PORTAGE Phrase-Based System for Chinese-to-English Translation

Roland Kuhn, George Foster, Samuel Larkin, and Nicola Ueffing

Institute for Information Technology, National Research Council of Canada (NRC Canada)
Gatineau, Québec, CANADA
Email: {Roland.Kuhn, George.Foster, Samuel.Larkin, Nicola.Ueffing}@cnrc-nrc.gc.ca

Abstract

This paper describes the participation of the machine translation team at NRC Canada in the Mandarin-to-English open evaluation task of the 2006 TC-STAR Workshop on Speech-to-Speech Translation. We describe PORTAGE, a statistical phrase-based machine translation system, and discuss the experimental results two variants of the system obtained in the evaluation. Both variants of the system used hand-coded rules to translate numbers and dates, along with a small number of Chinese names (those encountered in the dev corpus). The secondary variant of PORTAGE carried out Good-Turing smoothing of the phrase tables. Both variants, especially the primary one, performed extremely well on the two metrics related to weighted N-gram models (WNM); we suspect this is due to the incorporation of rules for translating Chinese names.

1. Introduction

PORTAGE is a research and development system that carries out statistical machine translation (SMT); it has been under development at the National Research Council of Canada (NRC Canada) since September 2004. This system was evaluated in the framework of the Chinese to English SLT "verbatim" track of the 2006 TC-STAR Workshop on Speech-to-Speech Translation. This track evaluates translation performance on transcriptions of Mandarin speech produced by ELDA; the transcriptions include spontaneous speech phenomena such as hesitations, corrections, *etc*.

The NRC group does not currently have a version of PORTAGE that is designed specifically for input derived from speech. Though the primary variant of PORTAGE tested in the evaluation had parameter weights optimized on the development corpus for TC-STAR's "verbatim" track, the two variants of PORTAGE that participated in TC-STAR had roughly the same design as those that participate in text-only tasks, such as the NAACL WMT06 Workshop (Johnson *et al.*, 2006). Both systems employed rule-based translation of numbers and dates, and a tiny dictionary of named entities (82 entities) based on names seen in the development corpus. The secondary variant employed Good-Turing smoothing of the phrase tables, which will be discussed in more detail below.

2. Basic Structure of PORTAGE

PORTAGE was described in detail in (Sadat *et al.*, 2005). The system operates in four main phases: **preprocessing** of raw data into tokens, with translation suggestions for some words or phrases being generated by rules; **decoding** to produce one or more translation hypotheses; error-driven **rescoring** to choose the best final hypothesis; and **postprocessing** to enforce appropriate case and punctuation.

2.1 Preprocessing

In Chinese texts, the characters are typically presented continuously, with only occasional punctuation marks dividing them. The first step of PORTAGE's Chinese preprocessing is to tokenize the input using the LDC segmenter: that is, to insert white spaces between groups of characters to create Chinese "words" analogous to those in European languages. The size of the Chinese vocabulary of the version of PORTAGE evaluated in TC-STAR was 49,235 "words" made up purely of Chinese characters. If words containing other symbols (*e.g.*, digits, letters from the Latin alphabet, *etc*.) are included, the Chinese vocabulary of the system is much larger: 516,816 "words".

In the next step of preprocessing, numbers and dates are translated from Chinese to English using hand-coded rules. These rules are not obligatory; the weight optimization process for the decoder (see next subsection, **2.1**) decides how strongly they should be enforced. Thus, in some cases the decoder may choose to override the rules and use a translation based on phrase translations stored in the phrase table instead. (A hypothetical example might be a case where the decoder chooses to translate the Chinese equivalent of "December 25" to "Christmas" – based on that translation being stored in a phrase table – rather than as "December 25" in English.) In this step, we also remove some Chinese interjections in the transcriptions that do not assist translation.

Finally, a small number of named entities are also translated. Just before the evaluation, we managed to find a Chinese informant who was able to provide translations for some of the named entities found in the TC-STAR development data. We informally tested two ways of incorporating these translations: adding them to an existing phrase table containing Chinese/English dictionary entries, and incorporating them as rules. The rule-based approach gave us more flexibility to control the expression of this valuable domain-specific knowledge, so we used it for both variants of PORTAGE that participated in the evaluation. In a later section, we analyze the impact of including these translations of named entities on PORTAGE's BLEU score for the TC-STAR translation task. These experiments showed that including the named-entity rules yielded a BLEU score that was approximately 0.5% higher (from about 12.4

BLEU to 12.9 BLEU). These rules also affected the two weighted N-gram (WNM) scores, though the results here are harder to interpret.

