Lessons from NRC's Portage System at WMT 2010

Samuel Larkin, Boxing Chen, George Foster, Ulrich Germann, Eric Joanis, Howard Johnson, and Roland Kuhn
National Research Council of Canada (NRC)
Gatineau, Québec, Canada.
`Firstname.Lastname@cnrc-nrc.gc.ca`

Abstract

NRC's Portage system participated in the English-French (E-F) and French-English (F-E) translation tasks of the ACL WMT 2010 evaluation. The most notable improvement over earlier versions of Portage is an efficient implementation of lattice MERT. While Portage has typically performed well in Chinese to English MT evaluations, most recently in the NIST09 evaluation, our participation in WMT 2010 revealed some interesting differences between Chinese-English and E-F/F-E translation, and alerted us to certain weak spots in our system. Most of this paper discusses the problems we found in our system and ways of fixing them. We learned several lessons that we think will be of general interest.

1 Introduction

Portage, the statistical machine translation system of the National Research Council of Canada (NRC), is a two-pass phrase-based system. The translation tasks to which it is most often applied are Chinese to English, English to French (henceforth "E-F"), and French to English (henceforth "F-E"): in recent years we worked on Chinese-English translation for the GALE project and for NIST evaluations, and English and French are Canada's two official languages. In WMT 2010, Portage scored 28.5 BLEU (uncased) for F-E, but only 27.0 BLEU (uncased) for E-F. For both language pairs, Portage truecasing caused a loss of 1.4 BLEU; other WMT systems typically lost around 1.0 BLEU after truecasing. In Canada, about 80% of translations between English and French are from English to French, so we would have preferred better results for that direction. This paper first describes the version of Portage that participated in WMT 2010. It then analyzes problems with the system and describes the solutions we found for some of them.

2 Portage system description

2.1 Core engine and training data

The NRC system uses a standard two-pass phrase-based approach. Major features in the first-pass loglinear model include phrase tables derived from symmetrized IBM2 alignments and symmetrized HMM alignments, a distance-based distortion model, a lexicalized distortion model, and language models (LMs) that can be either static or else dynamic mixtures. Each phrase table used was a merged one, created by separately training an IBM2-based and an HMM-based joint count table on the same data and then adding the counts. Each includes relative frequency estimates and lexical estimates (based on Zens and Ney, 2004) of forward and backward conditional probabilities. The lexicalized distortion probabilities are also obtained by adding IBM2 and HMM counts. They involve 6 features (monotone, swap and discontinuous features for following and preceding phrase) and are conditioned on phrase pairs in a model similar to that of Moses (Koehn *et al.*, 2005); a MAP-based backoff smoothing scheme is used to combat data sparseness when estimating these probabilities. Dynamic mixture LMs are linear mixtures of ngram models trained on parallel sub-corpora with weights set to minimize perplexity of the current source text as described in (Foster and Kuhn, 2007); henceforth, we'll call them "dynamic LMs".

Decoding uses the cube-pruning algorithm of (Huang and Chiang, 2007) with a 7-word distortion limit. Contrary to the usual implementation of distortion limits, we allow a new phrase to end

Proceedings of the Joint 5th Workshop on Statistical Machine Translation and MetricsMATR, pages 133–138,
Uppsala, Sweden, 11-16 July 2010.

more than 7 words past the first non-covered word, as long as the new phrase starts within 7 words from the first non-covered word. Notwithstanding the distortion limit, contiguous phrases can always be swapped. Out-of-vocabulary (OOV) source words are passed through unchanged to the target. Loglinear weights are tuned with Och's max-BLEU algorithm over lattices (Macherey *et al.*, 2008); more details about lattice MERT are given in the next section. The second pass rescores 1000-best lists produced by the first pass, with additional features including various LM and IBM-model probabilities; ngram, length, and reordering posterior probabilities and frequencies; and quote and parenthesis mismatch indicators. To improve the quality of the maxima found by MERT when using large sets of partially-overlapping rescoring features, we use greedy feature selection, first expanding from a baseline set, then pruning.

We restricted our training data to data that was directly available through the workshop's website; we didn't use the LDC resources mentioned on the website (*e.g.*, French Gigaword, English Gigaword). Below, "mono" refers to all monolingual data (Europarl, news-commentary, and shuffle); "mono" English is roughly three times bigger than "mono" French (50.6 M lines in "mono" English, 17.7 M lines in "mono" French). "Domain" refers to all WMT parallel training data except GigaFrEn (*i.e.*, Europarl, news-commentary, and UN).

