Learning to Automatically Post-Edit Dropped Words in MT

Jacob Mundt, Kristen Parton, Kathleen McKeown
Department of Computer Science
Columbia University
New York, NY 10027
jmm2328@columbia.edu,
{kathy,kristen}@cs.columbia.edu

Abstract

Automatic post-editors (APEs) can improve adequacy of MT output by detecting and reinserting dropped content words, but the location where these words are inserted is critical. In this paper, we describe a probabilistic approach for learning reinsertion rules for specific languages and MT systems, as well as a method for synthesizing training data from reference translations. We test the insertion logic on MT systems for Chinese to English and Arabic to English. Our adaptive APE is able to insert within 3 words of the best location 73% of the time (32% in the exact location) in Arabic-English MT output, and 67% of the time in Chinese-English output (30% in the exact location), and delivers improved performance on automated adequacy metrics over a previous rule-based approach to insertion. We consider how particular aspects of the insertion problem make it particularly amenable to machine learning solutions.

1 Introduction

Automatic post editors (APEs) use an algorithm to correct or improve the output of machine translation (MT). While human post editors have the intrinsic advantage of human linguistic knowledge, automatic post editors must have some other advantage over the MT system to be able to make improvements. The APE may have access to additional resources, either in the form of deeper contextual information or analysis unavailable to the decoder. Knight and Chander (1994) used additional analysis performed on the completed MT sentence to select determiners, while Ma and McKeown (2009) used redundancy in a question-answering task to help select better translations for verbs than were available in the MT phrase table. The APE may also have more knowledge about the specific translation goals of the system, allowing it to make different translation choices to better address those goals, even when selecting from the same phrase table. While MT systems trained on Bleu (Papineni et al., 2002) aim for fluency, Parton et al. (2012) used automatic post editing to adapt a black box MT system to prefer adequacy over fluency in a cross lingual question answering (CLQA) task where adequacy judgments determined task success.

Our motivation for improving adequacy is also CLQA, in our case over web forum data, as part of a new DARPA (Defense Agency Research Projects Agency) sponsored program called BOLT. CLQA system performance is evaluated by human relevance judgments comparing retrieved, translated passages to predetermined nuggets of information. As in Parton et al. (2012), an inadequate translation can cause an otherwise relevant passage to be judged irrelevant, so adequacy of MT is crucial to task performance. A critical problem in task-embedded translations is deletion of content words by MT systems and this is the focus of our work. Specifically, we are concerned with content words that are either translated into function words, or not translated at all in the MT output. These types of deletion are common in MT systems as a tradeoff to balance fluency and adequacy; Parton et al. (2012) detected these types of errors in 24% to 69% of sentences, with higher numbers of errors for web text over newswire copy. In our test sets, we also saw higher error rates for Chinese sources over Arabic.

Reference:
France and Russia are represented at both levels at the meeting...
MT:
It is both France and Russia at the meeting...

Figure 1. The MT drops the words "both levels", but the rephrasing of the rest of the sentence, while still expressing that France and Russia are at the meeting, presents no good place to reinsert "both levels".

A major challenge in automatic post editing, once the correct translation of a deleted word is found, is locating an insertion location that *maximizes adequacy*. This is a difficult problem for two reasons: first, the missing word was often dropped specifically to preserve fluency (to maximize the language model score). Additionally, phrases adjacent to a dropped word will typically be chosen to maximize fluency without the dropped word, as in Figure 1.

Parton et al. (2012) compare a rule-based automatic post editor with a feedback automatic post editor and for the rule-based approach use a simple alignment-based heuristic, inserting dropped content words adjacent to a partial translation if available, or between the translations of the dropped words' neighbors. In cases where the neighbors are not aligned to adjacent locations in the MT output, the correction is discarded. These heuristics provide reasonable results when translating between languages with similar word orders for the word being inserted and surrounding words. However, they can perform poorly in other cases; in translations from Arabic to English, subjects are often inserted after their verbs when the Arabic word order is VSO.

As an alternative to this heuristic, we present an approach for learning insertion positions from grammatical and positional features of the source sentence and aligned MT output. Since no gold standard training data is available for this problem, we also present a novel approach to generate high-adequacy insertion locations using reference translations. This method allows for better insertions of deleted words in languages with differing word order, improving adequacy of edited sentences. Further, in cases where Parton et al's heuristic method fails to determine an insertion point, this method can still succeed, allowing APE corrections to be applied in 14% more cases than their approach. Our evaluation using Chinese-English and Arabic-English MT systems shows that our insertion system can improve automated and human adequacy metrics in certain cases, when compared with both the original MT output and heuristic insertion.

