Domain adaptation of MT systems through automatic post-editing

Pierre Isabelle, Cyril Goutte and Michel Simard
Interactive Language Technologies Group
NRC Institute for Information Technology
283 Alexandre-Taché Boulevard
Gatineau (Quebec), Canada J8X 3X7
FirstName.LastName@nrc-cnrc.gc.ca

Abstract

It is generally acknowledged that the performance of rule-based machine translation (RMBT) systems can be greatly improved through domain-specific system adaptation. To that end, RBMT users often choose to invest significant resources into the development of ad hoc MT dictionaries. In this paper, we demonstrate that comparable customization effects can be achieved automatically. One effective way to do that is to post-edit the translations produced by a vanilla RBMT system using a specially-trained statistical machine translation (SMT) system. Our experiments indicate that this method is just as effective as manual customization of system dictionaries in reducing the need for manual post-editing.

1 Introduction

When high quality translations are needed, raw machine translation output is almost invariably considered to be inadequate. But if the machine output is reasonably good, it is sometimes possible to use it as a draft translation to be manually post-edited by a human translator. The hope, of course, is that MT post-editing will prove more efficient than manual translation.

While there does exist some documented cases of success, MT postediting has not really become mainstream among professional translators. In fact, most translators still consider that MT output is more of a nuisance than anything else in their work. Their main reasons are: 1) MT output contains too many errors; and 2) unlike humans, MT systems fail to learn from the post-editor's corrections.

The first problem is traditionnally addressed in part through system adaptation. The performance of a vanilla RBMT system can usually be improved a great deal by the addition of domain- or user-specific dictionaries. Unfortunately, the development of such dictionnaries often proves overly difficult or costly.

We will show in this paper that the issue of system adaptation can be reduced to the second issue mentioned above, namely the machine's ability to learn from the post-editor's work. Tentative post-editors are often dishearted by the need to correct the same mistakes over and over again. Fortunately, recent work on automatic post-editing (APE) of machine translations indicates that it is possible to build systems that can learn from human post-edits. In the first published proposal about the concept of APE, (Knight and Chander, 1994) consider the possibility of building "adaptive post-editors", that is:

> " an automatic program that can watch a human postedit documents, see which errors crop up over and over (these will be different for any given system/domain pair), and begin to emulate what the human is doing." (p. 779)

Their paper does not explore that possibility in any detail, though. (Allen and Hogan, 2000) propose the development of a "processing engine that could automatically fix up machine translation raw output before such texts are even given to a human posteditor". They are not very explicit about how such an engine would operate but their discussion suggests that it would be fed with manually developed sets of post-editing rules. (Elming, 2006) presents the first published results on the use of an APE module to correct MT output. He uses *transformation-based learning* to correct the output of the Patrans RBMT system and reports a 4.6 point increase in BLEU score.

In (Simard et al., 2007a) we presented a set of experiments on using standard phrase-based SMT technology to build an APE module for a classical rule-based MT system. The APE task is then viewed as a machine translation task in which our SMT system "translates" from the language of RBMT outputs into the language of their manually post-edited counterparts. In the experiments reported in the paper, the addition of such an APE module resulted in very substantial gains in translation quality (figures will be provided below).

Once we realize that APE is feasible, we are led to ask to what extent it could be used as a means of customizing a vanilla MT system. This question is a natural one because system customization is carried out precisely as a means to fix problems that have been observed in previous translations or are expected to pop up in future ones. The rest of this paper will attempt to provide an answer to that question.

In section 2, we position the present work with respect to our ongoing research on SMT and APE. In section 3 we present the *Job Bank* data that we will be using in our experiments. In section 4, we report on our experiments comparing the amount of manual work left to the post-editor under a range of different MT conditions: SMT versus RBMT, non-adapted versus adapted RBMT and pres-

ence or absence of an APE module. Our main finding is that in combination with the SMT-based APE module, the vanilla RBMT system performs almost as well as its manually adapted version. In other words, the APE has succeeded in capturing whatever benefits were brought by manual system adaptation. In section 5 we discuss that result before concluding the paper.