2.2 Preprocessing and postprocessing

We used our own English and French pre- and post-processing tools, rather than those available from the WMT web site. For training, all English and French text is tokenized with a language-specific tokenizer and then mapped to lowercase. Truecasing uses an HMM approach, with lexical probabilities derived from "mono" and transition probabilities from a 3-gram LM trained on truecase "mono". A subsequent rule-based pass capitalizes sentence-initial words. A final detokenization step undoes the tokenization.

2.3 System configurations for WMT 2010

In the weeks preceding the evaluation, we tried several ways of arranging the resources available to us. We picked the configurations that gave the highest BLEU scores on WMT2009 Newstest. We found that tuning with lattice MERT rather than N-best MERT allowed us to employ more parameters and obtain better results.

E-F system components:

1. Phrase table trained on "domain";
2. Phrase table trained on GigaFrEn;
3. Lexicalized distortion model trained on "domain";
4. Distance-based distortion model;
5. 5-gram French LM trained on "mono";
6. 4-gram LM trained on French half of GigaFrEn;
7. Dynamic LM composed of 4 LMs, each trained on the French half of a parallel corpus (5-gram LM trained on "domain", 4-gram LM on GigaFrEn, 5-gram LM on news-commentary and 5-gram LM on UN).

The F-E system is a mirror image of the E-F system.

3 Details of lattice MERT (LMERT)

Our system's implementation of LMERT (Macherey *et al.*, 2008) is the most notable recent change in our system. As more and more features are included in the loglinear model, especially if they are correlated, N-best MERT (Och, 2003) shows more and more instability, because of convergence to local optima (Foster and Kuhn, 2009). We had been looking for methods that promise more stability and better convergence. LMERT seemed to fit the bill. It optimizes over the complete lattice of candidate translations after a decoding run. This avoids some of the problems of N-best lists, which lack variety, leading to poor local optima and the need for many decoder runs.

Though the algorithm is straightforward and is highly parallelizable, attention must be paid to space and time resource issues during implementation. Lattices output by our decoder were large and needed to be shrunk dramatically for the algorithm to function well. Fortunately, this could be achieved via the finite state equivalence algorithm for minimizing deterministic finite state machines. The second helpful idea was to separate out the features that were a function of the phrase associated with an arc (*e.g.*, translation length and translation model probability features). These features could then be stored in a smaller phrase-feature table. Features associated with language or distortion models could be handled in a larger transition-feature table.

The above ideas, plus careful coding of data structures, brought the memory footprint down sufficiently to allow us to use complete lattices from the decoder and optimize over the complete development set for NIST09 Chinese-English. However, combining lattices between decoder runs again resulted in excessive memory requirements. We achieved acceptable performance by searching only the lattice from the latest decoder run; perhaps information from earlier runs, though critical for convergence in N-best MERT, isn't as important for LMERT.

Until a reviewer suggested it, we had not thought of pruning lattices to a specified graph density as a solution for our memory problems. This is referred to in a single sentence in (Macherey *et al.*, 2008), which does not specify its implementation or its impact on performance, and is an option of OpenFst (we didn't use OpenFst). We will certainly experiment with lattice pruning in future.

Powell's algorithm (PA), which is at the core of MERT, has good convergence when features are mostly independent and do not depart much from a simple coordinate search; it can run into problems when there are many correlated features (as with multiple translation and language models). **Figure 1** shows the kind of case where PA works well. The contours of the function being optimized are relatively smooth, facilitating learning of new search directions from gradients.

Figure 2 shows a more difficult case: there is a single optimum, but noise dominates and PA has difficulty finding new directions. Search often iterates over the original co-ordinates, missing optima that are nearby but in directions not discoverable from local gradients. Probes in random directions can do better than iteration over the same directions (this is similar to the method proposed for N-best MERT by Cer *et al.*, 2008). Each 1-dimensional MERT optimization is exact, so if our probe stabs a region with better scores, it will be discovered. **Figures 1** and **2** only hint at the problem: in reality, 2-dimensional search isn't a problem. The difficulties occur as the dimension grows: in high dimensions, it is more important to get good directions and they are harder to find.

For WMT 2010, we crafted a compromise with the best properties of PA, yet allowing for a more aggressive search in more directions. We start with PA. As long as PA is adding new direction vectors, it is continued. When PA stops adding new directions, random rotation (orthogonal transformation) of the coordinates is performed and PA is restarted in the new space. PA almost always fails to introduce new directions within the new coordinates, then fails again, so another set of random coordinates is chosen. This process repeats until convergence. In future work, we will look at incorporating random restarts into the algorithm as additional insurance against premature convergence.