2 Related Work

Our work builds on Parton et al. (2012) who compellingly show that a *feedback* and *rule-based* APE each have different advantages. The feedback post editor adds several potential corrections to the MT phrase table and feeds the updates back into another pass through the MT decoder, while the rule-based editor inserts the top-ranked correction directly into the original MT output. They found while the feedback system was preferred by the TERp (Snover et al., 2009) and Meteor (Denkowski and Lavie, 2011) automated adequacy metrics, the rule-based system was perceived to improve adequacy more often by human reviewers, often at the expense of fluency, noting that "with extra effort, the meaning of these sentences can usually be inferred, especially when the rest of the sentence is fluent." Our work attempts to increase adequacy through better insertion.

Previous general APE systems target specific types of MT errors, like determiner selection (Knight and Chandler, 1994), grammatical errors (Doyon et al., 2008), and adequacy errors (Parton et al. 2012). In contrast, fully adaptive APE systems try to learn to correct all types of errors by example, and can be thought of as statistical MT systems that translate from bad text in the target language to good text in the target language (Simard et al., 2007; Ueffing et al., 2008; Kuhn et al., 2011).

Similarly, Dugast et al. (2007) present the idea of statistical post editing, that is, using bad MT output and good reference output as training data for post editing. As their system proves more adept at correcting certain types of errors than others, they suggest the possibility of a hybrid post editing system, "breaking down the 'statistical layer' into different components/tools each specialized in a narrow and accurate area," which is similar to the approach followed in this paper. Isabelle et al. (2007) also use learning methods to replace the need for a manually constructed post editing dictionary. While they study a corpus of MT output and manually post-edited text to derive a custom

dictionary, our system attempts to learn the rules for a specific type of edit: missing word insertion.

Taking a statistical approach to system combination, Zwarts and Dras (2008) built a classifier to analyze the syntax of candidate translations and use abnormalities to weed out bad options. Our classifier could be seen as a special case of this, looking for an area of bad syntax where a word was potentially dropped. As noted though, the MT system's language model often "patches up" the syntax around the missing word, leading to areas that are syntactically valid, though inadequate.

The TER-Plus metric (Snover et al., 2009) provides a variety of techniques for aligning a hypothesis to a reference translation, as well as determining translation adequacy amongst deletions and substitutions. While we use TER-Plus as a metric, we also use it as a guide for determining where a missing word should be inserted to maximize adequacy against a reference. While our effort focuses on learning the highest adequacy insertion from examples with reference translations, there is significant work in trying to assess adequacy directly from source and target, without references (Specia et al., 2011; Mehdad et al., 2012).

3 Method

The APE has 3 major phases: error detection, correction, and insertion. The first two phases are performed identically as described in Parton et al. (2012) and will be summarized briefly here, while the third phase differs substantially and will be described in greater detail.

3.1 Input and Pre-processing

We constructed two separate pipelines for Arabic and Chinese. The Arabic data was tagged using MADA+TOKEN (Habash et al., 2009). Translated English output was recased with Moses, and POS and NER tags were applied using the Stanford POS tagger (Toutanova et al., 2003) and NER tagger (Finkel et al.,2005).

For Chinese data, POS tags were applied to both source and output using the Stanford POS tagger (Toutanova et al., 2003).

3.2 MT systems

The Arabic MT system is an implementation of HiFST (de Gispert et al., 2010) trained on corpora from the NIST MT08 Arabic Constrained Data track (5.9M parallel sentences, 150M words per language). The Chinese MT system is the SRInterp system, developed by SRI for the DARPA BOLT project, based on work discussed in Zheng et al. (2009). It was trained on 2.3 million parallel sentences, predominantly newswire with small amounts of forum, weblog, and broadcast news data.

3.3 Error Detection and Correction

Errors are detected by locating mistranslated named entities (for Arabic only) and content words that are translated as function words or not translated at all, by looking at alignments and POS tags (Parton and McKeown, 2010).

Arabic error corrections are looked up in a variety of dictionaries, including an MT phrase table with probabilities from a second Arabic MT system, Moses (Koehn et al., 2007), using data from the GALE program available from LDC (LDC2004T17, LDC2004E72, LDC2005E46, LDC2004T18, LDC2007T08, and LDC2004E13). Secondary sources include an English synonym dictionary from the CIA World Factbook[1], and dictionaries extracted from Wikipedia and the Buckwalter analyzer (Buckwalter, 2004). Arabic additionally uses a large parallel background corpus of 120,000 Arabic newswire and web documents and their machine translations from a separate, third Arabic MT system, IBM's Direct Translation Model 2 (Ittycheriah 2007).