2 SMT and Automatic Post-Editing

The work reported here is based on the paradigm of phrase-based statistical machine translation (Marcu and Wong, 2002; Koehn et al., 2003). Our PORTAGE SMT system (Sadat et al., 2005) is a typical exemplification of that paradigm. In recent months, we tested it in the role of an APE module for a commercial RBMT system (Simard et al., 2007a), and in this paper we show that APE constitutes a good way to adapt the RBMT system to a new domain.

2.1 The PORTAGE SMT system

PORTAGE is a phrase-based, statistical machine translation system, developed at the National Research Council of Canada (NRC) (Sadat et al., 2005).[1] Like other SMT systems, it learns to translate from existing parallel corpora.

The system translates text in three main steps: preprocessing of raw data into tokens; decoding to produce one or more translation hypotheses; and error-driven re-scoring to choose the best hypothesis. For languages such as French and English, the first of these steps (tokenization) is mostly a straightforward process and we do not describe it any further here. Decoding is the central task in SMT, involving a search for the hypotheses t that have highest probabilities of being translations of the current source sentence s according to a model of $P(t|s)$. PORTAGE implements a dynamic programming beam search decoding algorithm similar to that of Koehn (2004), in which translation hypotheses are constructed by combining in various ways the target-language part of phrase pairs whose source-language part matches the input. These phrase pairs come from large *phrase tables* constructed by collecting matching pairs of contiguous text segments from word-aligned bilingual corpora.

PORTAGE's model for $P(t|s)$ is a log-linear combination of four main components: one or more n-gram target-language models, one or more phrase translation models, a distortion (word-reordering) model, and a sentence-length feature. The phrase-based translation model is similar to that of Koehn, with the exception that phrase probability estimates $P(\tilde{s}|\tilde{t})$ are smoothed using the Good-Turing technique (Foster et al., 2006). The distortion model is also very similar to Koehn's, with the exception of a final cost to account for sentence endings.

Feature function weights in the log-linear model are set using Och's minimum error rate algorithm (Och, 2003). This is essentially an iterative two-step process: for a given set of source sentences, generate n-best translation hypotheses, that are representative of the entire decoding search space; then, apply a variant of Powell's algorithm to find weights that optimize the BLEU score over these hypotheses, compared to reference translations. This process is repeated until the set of translations stabilizes, i.e. no new translations are produced at the decoding step.

To improve raw output from decoding, PORTAGE relies on a re-scoring strategy: given a list of n-best translations from the decoder, the system reorders this list, this time using a more elaborate log-linear model that incorporates additional feature functions, over and above those used at the decoding stage. The extra feature functions include IBM-1 and IBM-2 model probabilities (Brown et al., 1993) and an IBM-1-based feature function designed to detect whether any word in one language appears to have been left without satisfactory translation in the other language; all of these feature functions can be used in both language directions, i.e. source-to-target and target-to-source.

2.2 Previous results on SMT-based APE

Translation post-editing can be viewed as a transformation process that takes as input raw target-language text coming from an MT system and produces as output target-language text in which "errors" have been corrected. Viewed in that way, it is conceptually similar to the translation task itself. Thus, there doesn't seem to be any a priori reason why a machine translation system could not handle the APE task. Indeed, assuming that the kind of data described in section 3 is available, the idea of using a statistical MT system for post-editing is appealing because the underlying models are fully general and the technology is now widely available.

In (Simard et al., 2007a) we presented the results of some experiments on the use of PORTAGE to post-edit the output of a commercial rule-based machine translation system which is currently being used by the Canadian government's department of Human Resources and Social Development (HRSDC) to translate job ads between English and French. One important result of our experiments was that the combined RBMT + APE system performed substantially better than the RBMT system alone. Specifically, the APE step turned out to reduce the amount of manual post-editing needed by about one third in the French-to-English direction and by about 12% in the English-to-French direction. Another interesting finding of the same paper is that in the case of relatively small training sets (less than 500K words), the translations produced by the combined RBMT + PORTAGE APE were significantly better than those produced by PORTAGE as a standalone SMT system.

The detailed results of these APE experiments will be presented below, as part of a larger set that includes our new results on domain adaptation.