Table 1 shows the non-Chinese names found in the Chinese development corpus that are included in this phrase table. Note the ethnic and geographic diversity of these names, and also how the choice of names reflects news events at a given time (the list is full of personalities from the Clinton era, and reflects the events of the civil war in Yugoslavia). **Table 2** shows the Chinese names in the same phrase table. One Chinese name has two English translations (Chao Tzu-yang, Zhao Ziyang) and two Chinese names (Qin Yongmin and Wei Ke) each have two different ideogrammatic "spellings" in Chinese. Note also the inconsistency in English punctuation for Chinese names, which might well reduce scores for automatic measures of MT quality like BLEU: *e.g.*, "Jin YinQin" might also have been spelled "Jin Yinqin", "Jin Yin-qin", or "Jin Yin Qin". All these considerations show the extreme difficulty of designing a good module for translating named entities from Chinese to English. Nevertheless, in future work on the Chinese-to-English system, we plan to invest more effort in named entity recognition and translation.

Albright	Angola	Annan	Anwar	Arafat	Ashrawi
Aziz	Belgrade	Bob	Bosnia	Butler	Clinton
Conte	Dayton	Galster	Hanan Ashrawi	Herze-govina	Holbrooke
Hun Sen	Ivanov	Jimmy Carter	Keizo Obuchi	Kissinger	Kurd
Lansana Conte	Lewinsky	Lieber-man	Lord	Mama-douba	McClary
Milosevic	Netan-yahu	Ocalan	Omar	Pinochet	Ross
Schwarz	Serushago	Solana	Tate	Vance	Westen-dorp

Table 1: Non-Chinese Names Translated From Chinese

Chao Tzu-yang = Zhao Ziyang	Chen Zhonghe	Chen Zhong-xin	Deng Xiao-ping	Dong Fang	Guo Zheng-liang
He Xintong	He Zong-an	Hi Jiangxia	Hu Jintao	Hu Yaobang	Hua Guofeng
Huang Dashu	Jiang Tian	Jiang Zemin	Jin YinQin	Lee Teng-hui	Li Zhaoxing
Lian Zhan	Ma Kaiyue	Ma Ying-jeou	Qi Yong-min	Qin Qingguo	Qin Yongmin (2)
Shen Honghui	Ti Na	Ti Najin	Wan Li	Wang Ce	Wang Wenjiang
Wang Yiru	Wang Youcai	Wei Ke (2)	Wu Zuodong	Xiao Ti Na	Xu Guang
Xu Wenli	Zhang Jianlong	Zhao Wancheng	Zhu Rongji		

Table 2: Chinese Names Translated From Chinese

2.2 Decoding

Decoding is the central phase in statistical machine translation (SMT), involving a search for the word sequence hypotheses $\boldsymbol{T}$ that have the highest probabilities of being translations of the source sentence $\boldsymbol{S}$ according to a model for $P(\boldsymbol{T}|\boldsymbol{S})$. The PORTAGE decoder's model for $P(\boldsymbol{T}|\boldsymbol{S})$ is a loglinear model that incorporates the following features: phrase translation models in the $P(\boldsymbol{S}|\boldsymbol{T})$ direction, language models, a distortion penalty, and a word penalty. The phrase models are learned from word-aligned parallel corpora using the "diag-and" method described in (Koehn *et al.*, 2003), and the language models are trained using the SRILM toolkit (Stolcke 2002) with Kneser-Ney smoothing. Loglinear weights are set to maximize BLEU score, using Och's algorithm on a development corpus (Och 2003). As mentioned above, this algorithm also implicitly determines the weights on the rules for translating numbers, dates, and domain-specific named entities. These rules generate entries which are inserted into all phrase tables with a specified probability. In all experiments we report below, this probability was set to 1. This essentially means that rule-based translations will override automatically learned translations unless there is very strong counter-evidence from the language model. The ability of the language model to select alternative translations depends on its weight relative to the translation model as assigned by Och's algorithm.

For each preprocessed source sentence, N best translations are identified using Viterbi beam search with a loglinear model.

2.3 Rescoring

The N best hypotheses generated by the decoder can be rescored. Our secondary submission for TC-STAR carried out rescoring on 1000-best lists, using a loglinear model with the same features as used by the decoder, plus IBM model probabilities in both directions ($P(\boldsymbol{T}|\boldsymbol{S})$ and $P(\boldsymbol{S}|\boldsymbol{T})$), plus IBM-based features to detect untranslated words in both directions. Och's algorithm was used to learn loglinear weights, with maximum BLEU as the learning criterion.

2.4 Postprocessing

Raw English output is truecased using the method described in (Agbago *et al.*, 2005). The method uses a combination of statistical components, including an N-gram language model, a case mapping model, and a specialized language model for unknown words. After the output has been truecased, it is detokenized using simple heuristics.