Chinese corrections are looked up in the phrase table of our Chinese MT, SRI's SRInterp system (Zheng et al., 2009), and also in a dictionary extracted from forum data, Wikipedia and similar sources (Ji et al., 2009; Lin et al., 2011).

3.4 Synthesizing a Gold Standard

Once an error is detected and a high-probability replacement is found, it must be inserted into the existing MT output. The straightforward solution is to use standard machine learning techniques to adapt to the translation errors made by a specific MT system on a specific language, but doing this is

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

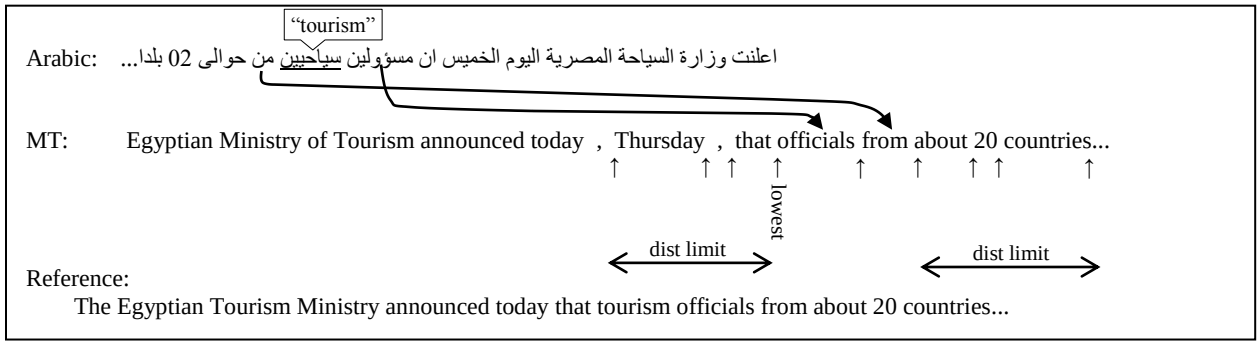

Figure 2. Synthesizing gold standard training data. The APE selects the nearly-correct alternative "tourist" for "سياحيين", and then TERp scores are evaluated at several potential insertion locations, up to a defined distortion limit from the source word's neighbors. The gold standard location is chosen as the one with the best (lowest) TERp score; here, the location is not between its neighbors, but before both of them.

complicated by the lack of training data. One option would have been to have human annotators select insertion locations for all the corrections detected above.

We took a different approach and elected to synthesize gold standard data. Since we had reference translations for our translation data (4 references for each Arabic sentence and 1-4 for each Chinese sentence), we exploited these sentences to find highly-probable correct insertion locations for each correction.

The TER-Plus metric (Snover et al., 2009) generates an adequacy score by penalizing deletions, insertions, substitutions, and shifts, in addition to allowing stem matches, synonym matches, and paraphrasing. This often allows it to calculate a set of shifts that largely align MT output to a reference, even when the MT output uses significantly different words and ordering.

If the missing word appears in any of the reference translations, TER-Plus is evaluated repeatedly, comparing that reference to the MT output with the missing word inserted at each possible insertion point, to find the location that is most aligned to the reference—the location with the lowest TER-Plus score (Figure 2). Similar to many statistical MT systems, we impose a hard distortion limit, trained on development data, that prevents words from moving more than a set amount from their neighbors' aligned output phrases. In fact, the insertion heuristic presented in Parton et al. (2012) can be thought of as having a distortion limit of 1.

Although the detected gold standard locations typically correspond with human judgments of "correctness" on where a missing word should be inserted, another view is that the classifier is learning to insert at the location that maximizes the TER-Plus score for the output sentence, which should at least raise the score over the heuristic method. It should be noted that not all sentences with detected errors will generate valid gold standard insertion locations. When the missing word does not appear in any of the reference translations, we discard the sentence from our insertion training data and do not attempt to find the highest TERp insertion.

3.5 Training and Insertion

Once we have a set of synthesized gold standard training data, a standard MT classifier can be trained to recognize good insertion locations. We used the BayesNet classifier from Weka (Hall et al., 2009). In addition to giving good results on recognition of individual insertions, it also reports classification probabilities rather than binary output. Since we have to choose amongst a number of insertion locations, this allows us to choose the highest confidence insertion location.