2.3 APE as domain adaptation

Current SMT systems tend to be heavily domain-dependent: they are usually trained from scratch on a domain-specific corpus. In contrast, commercial RBMT systems are usually provided in a generic ("vanilla") version that can be used to translate in any domain whasoever. However, their out-of-the-box performance will not be optimal, especially in cases where the texts to be translated

[1] A version of the PORTAGE system is made available by the NRC to Canadian universities for research and education purposes.

belong to a highly specialized domain. A significantly better performance can be attained if the user is prepared to invest in adapting the system to the relevant text domain.

The relevant adaptations can in principle cover various components of the RBMT system: dictionaries, syntax, semantic rules, etc. Indeed, some of the most successful RBMT systems to date have been developed from scratch to translate particular "sublanguages" (Kittredge and Lehrberger, 1983). But in most cases, adaptation will merely consist of providing a domain-specific dictionary. In fact, most commercial systems will not allow their users to make changes to other system components in view of the high level of technological expertise that is required for doing it successfully.

Thus, RBMT users sometimes choose to invest significant resources into the development of their own domain-specific MT dictionaries. In the experiments described below, we will quantify the impact of one such effort on MT quality. We will then investigate to what extent an APE module would be able to produce a comparable impact.

3 A case study

The Canadian government's department of Human Resources and Social Development (HRSDC) maintains a web site called *Job Bank*,[2] where potential employers can post ads for open positions in Canada. Over one million ads are posted on *Job Bank* every year, totalling more than 180 million words. By virtue of Canada's Official Language Act, HRSDC is under legal obligation to post all ads in both French and English. In practice, this means that ads submitted in English must be translated into French, and vice-versa.

To address this task, the department has put together a complex setup, involving text databases, translation memories, machine translation and human post-editing. Employers submit ads to the *Job Bank* website by means of HTML forms containing "free text" data fields. Some employers do periodical postings of identical ads; the department therefore maintains a database of previously posted ads, along with their translations, and new ads are systematically checked against this database. The translation of one third of all ads posted on the *Job Bank* is actually retrieved in this way. Also, employers will often post ads which, while not entirely identical, still contain identical sentences. The department therefore also maintains a translation memory of individual sentence pairs from previously posted ads; another third of all text is typically found *verbatim* in this way.

The remaining text is submitted to a machine translation system, enriched with dedicated resources and lexicons. The MT output is then post-edited by human experts. There are currently as many as 20 post-editors working full-time, most of whom are junior translators.

HRSDC kindly provided us with a sample of data from the *Job Bank*. This corpus consists of a collection of parallel "blocks" of textual data. Each block contains four parts: 1) a source language text S as submitted by the employer; 2) a translation T_1 of S produced by the vanilla,"out of the box" version of a commercial RBMT system; 3) a translation T_2 of S produced by a lexically adapted version of the same RBMT system; and 4) a reference translation T_R that has been manually post-edited.

[2] http://www.jobbank.gc.ca

	En-to-Fr	Fr-to-En
T_1 (vanilla RBMT)	62.60	69.96
T_2 (customized RBMT)	53.33	58.77

Table 1: Translation error rates (TER) observed for the vanilla (T_1) and lexicallly adapted (T_2) versions of the RBMT system, relative to reference translations (T_R).

The source language side of the corpus contains less than half a million words in each of French and English. In volume, this corresponds to less than a week of *Job Bank* data. Basic corpus statistics are provided in Table 2 (see Section 4). Most blocks contain only one sentence, but some span several sentences. The longest block contains 401 tokens over several sentences. Overall, blocks are quite short: the median number of tokens per source block is only 9 for French-to-English and 7 for English-to-French.

Our main metric for evaluating the quality of the post-editing will be the Translation Edit Rate (TER, cf. Snover et al. (2006)). The TER counts the number of edit operations, including phrasal shifts, needed to change a hypothesis translation into an adequate and fluent sentence, and normalised by the length of the final sentence. Although the question of the merits of TER with respect to other established MT evaluation measures is open to (lively) discussion, it should be noted that our focus is on reducing the *post-editing effort*. The TER closely corresponds to the amount of post-editing work performed on the *Job Bank* application, and therefore appears to be the most suitable metric for a task-based evaluation. This motivates the choice of TER as our main metric, although we also report our experimental results using the more traditional BLEU score (Papineni et al., 2002).