Our LMERT implementation has room for improvement: it may still run into over-fitting problems with many correlated features. However, during preparation for the evaluation, we noticed that LMERT converged better than N-best MERT, allowing models with more features and higher BLEU to be chosen.

After the WMT submission, we discovered that our LMERT implementation had a bug; our submission was tuned with this buggy LMERT. Comparison between our E-F submission tuned with N-best MERT and the same system tuned with bug-fixed LMERT shows BLEU gains of +1.5-3.5 for LMERT (on dev, WMT2009, and WMT2010, with no rescoring). However, N-best MERT performed very poorly in this particular case; we usually obtain a gain due to LMERT of +0.2-1.0 (*e.g.*, for the submitted F-E system).

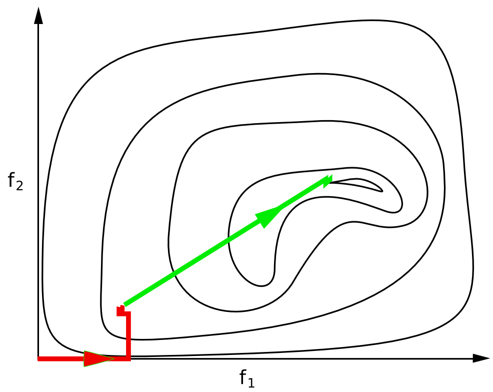

Figure 1: Convergence for PA (Smooth Feature Space)

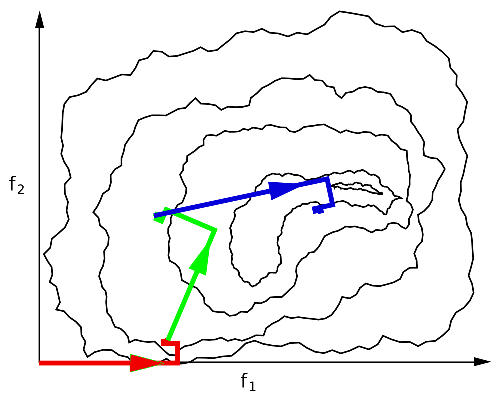

Figure 2: Convergence for PA with Random Rotation (Rough Feature Space)

4 Problems and Solutions

4.1 Fixing LMERT

Just after the evaluation, we noticed a discrepancy for E-F between BLEU scores computed during LMERT optimization and scores from the 1-best list immediately after decoding. Our LMERT code had a bug that garbled any accented word in the version of the French reference in memory; previous LMERT experiments had English as target language, so the bug hadn't showed up. The bug didn't affect characters in the 7-bit ASCII set, such as English ones, only accented characters. Words in candidate translations were not garbled, so correct translations with accents received a lower BLEU score than they should have. As **Table 1** shows, this bug cost us about 0.5 BLEU for WMT 2010 E-F after rescoring (according to NRC's internal version of BLEU, which differs slightly from WMT's BLEU). Despite this bug, the system tuned with buggy LMERT (and submitted) was still better than the best system we obtained with N-best MERT. The bug didn't affect F-E scores.

	Dev	WMT2009	WMT2010
LMERT (bug)	25.26	26.85	27.55
LMERT (no bug)	25.43	26.89	28.07

Table 1: LMERT bug fix (E-F BLEU after rescoring)

4.2 Fixing odd translations

After the evaluation, we carefully studied the system outputs on the WMT 2010 test data, particularly for E-F. Apart from truecasing errors, we noticed two kinds of bad behaviour: translations of proper names and apparent passthrough of English words to the French side.

Examples of E-F translations of proper names from our WMT 2010 submission (each from a different sentence):

Mr. Onderka → M. Roman, Lukáš Marvan → G. Lukáš, Janey → The, Janette Tozer → Janette, Aysel Tugluk → joints tugluk, Tawa Hallae → Ottawa, Oleson → production, Alcobendas → ;

When the LMERT bug was fixed, some but not all of these bad translations were corrected (*e.g.*, 3 of the 8 examples above were corrected).

Our system passes OOV words through unchanged. Thus, the names above aren't OOVs, but words that occur rarely in the training data, and for which bad alignments have a disproportionate effect. We realized that when a source word begins with a capital, that may be a signal that it should be passed through. We thus designed a passthrough feature function that applies to all capitalized forms not at the start of a sentence (and also to forms at the sentence start if they're capitalized elsewhere). Sequences of one or more capitalized forms are grouped into a phrase suggestion (*e.g.*, Barack Obama → barrack obama) which competes with phrase table entries and is assigned a weight by MERT.