3 Phrase Table Smoothing

It is surprising how little **systematic** attention has been paid to phrase table smoothing in the SMT literature. That is, although it is practiced by several SMT groups, and is often mentioned casually in SMT papers whose main subject is something else, there does not (to our knowledge) exist a published paper describing detailed experiments in which various phrase table smoothing techniques are compared. By contrast, surveys of the statistical language modeling literature such as (Chen and

Goodman 1999; Goodman 2001) deal extensively with techniques for smoothing language models.

Smoothing is typically applied when training data is too sparse to estimate accurately the parameters of a statistical model. For instance, given a trigram language model for a language with a vocabulary of 100,000 = 10^5 words, there are $(10^5)^3 = 10^{15}$ parameters to be estimated. Even for gigantic training corpora, this is impractical. Thus, trigram distributions estimated from the training data are smoothed with lower-order distributions, such as bigram and unigram distributions.

In the case of phrase-based SMT systems, two core components – the phrase translation models P(***t***|***s***) and P(***s***|***t***), both contained in data structures called "phrase tables", where "phrases" ***s*** and ***t*** are contiguous sequences of words in the source and target language respectively – are estimated on the basis of extraordinarily sparse data. The maximum phrase length permitted in a typical SMT system might be on the order of eight words for both languages. If the vocabulary for each language consists (again) of roughly 100,000 words, accurately estimating the frequency of co-occurrence of all possible eight-word source-language sequences with all possible eight-word target-language sequences would require estimation of roughly $(10^5)^8 * (10^5)^8 = 10^{80}$ parameters. Thus, phrase tables are estimated on the basis of data that are even more sparse, compared to the number of parameters they represent, than the data used to train N-gram language models. The logical conclusion is that it is even more important to smooth phrase tables correctly than it is to smooth N-gram language models.

Recently, we have begun to compare phrase table smoothing techniques; some of this work is described in (Johnson *et al*., 2006). Consider P(***s***|***t***); typically, this is estimated by dividing the number of times ***s*** and ***t*** were observed to be aligned by the number of times ***t*** was observed (the relative frequency estimate). We distinguish between "glass box" and "black box" techniques for smoothing such relative frequency estimates.

Glass box techniques break phrases down into their component words: for instance, if phrase ***s*** is made up of words s_1 through s_n ($s = s_1 \ldots s_n$) then one might estimate P(***s***|***t***) through some sort of combination of the information found in P(s_1|***t***), …, P(s_n|***t***). The "noisy-or" technique described in (Zens and Ney, 2004) and the technique described in (Koehn *et al*., 2005) differ, but they are both "glass box" techniques, because they both smooth relative frequency estimates by using estimates based on individual words in phrase ***s***.

By contrast, "black box" smoothing techniques directly manipulate the count of co-occurrences of ***s*** and ***t***, without decomposing these phrases: the phrases are treated as black boxes which cannot be analyzed. In practice, this involves shrinking counts of infrequent phrase co-occurrences, on the grounds that they are less reliable. Such observations with low counts represent a tiny sample from the large population of possible events with low probability, most of which were not observed in the training data. To leave some probability mass in the model available for such unobserved events, it is necessary to subtract probability mass from observed events (with low-count observations giving up proportionately more probability mass than high-count observations). "Black box" smoothing is less sophisticated than "glass box" smoothing, but has the advantage of being very easy to implement: one simply applies the appropriate formula to a set of phrase co-occurrence counts to generate modified counts, then produces a new phrase table from the modified counts.

Another nice feature of "black box" smoothing of phrase tables is that it includes techniques that are directly analogous to techniques used for smoothing N-gram language models. In (Johnson *et al*., 2006) we tried several of these, including Good-Turing smoothing and Kneser-Ney smoothing.

For the TC-STAR experiments, we applied Good-Turing phrase table smoothing. Good-Turing smoothing (Church and Gale, 1991) modifies observed counts **c** according to the following formula:

$$\mathbf{c_g = (c + 1)*n_{c+1} / n_c}$$

where $\mathbf{c_g}$ is the modified count value used to replace **c** in subsequent relative frequency estimates, and $\mathbf{n_c}$ is the number of events having count **c**. When applied to phrase pair counts, this means, for instance, that all counts of 1 are replaced by a new value of $2*\mathbf{n_2/n_1}$, where $\mathbf{n_2}$ is the number of doubleton pairs observed, and $\mathbf{n_1}$ is the number of singleton pairs observed. Since $\mathbf{n_2/n_1}$ is considerably less than 0.5 – it is 0.088 in our largest phrase table, containing 25M entries - this results in all counts of 1 being reduced to 0.176.

The observant reader will have noticed there is a potential problem here for high counts: when estimating the modified value for c = 347,623 one may have zero count for c = 347,624; according to the formula, this should yield a modified count of 0 for c = 347,623 – highly undesirable! A form of least squares fitting is applied to handle this problem, as suggested in (Gale and Church, 1991).