Table 1 shows the global TER figures obtained by comparing the respective outputs of the vanilla and lexically adapted versions of the RBMT system to their manually post-edited references over the whole corpus. For example, the TER figure of 69.96 for French-to-English translation indicates that on average almost 7 out of 10 words produced by the vanilla version of the RBMT system (T_1) need to be edited to produce the reference translation (T_R). The score of 62.60 for the other language direction is just slightly better. We can clearly see the impact of adding domain-specific dictionaries (T_2): in both language directions, the TER drops by about 10 points. However, it remains above the 50% mark.

Clearly a lot of post-editing effort appears to be required from the *Job Bank* translators. The apparent harshness of this result is somewhat mitigated by two factors. First, the distribution of the block-based TER shows a large disparity in performance (Simard et al., 2007a). On one hand, some MT output has to be entirely rewritten; on the other hand, more than 10% of the blocks have a TER of 0. The global score therefore hides a large range of performance. The second factor is that the TER measures the distance to

an adequate *and fluent* result. A high TER does not mean that the raw MT output is not understandable, but just that a substantial amount of manual post-editing is required in order to make it fluent.

We now turn to some experiments on the benefits of learning an automatic post-editing module and the potential of such a module in domain adaptation.

4 Experimental results

In each direction (French-to-English and English-to-French), we held out from the *JobBank* corpus two subsets of approximately 1000 randomly picked blocks. The *validation* set is used for testing the impact of various high-level choices such as pre-processing, or for obtaining preliminary results based on which we set up new experiments. The *test* set is used only once, in order to obtain the final experimental results reported here.

The rest of the data constitutes the *training* set, from which we sampled a subset of 1000 blocks as *development* set, for optimizing the log-linear model parameters used for decoding and re-scoring; the rest of the *training* set is used for estimating IBM translation models, constructing phrase tables and estimating the parameters of a target language model. In order to check the sensitivity of experimental results to the choice of the development set, we performed a run of preliminary experiments using different samples of 1000 blocks. The experimental results were nearly identical and highly consistent, showing that the choice of a particular development set has no influence on our conclusions. All experiments reported below are based on the same development set.

The composition of the various sets is detailed in Table 2. All data was tokenized and lowercased; all evaluations were performed independent of case. Note that the *validation* and *test* sets were originally made out of 1000 blocks sampled randomly from the data. These sets turned out to contain blocks identical to blocks from the training sets. Considering that these would normally have been handled by the translation memory component (see the HRSDC work-flow description in Section 3), we removed those blocks for which the source part was already found in the *training* set, hence their smaller sizes.

Main results

Our main experimental results are summarised in Table 3. There are sever 7.3.1) The top-level auxiliary file: submit.al interesting points to be made here. First, the results obtained on the test set with the two different versions of the RBMT system are in line with those reported in Table 1 for the overall *Job Bank* corpus: the domain-specific dictionary reduces the TER by about 9 points. We can also see that this translates as a 7-10 point increase in BLEU score.

Next, we can see that SMT performs surprisingly well, given that the size of the training corpus was relatively small (by SMT standards). PORTAGE does much better than the vanilla RBMT system (line 3 vs. line 1). The gain appears to be much larger on French-to-English data. This may be due to the fact that significantly more training data was available in that direction. However, it also seems that the RBMT system does slightly worse on French-to-English than on English-to-French. This is also evident when comparing the SMT system to the lexicon-enriched RBMT system. In fact, the domain-specific lexicon brings the RBMT system roughly to the same level as PORTAGE SMT on English-to-French, at least in terms of TER. BLEU is still slightly better for SMT, but as explained in section 2.1, the SMT system is trained to maximise BLEU.

Observe now the large gains obtained when we add our PORTAGE-based APE module to the vanilla RBMT system (T_1 + APE): in French-to-English translation the TER goes down by about 27% and the BLEU score goes up by almost 20 points! Interestingly, this combination turns out to be better than SMT alone: in English-to-French, the TER is 5 points lower than with PORTAGE SMT and the BLEU score about 4 points higher. Thus, our best MT results on the *Job Bank* data have been obtained neither by RBMT nor SMT but by a combination in which an RBMT "draft" is post-edited by an SMT module.