The passthrough feature function yields a tiny improvement over the E-F system with the bug-fixed LMERT on the dev corpus (WMT2008): +0.06 BLEU (without rescoring). It yields a larger improvement on our test corpus: +0.27 BLEU (without rescoring). Furthermore, it corrects all the examples from the WMT 2010 test shown above (after the LMERT bug fix 5 of the 8 examples above still had problems, but when the passthrough function is incorporated all of them go away). Though the BLEU gain is small, we are happy to have almost eradicated this type of error, which human beings find very annoying.

The opposite type of error is apparent passthrough. For instance, "we're" appeared 12 times in the WMT 2010 test data, and was translated 6 times into French as "we're" - even though better translations had higher forward probabilities. The source of the problem is the backward probability P(E="we're"|F="we're"), which is 1.0; the backward probabilities for valid French translations of "we're" are lower. Because of the high probability P(E="we're"|F="we're") within the loglinear combination, the decoder often chooses "we're" as the French translation of "we're".

The (E="we're", F="we're") pair in WMT 2010 phrase tables arose from two sentence pairs where the "French" translation of an English sentence is a copy of that English sentence. In both, the original English sentence contains "we're". Naturally, the English words on the "French" side are word-aligned with their identical twins on the English side. Generally, if the training data has sentence pairs where the "French" sentence contains words from the English sentence, those words will get high backward probabilities of being translated as themselves. This problem may not show up as an apparent passthrough; instead, it may cause MERT to lower the weight of the backward probability component, thus hurting performance.

We estimated English contamination of the French side of the parallel training data by manually inspecting a random sample of "French" sentences containing common English function words. Manual inspection is needed for accurate estimation: a legitimate French sentence might contain mostly English words if, *e.g.*, it is short and cites the title of an English work (this wouldn't count as contamination). The degree of contamination is roughly 0.05% for Europarl, 0.5% for news-commentary, 0.5% for UN, and 1% for GigaFrEn (in these corpora the French is also contaminated by other languages, particularly German). Foreign contamination of English for these corpora appears to be much less frequent.

Contamination can take strange forms. We expected to see English sentences copied over intact to the French side, and we did, but we did not expect to see so many "French" sentences that interleaved short English word sequences with short French word sequences, apparently because text with an English and a French column had been copied by taking lines from alternate columns. We found many of these interleaved "French" sentences, and found some of them in exactly this form on the Web (*i.e.*, the corruption didn't occur during WMT data collection). The details may not matter: whenever the "French" training sentence contains words from its English twin, there can be serious damage via backward probabilities.

To test this hypothesis, we filtered all parallel and monolingual training data for the E-F system with a language guessing tool called text_cat (Cavnar and Trenkle, 1994). From parallel data, we filtered out sentence pairs whose French side had a high probability of not being French; from LM training data, sentences with a high non-French probability. We set the filtering level by inspecting the guesser's assessment of news-commentary sentences, choosing a rather aggressive level that eliminated 0.7% of news-commentary sentence pairs. We used the same level to filter Europarl (0.8% of sentence pairs removed), UN (3.4%), GigaFrEn (4.7%), and "mono" (4.3% of sentences).

	Dev	WMT2009	WMT2010
Baseline	25.23	26.47	27.72
Filtered	25.45	26.66	27.98

Table 2: Data filtering (E-F BLEU, no rescoring)

Table 2 shows the results: a small but consistent gain (about +0.2 BLEU without rescoring). We have not yet confirmed the hypothesis that copies of source-language words in the paired target sentence within training data can damage system performance via backward probabilities.

4.3 Fixing problems with LM training

Post-evaluation, we realized that our arrangement of the training data for the LMs for both language directions was flawed. The grouping together of disparate corpora in "mono" and "domain" didn't allow higher-quality, truly in-domain corpora to be weighted more heavily (*e.g.*, the news corpora should have higher weights than Europarl, but they are lumped together in "mono"). There are also potentially harmful overlaps between LMs (*e.g.*, GigaFrEn is used both inside and outside the dynamic LM).

We trained a new set of French LMs for the E-F system, which replaced all the French LMs (#5-7) described in section 2.3 in the E-F system:

1. 5-gram LM trained on news-commentary and shuffle;
2. Dynamic LM based on 4 5-gram LMs trained on French side of parallel data (LM trained on GigaFrEn, LM on UN, LM on Europarl, and LM on news-commentary).

We did not apply the passthrough function or language filtering (section 4.2) to any of the training data for any component (LMs, TMs, distortion models) of this system; we did use the bug-fixed version of LMERT (section 4.1).

The experiments with these new French LMs for the E-F system yielded a small decrease of NRC BLEU on dev (-0.15) and small increases on WMT Newstest 2009 and Newstest 2010 (+0.2 and +0.4 respectively without rescoring). We didn't do F-E experiments of this type.

