrià, Gonzalo Iglesias, Graeme Blackwood, Eduardo R. Banga, and William Byrne. 2010. Hierarchical phrase-based translation with weighted finite-state transducers and shallow-n grammars. *Computational Linguistics*, 36(3):505–533.

Denkowski, Michael and Alon Lavie. 2011. Meteor 1.3: Automatic Metric for Reliable Optimization and Evaluation of Machine Translation Systems. In *EMNLP 2011: Proc. of the Sixth WMT*.

Doyon, Jennifer, Christine Doran, C. Donald Means, and Domenique Parr. 2008. Automated machine translation improvement through post-editing techniques: analyst and translator experiments. In *AMTA*, pp. 346–353.

Elming, Jakob. 2006. Transformation-based corrections of rule-based MT. In *EAMT*, pp. 219–226.

Finkel, Jenny Rose, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *ACL*, pp. 363–370.

Habash, Nizar, Owen Rambow, and Ryan Roth. 2009. MADA+TOKAN: A toolkit for Arabic tokenization, diacritization, morphological disambiguation, pos tagging, stemming and lemmatization. *Proc. of the 2nd International Conference on Arabic Language Resources and Tools (MEDAR)*, pp. 242–245.

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

Isabelle, Pierre, Cyril Goutte, and Michel Simard. 2007. Domain adaptation of MT systems through automatic post-editing. *MT Summit XI*.

Knight, Kevin and Ishwar Chander. 1994. Automated postediting of documents. In *AAAI '94*, pp. 779–784.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *ACL '07: Interactive Poster and Demonstration Sessions*, pp. 177–180.

Ma, Wei-Yun and Kathleen McKeown. 2009. Where's the verb?: correcting machine translation during question answering. In *ACL-IJCNLP*, pp. 333–336.

Mareček, David, Rudolf Rosa, Petra Galuščáková, and Ondrej Bojar. 2011. Two-step translation with grammatical post-processing. In *Proc. of the Sixth WMT*, pp. 426–432.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL*, pp. 311–318.

Parton, Kristen and Kathleen McKeown. 2010. MT error detection for cross-lingual question answering. In *COLING (Posters)*, pp. 946–954.

Parton, Kristen, Kathleen McKeown, James Allan, and Enrique Henestroza. 2008. Simultaneous multilingual search for translingual information retrieval. In *CIKM*, pp. 719–728.

Popović, Maja and Hermann Ney. 2007. Word error rates: Decomposition over POS classes and applications for error analysis. In *Proc. of the Second WMT*, pp. 48–55.

Sadat, Fatiha and Nizar Habash. 2006. Combination of arabic preprocessing schemes for statistical machine translation. In *Proceedings of the Conference of the Association for Computational Linguistics*, Sydney, Australia.

Simard, Michel, Cyril Goutte, and Pierre Isabelle. 2007. Statistical phrase-based post-editing. In *HLT-NAACL*, pp. 508–515.

Snover, Matthew, Nitin Madnani, Bonnie J. Dorr, and Richard Schwartz. 2009. Fluency, adequacy, or HTER?: exploring different human judgments with a tunable MT metric. In *StatMT '09: Proc. of the Fourth WMT*, pp. 259–268.

Specia, Lucia, Najeh Hajlaoui, Catalina Hallett, and Wilker Aziz. 2011. Predicting machine translation adequacy. In *MT Summit XIII*.

Stymne, Sara and Lars Ahrenberg. 2010. Using a grammar checker for evaluation and postprocessing of statistical machine translation. In *Proc. of the Seventh International Conference on Arabic Language Resources and Tools*.

Suzuki, Hirokazu. 2011. Automatic post-editing based on SMT and its selective application by sentence-level automatic quality evaluation. *MT Summit XIII*.

Toutanova, Kristina, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *NAACL-HLT*, pp. 173–180.

Vilar, David, Jia Xu, Luis Fernando D'Haro, and Hermann Ney. 2006. Error analysis of machine translation output. In *LREC*, pp. 697–702.