Finally, a comparison between the last two lines of Table 3 brings a rather clear answer to the question we raised in our introduction, that is, to what extent can APE be used as a means of customizing a vanilla RBMT system? The use of the statistical APE layer has essentially closed the 9% TER gap that we were observing between the vanilla and adapted RBMT systems, with a difference reduced to 0.5% in one direction and 1.3% in the other. Note that this effect has been obtained with a modestly-sized post-editing corpus (less than 500K words). As suggested by learning curves presented below, a larger corpus tends to close the gap even further.

This strongly suggests that the statistical APE layer is able to automatically extract from the corpus most of the useful information that was contained in the lexicon. Building a domain-specific lexicon can be a labour-intensive process. Consequently, the results presented here appear very promising because training an APE layer is essentially a fixed cost regardless of the underlying MT system.

Learning curves

In order to investigate the influence of the amount of training material on the above effect, we compute learning curves for the SMT system and both APE approaches, cf. Figure 1.

On all plots, the improved performance obtained by the APE approaches over standard SMT are obvious. However, as noted by Simard et al. (2007a), while the amount of data increases, the SMT system seems to improve faster than the APE module. Although we do not currently have enough data to verify it, we conjecture that given sufficient data, SMT will eventually take over. However, we now have reasons to believe that this will only happen with a massive amount of data (Simard et al., 2007b).

The relationship between the two APE approaches seems to depend on the translation direction. In French-to-English translation, the two systems yield a very similar performance when the full training set is used. This suggests that, in that direction, the APE is able to "learn" the effect of the lexicon using very little data. In English-to-French translation, the difference between the two APE ap-

Corpus	English-to-French					French-to-English				
		words:					words:			
	blocks	S	T_1	T_2	T_R	blocks	S	T_1	T_2	T_R
training	29577	321k	391k	403k	434k	37005	509k	525k	515k	468k
validation	893	10.3k	12.7k	13.0k	13.9k	974	13.6k	14.0k	13.7k	12.4k
test	907	9.7k	11.8k	12.1k	12.9k	966	13.5k	13.6k	13.4k	12.3k

Table 2: Data and split used in our experiments (in thousands of words). 'S' is the source-language text, 'T_1' is the output of the vanilla version of the RBMT system, 'T_2' is the output of the lexically adapted version of that system and 'T_R' is the final, manually post-edited text.

	English-to-French		French-to-English	
	TER	BLEU	TER	BLEU
T_1 (vanilla RBMT)	62.2	23.3	68.8	24.4
T_2 (customized RBMT)	53.5	32.9	59.3	31.2
PORTAGE SMT	53.7	36.0	43.9	41.0
T_1 + APE	**48.6**	**39.8**	**41.5**	**44.2**
T_2 + APE	**47.3**	**41.6**	**41.0**	**44.9**

Table 3: Experimental Results: For TER, lower (error) is better, while for BLEU, higher (score) is better. Results for Automatic Post-Editing are in bold.

proaches is slightly larger. However, even with just a small amount of training data, that difference is much smaller than the 10 BLEU points that separate the raw outputs of the vanilla and customized RBMT systems.

Notice also that, depending on the setting, it takes between 1000 and 8000 blocks of training material for the APE system to outperform the lexicon-enriched baseline MT. This is a very small amount of data in comparison with the corpora routinely handled in SMT.

5 Discussion

In light of the experimental results presented above, the main question this study poses is the following: Is customising a rule-based MT system a worthwhile effort? Even when it is limited to the preparation of a domain-specific dictionary, manual adaptation is very expensive. The English-to-French and French-to-English MT dictionaries developed by the *Job Bank* translators are still considered incomplete. Yet, they comprise a total of about 18 000 entries whose development necessitated an effort estimated to 18 person/month. SMT systems are fairly common and well understood, even if they are not yet as well represented on the commercial market. Our results suggest that using such systems in an Automatic Post-Editing layer can result in sizeable reduction of the human post-editing effort using a relatively modest quantity of already post-edited text.