4.4 Pooling improvements

The improvements above were (individual uncased E-F BLEU gains without rescoring in brackets): LMERT bug fix (about +0.5); passthrough feature function (+0.1-0.3); language filtering for French (+0.2). There was also a small gain on test data by rearranging E-F LM training data, though the loss on "dev" suggests this may be a statistical fluctuation. We built these four improvements into the evaluation E-F system, along with quote normalization: in all training and test data, diverse single quotes were mapped onto the ascii single quote, and diverse double quotes were mapped onto the ascii double quote. The average result on WMT2009 and WMT2010 was +1.7 BLEU points compared to the original system, so there may be synergy between the improvements. The original system had gained +0.3 from rescoring, while the final improved system only gained +0.1 from rescoring: a post-evaluation rescored gain of +1.5.

An experiment in which we dropped lexicalized distortion from the improved system showed that this component yields about +0.2 BLEU. Much earlier, when we were still training systems with N-best MERT, incorporation of the 6-feature lexicalized distortion often caused scores to go down (by as much as 2.8 BLEU). This illustrates how LMERT can make incorporation of many more features worthwhile.

4.5 Fixing truecasing

Our truecaser doesn't work as well as truecasers of other WMT groups: we lost 1.4 BLEU by truecasing in both language directions, while others lost 1.0 or less. To improve our truecaser, we tried: 1. Training it on all relevant data and 2. Collecting 3-gram case-pattern statistics instead of unigrams. Neither of these helped significantly. One way of improving the truecaser would be to let case information from source words influence the case of the corresponding target words. Alternatively, one of the reviewers stated that several labs involved in WMT have no separate truecaser and simply train on truecase text. We had previously tried this approach for NIST Chinese-English and discarded it because of its poor performance. We are currently re-trying it on WMT data; if it works better than having a separate truecaser, this was yet another area where lessons from Chinese-English were misleading.

5 Lessons

LMERT is an improvement over N-best MERT. The submitted system was one for which N-best MERT happened to work very badly, so we got ridiculously large gains of +1.5-3.5 BLEU for non-buggy LMERT over N-best MERT. These results are outliers: in experiments with similar configurations, we typically get +0.2-1.0 for LMERT over N-best MERT. Post-evaluation, four minor improvements – a case-based passthrough function, language filtering, LM rearrangement, and quote normalization – collectively gave a nice improvement. Nothing we tried helped truecaser performance significantly, though we have some ideas on how to proceed.

We learned some lessons from WMT 2010.

Always test your system on the relevant language pair. Our original version of LMERT was developed on Chinese-English and worked well there, but had a bug that surfaced only when the target language had accents.

European language pairs are more porous to information than Chinese-English. Our WMT system reflected design decisions for Chinese-English, and thus didn't exploit case information in the source: it passed through OOVs to the target, but didn't pass through upper-case words that are likely to be proper nouns.

It is beneficial to remove foreign-language contamination from the training data.

When entering an evaluation one hasn't participated in for several years, always read system papers from the previous year. Some of the WMT 2008 system papers mention passthrough of some non-OOVs, filtering out of noisy training data, and using the case of a source word to predict the case of the corresponding target word.

References

William Cavnar and John Trenkle. 1994. N-Gram-Based Text Categorization. *Proc. Symposium on Document Analysis and Information Retrieval*, UNLV Publications/Reprographics, pp. 161-175.

Daniel Cer, Daniel Jurafsky, and Christopher D. Manning. 2008. Regularization and search for minimum error rate training. *Proc. Workshop on SMT,* pp. 26-34.

George Foster and Roland Kuhn. 2009. Stabilizing Minimum Error Rate Training. *Proc. Workshop on SMT*, pp. 242-249.

George Foster and Roland Kuhn. 2007. Mixture-Model Adaptation for SMT. *Proc. Workshop on SMT*, pp. 128-135.

Liang Huang and David Chiang. 2007. Forest Rescoring: Faster Decoding with Integrated Language Models. *Proc. ACL*, pp. 144-151.

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh System Description for the 2005 IWSLT Speech Transcription Evaluation. *MT Eval. Workshop.*

Wolfgang Macherey, Franz Josef Och, Ignacio Thayer, and Jakob Uszkoreit. 2008. Lattice-based Minimum Error Rate Training for Statistical Machine-Translation. *Conf. EMNLP*, pp. 725-734.

Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. *Proc. ACL*, pp. 160-167.

Richard Zens and Hermann Ney. 2004. Improvements in Phrase-Based Statistical Machine Translation. *Proc. HLT/NAACL*, pp. 257-264.