Machine learning is particularly well-suited to this problem. It allows easy adaptation to different languages and MT systems. Secondly, by tuning the system for high recall, we can bias the system towards making edits rather than leaving the sentence unchanged. In adequacy-focused tasks, leaving the sentence without a content word is often a poor choice, and an incorrect insertion location, if not too far from the correct point, can result either in improvement, or in no perceived change: as noted, humans are good at making sense of misordered translations. Of course, a bad insertion can degrade accuracy, but prediction errors occur more

	N	error	edit RB	edit ML	gold
Train					
Arabic	4115	54%	37%	-	842 (21%)
Chinese	6318	63%	28%	-	679 (11%)
Test					
Arabic	813	60%	41%	47%	168 (21%)
Chinese	1470	58%	25%	31%	201 (14%)

Table 1. Data details, showing the total number of sentences in each set (**N**), the percentage with a detected dropped or mistranslated word error (**error**), and the percent of sentences that were edited by the rule-based APE (**edit RB**) and the adaptive APE (**edit ML**). Note that only 10-20% of data can be used as synthetic gold standard data (**gold)** for machine learning. For test data, the adaptive post editor is able to edit more sentences than the heuristic rule based one.

	N	exact	within 1	within 3	mean error
Arabic	168	32%	52%	73%	1.81
Chinese	244	30%	46%	67%	2.32

Table 2. Classifier accuracy when determining word insertion location.

often on sentences that already have poor translations.

The features used for each potential insertion location are positional and syntactic:

- **Insertion point location**: relative and absolute location where insertion is being considered.
- **Neighbor offsets**: relative offset from the English phrases aligned to the word's source language neighbors.
- **Partial translation offset**: relative offset from a partial translation, for cases where a content word was translated as a function word.
- **Part-of-speech**: POS tag of the left and right neighbors of the insertion location, and bigrams of these neighbors and the POS of the word being inserted.
- **Simplified part-of-speech**: same as above, but POS tags are mapped to a simple, language agnostic set first.

Feature selection was performed on our original feature set using Weka's Chi Squared method, which indicated that the offset and POS unigram features were the most useful. We noticed that for our training set size, the POS bigram features led to overfitting and poor results on unseen data. By creating a smaller set of simplified tags and tag bigrams, we were able to retain some very shallow syntactic information while avoiding overfitting.

To use the trained classifier for insertion on test data, we simply run the classifier on each possible insertion point (within the hard distortion limit) for a missing word, and choose the insertion point with the highest positive confidence as the predicted insertion location.

4 Experiments

The Arabic training data consisted of 4115 sentences sampled from past years of the NIST Arabic-English task MT02 – MT05, each with 4 reference translations, and the Arabic test data was 813 sentences from the NIST MT08 newswire set. The Chinese training data consists of 6318 sentences, combined from forum, weblog data, and newswire data from NIST Chinese MT08 eval set and the DARPA GALE project. The Chinese test data is the NIST Chinese MT06 eval set, 1470 sentences. All data had at least one reference, and some sources included up to four.

We tested three automated metrics on the baseline MT output, output from the original rule-based APE described in Parton et al. (2012), and output from the APE with adaptive insertion on both Chinese and Arabic. Metrics are BLEU (Papineni et al., 2002), Meteor and TERp. Since BLEU is based on strict matching of bigrams, we do not expect post editing to improve the BLEU score in most cases, since it is rare that both the word inserted and its neighbors match the reference translation exactly. Meteor and TERp include adequacy and so should be more representative of the performance of our improved insertion algorithm. Note that TERp was also used to train our insertion system as well; one way of viewing the classifier is as a predictor for high-TERp insertion locations,