It remains to be seen whether our results will carry over to other types of data and different domains. Preliminary results obtained on the WMT shared task seem encouraging (Simard et al., 2007b). A related issue is that we expect, based on the learning curves that we computed on our limited data, that when the size of the training data grows, the classical SMT approach will overtake the APE approach. Our interpretation of that effect is that as the data grows, the bottleneck of the APE approach is the baseline MT layer. While the SMT system keeps improving with additional training data, the APE layer may be limited by the shortcomings of the baseline MT output. Again, preliminary results suggest that this effect may not become critical until much larger data sizes. But, this suggests an extension of the APE approach where the source text is used as an auxiliary input to the APE layer, in order to bypass the possible limitations of the underlying baseline MT system. After all, human post-editors do sometimes look at the source language text, so why should APE's not do the same?

6 Conclusion

In this paper we have reported the results of a set of experiments about the use of phrase-based SMT technology for building an APE module for RBMT systems.

The *Job Bank* data that we used included source language texts together with two different RBMT translations, as well as a manually post-edited reference translation. One of two RBMT outputs had been produced by the vanilla version of the system while the other was also making use of manually-developed domain-specific dictionaries.

We then trained our PORTAGE SMT system to automatically "translate" (that is, post-edit) the output of either MT system into the language of reference translations. We also trained PORTAGE to directly translate from the source into the target language. We used the TER metric to evaluate the translations resulting from various system configurations. Our main findings were: 1) the TER of the RBMT system was 10% lower for the adapted version, but still above 50%; 2) despite the small size of the training set, French-to-English PORTAGE translations turned out to be significantly better than those of the adapted RBMT system; 3) however, significantly better results were achieved by combining the RBMT system with a PORTAGE-based APE module; and 4) finally, in combination with that APE module, the vanilla version of the RBMT system did almost