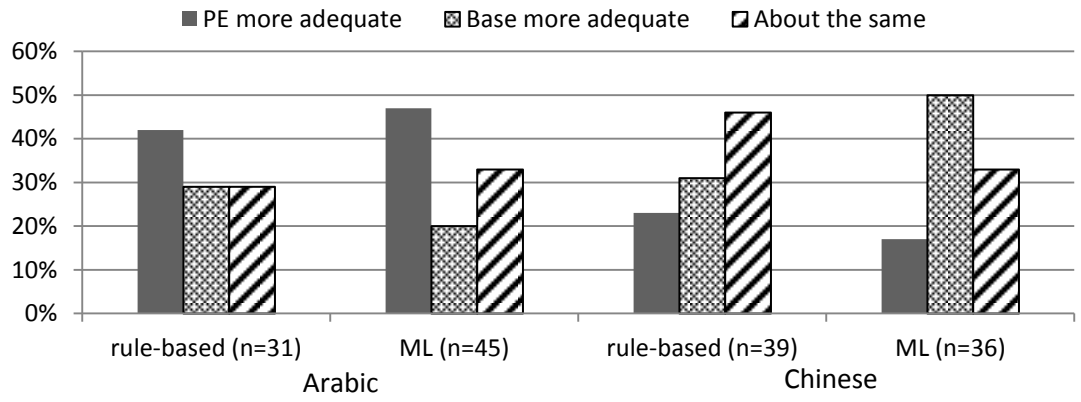

Figure 3. Human judgements on automatically post edited sentences. For each language, the results for the rule-based heuristic insertion algorithm is shown, along with our probabilistic ML approach. The total number of human comparisons performed is shown for each experiment.

trained on insertion locations that were shown to maximize TERp scores in the training set.

We also report the classifier results, showing what percentage of sentences were used to generate the synthetic gold standard, how often our classifier predicted the gold standard answer, and the average difference between our predicted insertion location and the gold standard.

5 Results

We are able to generate gold standard data for around 10-20% of the data using the TERp-based method described above, depending on the specific language (Table 1). The remaining cases do not have synthetic gold standard data, because it was not possible to align the word to be inserted with any of the provided references.

One clear advantage of the machine learning-based post editor is the ability to edit more sentences, as seen in Table 1. The rule-based editor cannot edit a sentence when the neighbors of the dropped word in the source are aligned to non-adjacent words in the MT output. The classifier in the adaptive editor always returns the highest-likelihood location within the distortion limit.

Turning to actual classifier accuracy, the exact gold standard insertion location is predicted 30% of the time in Chinese and 32% of the time in Arabic (Table 2). This is a meaningful result, since this is a multiclass prediction problem (where the number of possible places to insert is always at least twice the distortion limit). Also, the classification problem is continuous in some respects. Getting an insertion location *near* the correct one is better than getting one far away. We can predict the answer within 3 of the gold standard location 67-73% of the time. The mean error (in words) from the correct location is under 2 for Arabic and slightly higher for Chinese.

A simple human comparison was also performed, presenting the base MT output and the output of the APE, along with a reference translation, to 6 human annotators, who were asked to judge whether the APE was more adequate, the baseline was more adequate, or that the two translations had about the same adequacy. The number of human comparisons performed is noted in Figure 3 for each experiment. While we have a small number of survey results, the ML approach is preferred 47% of the time in Arabic, versus 42% for the rule-based APE. The ML APE also degrades only 20% of the Arabic sentences, whereas the rule-based system degrades 29%. This suggests that some of the degraded sentences were degraded because of a correct word inserted in an incorrect location.

Both APEs do significantly worse overall in Chinese, but the ML APE performs more poorly than the rule-based APE, both on number of sentences improved and number of sentences degraded. There may be attributes of the Chinese language that make reinsertion more difficult, but Chinese also had nearly 20% less training data than Arabic, and this may indicate that the performance of the ML APE suffered because of this.

6 Conclusions and future directions

We showed that a statistical approach to reinserting missing words is a feasible tactic, often able to predict locations near the correct location and sometimes even predicting the insertion location exactly. Though the insertion problem did not have human labeled gold standard data, we were able to generate it from reference translations. We also showed that the statistical approach can edit slightly more sentences than the original heuristic APE, leading to more adequacy improvements. Initial human judgments indicate that the statistical method increases adequacy in Arabic when compared with the rule-based approach, but is unable to improve adequacy in Chinese, possibly due to limited training data.

One area to be investigated is other methods for generating training data. Our TERp-based method requires that the inserted word (or a stem/synonym) be in the reference translation, but more flexible approaches may be possible using source and target POS tags or even full parses. Even better would be an approach that does not rely on reference translations, since this requirement limits the amount of training data we can generate. While earlier attempts have shown that purposely deleting words from correct English sentences provides poor training examples (since the "missing" areas are not adjusted by the language model to appear fluent), it may be possible to post-process the sentences after deletion, or even delete words from source sentences and then translate them.

Additionally, one continuing problem with this approach is the inability to apply more complicated modifications near the insertion point beyond simple insertion and replacement. Learning to apply more complicated changes (deleting nearby function words, fixing tense, determiners, and agreement) may be possible with sufficient training data and may help to improve fluency, rather than focusing almost exclusively on adequacy as we did here. This would be especially helpful in sentences with insertions located in contiguous areas of the sentence.

References

Tim Buckwalter. 2004. Buckwalter Arabic morphological analyzer version 2.0. Linguistic Data Consortium, University of Pennsylvania, 2004. LDC Catalog No.: *LDC2004L02*.

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

Michael Denkowski and Alon Lavie. 2011. Meteor 1.3: Automatic Metric for Reliable Optimization and Evaluation of Machine Translation Systems. In *Proceedings of the EMNLP 2011 Workshop on Statistical Machine Translation*.

Jennifer Doyon, Christine Doran, C. Donald Means, and Domenique Parr. 2008. Automated machine translation improvement through post-editing techniques: analyst and translator experiments. In *Proceedings of the 8th AMTA*, pages 346-353.

Loïc Dugast, Jean Senellart, and Philipp Koehn. 2007. Statistical Post-Editing on SYSTRAN's Rule-Based Translation System. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 220-223.

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

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

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten. 2009. The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, Volume 11, Issue 1.

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

Abraham Ittycheriah and Salim Roukos. 2007. Direct Translation Model 2. In *Proceedings of NAACL HLT 2007*, pages 57–64.

Heng Ji, Ralph Grishman, Dayne Freitag, Matthias Blume, John Wang, Shahram Khadivi, Richard Zens and Hermann Ney. 2009. Name Translation for Distillation. *Handbook of Natural Language Processing and Machine Translation: DARPA Global Autonomous Language Exploitation*.

Kevin Knight and Ishwar Chander. 1994. Automated postediting of documents. In *AAAI '94: Proceedings of the twelfth national conference on Artificial intelligence (vol. 1)*, pages 779–784, Menlo Park, CA, USA. American Association for Artificial Intelligence.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi,

Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondˇrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In ACL '07: Interactive Poster and Demonstration Sessions, pages 177–180.

Roland Kuhn, Jean Senellart, Jeff Ma, Antti-Veikko Rosti, Rabih Zbib, Achraf Chalabi, Loïc Dugast, George Foster, John Makhoul, Spyros Matsoukas, Evgeny Matusov, Hazem Nader, Rami Safadi, Richard Schwartz, Jens Stephan, Nicola Ueffing, and Jin Yang. 2011. Serial System Combination for Integrating Rule-based and Statistical Machine Translation. In *Handbook of Natural Language Processing and Machine Translation: DARPA Global Autonomous Language Exploitation*, Joseph Olive, Caitlin Christianson, John McCary (Eds.), Springer, pages 361-374.

Wen-Pin Lin, Matthew Snover and Heng Ji. 2011. Unsupervised Language-Independent Name Translation Mining from Wikipedia Infoboxes. In *Proc. EMNLP2011 Workshop on Unsupervised Learning for NLP*.

Wei-Yun Ma and Kathleen McKeown. 2009. Where's the verb?: correcting machine translation during question answering. In *ACL-IJCNLP '09: Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 333–336, Morristown, NJ, USA. Association for Computational Linguistics.

Yashar Mehdad, Matteo Negri, Marcello Federico. 2012. Match without a Referee: Evaluating MT Adequacy without Reference Translations. In *Proceedings of the 7th Workshop on Statistical Machine Translation*, pages 171–180.

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

Kristen Parton, Nizar Habash, Kathleen McKeown, Gonzalo Iglesias, Adria de Gispert. 2012. Can Automatic Post-Editing Make MT More Meaningful? In *Proceedings of the 16th EAMT Conference*.

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

Matthew Snover, 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: Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 259–268, Morristown, NJ, USA. Association for Computational Linguistics.

Lucia Specia, Najeh Hajlaoui, Catalina Hallett and Wilker Aziz. Predicting Machine Translation Adequacy. 2011. *Machine Translation Summit XIII*, September, Xiamen, China.

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

Nicola Ueffing, Jens Stephan, Evgeny Matusov, Loïc Dugast, George Foster, Roland Kuhn, Jean Senellart, and Jin Yang. 2008. Tighter Integration of Rule-based and Statistical MT in Serial System Combination. In *COLING 2008*, pages 913-920.

Jing Zheng, Necip Fazil Ayan, Wen Wang, and David Burkett. 2009. Using Syntax in Large-Scale Audio Document Translation. In *Proc. Interspeech 2009*, Brighton, England.

Simon Zwarts, Mark Dras. 2008. Choosing the Right Translation: A Syntactically Informed Classification Approach. In *COLING 2008*, pages 1153-1160.