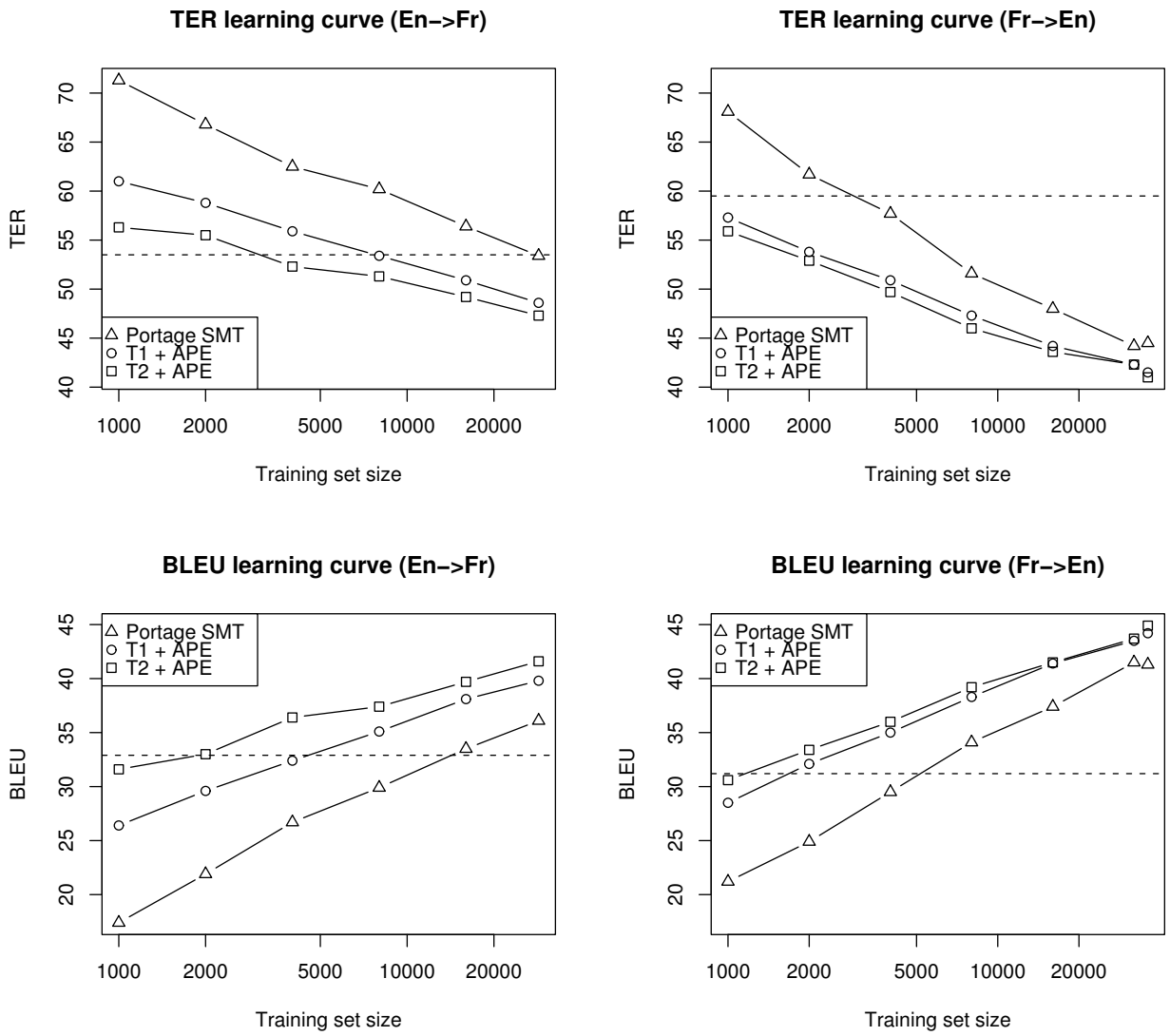

Figure 1: TER and BLEU scores as the amount of training data increases (log scale) for SMT alone (triangles) and both types of automatic post-editing: from vanilla MT (T_1 – circles) and from customized MT (T_2 – squares). The horizontal lines correspond to the performance of the customized MT without the APE layer. English-to-French is on the left and French-to-English on the right.

as well as the adapted version.

We conclude that an SMT-based post-editor appears to be an excellent way to improve the output of a vanilla RBMT system and constitutes a worthwhile alternative to costly manual adaptation efforts for such systems.

Acknowledgements

The work reported here was part of a collaboration between the National Research Council of Canada and the department of Human Resources and Social Development Canada. Special thanks go to Souad Benayyoub, Jean-Frédéric Hübsch, Nadia Molina and the rest of the *Job Bank* team at HRSDC for preparing data that was essential to this project.

References

Jeffrey Allen and Christofer Hogan. 2000. Toward the development of a post-editing module for Machine Translation raw output: a new productivity tool for processing controlled language. In *Third International Controlled Language Applications Workshop (CLAW2000)*, Washington, USA.

Peter F Brown, Stephen A Della Pietra, Vincent J Della Pietra, and Robert L Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311.

Jakob Elming. 2006. Transformation-based corrections of rule-based MT. In *Proceedings of the EAMT 11th Annual Conference*, Oslo, Norway.

George Foster, Roland Kuhn, and Howard Johnson. 2006. Phrasetable smoothing for statistical machine transla-

tion. In *Proceedings of EMNLP 2006*, pages 53–61, Sydney, Australia.

Richard Kittredge and John Lehrberger, editors. 1983. *Studies of Language in Restricted Domains*. Walter DeGruyter.

Kevin Knight and Ishwar Chander. 1994. Automated postediting of documents. In *Proceedings of National Conference on Artificial Intelligence*, pages 779–784, Seattle, USA.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. In *Proceedings of HLT-NAACL 2003*, pages 127–133, Edmonton, Canada.

Philipp Koehn. 2004. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. In *Proceedings of AMTA 2004*, pages 115–124, Washington, USA.

Daniel Marcu and William Wong. 2002. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of EMNLP 2002*, Philadelphia, USA.

Franz Josef Och. 2003. Minimum error rate training in Statistical Machine Translation. In *Proceedings of ACL-2003*, pages 160–167, Sapporo, Japan.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In ACL-02, editor, *Proc. ACL'02*.

Fatiha Sadat, Howard Johnson, Akakpo Agbago, George Foster, Roland Kuhn, Joel Martin, and Aaron Tikuisis. 2005. PORTAGE: A phrase-based machine translation system. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, pages 129–132, Ann Arbor, USA.

Michel Simard, Cyril Goutte, and Pierre Isabelle. 2007a. Statistical phrase-based post-editing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 508–515, Rochester, USA, April. Association for Computational Linguistics.

Michel Simard, Nicola Ueffing, Pierre Isabelle, and Roland Kuhn. 2007b. Rule-based translation with statistical phrase-based post-editing. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 203–206, Prague, Czech Republic, June. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of AMTA-2006*, Cambridge, USA.