4 Experiments

4.1 Data

The training data used for both our primary and secondary submissions are shown in **Table 3**. For size reasons, we divided the parallel training material into two portions: the UN corpus and all others (except the Xinhua corpus, which hurt performance). We trained one phrase table on each of these two corpora, and one language model on each of their English halves. One additional phrase table was derived from a merge of the supplied Chinese-English lexicon and named-entity list. One additional language model was trained on the Xinhua and CNA subsets of the English Gigaword corpus.

CORPUS	USE	SENTENCES
Non-UN parallel	Phrasetable 1	3,164,180
UN parallel	Phrasetable 2	4,979,345
Lexicon + NE list	Phrasetable 3	*1,155,405
Non-UN English	LM 1	3,164,180
UN English	LM 2	4,979,345
Gigaword English	LM 3	11,681,852

Table 3: Training corpora. (* Indicates the number of entries in the table rather than number of sentence pairs.)

The development and test corpora are shown in **Table 4**. We divided the supplied development set into three parts (corresponding to documents in chronological order), and used the first two for tuning loglinear weights in the primary submission, and the third for testing. Apart from the named-entity translation rules described above, this material was not used in any other way, i.e. it was not incorporated into any phrase tables. The development data for tuning loglinear weights in the secondary submission was LDC's multiple-translation Chinese corpus, part 3.

CORPUS	USE	SENTENCES
TC-STAR dev., 1st two docs	Dev for sys1	336
Multiple translation, part 3 LDC2004T07	Dev for sys2	935
TC-STAR dev., last doc	Test	158
TC-STAR eval	Eval	1232

Table 4: Development and test corpora.

4.2 Results

We report all results using our version of the case-independent BLEU metric, which is identical to the official version except for small differences in tokenization. **Table 5** compares this metric with the official results on the evaluation set.

System	Official BLEU	NRC BLEU
Primary	13.67	13.29
Secondary	14.25	14.55

Table 5: Evaluation results with NRC's BLEU metric.

Table 6 shows the results of a set of tests to determine the best smoothing method for the secondary system, comparing plain relative frequencies, Good-Turing smoothing as described above, and Kneser-Ney smoothing as described in (Johnson *et al*., 2006). The results on the test corpus appear rather inconclusive: relative frequency estimation is the best technique when no rescoring step is used, but Good-Turing is best when rescoring is applied (using 1000-best lists, with the *base* feature group described below). As rescoring seems to give a slight advantage, we used rescored Good-Turing for the evaluation run. It can be seen from the *eval* column that rescored Kneser-Ney would have been a slightly better choice, though it is not clear that the differences in any of these results are statistically significant.

Smoothing	Test	Eval
Relative freq.	**16.42**	14.19
Good-Turing	15.63	14.10
Kneser-Ney	16.13	**14.61**
Rescored results		
Relative freq.	16.22	14.42
Good-Turing	**16.43**	14.55
Kneser-Ney	16.40	**14.67**

Table 6: Results for different smoothing techniques with secondary system.

Another set of tests was carried out to determine the best configuration for rescoring the primary system. The features added to the loglinear model (in addition to the basic set for decoding) are summarized in **Table 7**. All features were used in both directions (source to target and target to source) except for Consensus and charlen. Two groups of features were tested: base group comprising only IBM2 and MissingWord, and an extended group that includes all listed features.

Feature	Description
IBM1	IBM model 1 probability.
IBM2	IBM model 2 probability.
MissingWord	Sum over words ***t*** in target hypothesis of p(***t***|***sbest***)/p(***tbest***|***sbest***), where ***sbest*** is the best IBM1 translation for ***t*** in current source sentence, and ***tbest*** is the best known translation for ***sbest***.
Consensus	Average Levenshtein distance to all other hypotheses in nbest list.
charlen	Length in characters of hypothesis.

Table 7: Additional features used for rescoring.

The results of the rescoring tests are shown in **Table 8**, for different sizes of nbest list. We had planned to decide on the basis of experiments on the test corpus which type of rescoring to carry out on "eval" in the results submitted for the evaluation. On the basis of partial results on "test", we decided to submit Nbest=1 (no rescoring) results. From the results on "eval" carried out after the evaluation, it looks as though we made a wise choice! ***(Note to reviewers: as can be seen from the table, we have not yet finished this set of experiments. The complete results will, of course, be included in the final version of this paper if it is accepted).***

Nbest size	Feature set	Test	Eval
1	---	16.34	13.37
200	Base	16.65	13.25
200	Extended	17.12	13.25
500	Base	15.92	12.91
500	Extended	17.16	(ongoing)
1000	Base	16.40	13.18
1000	Extended	(ongoing)	(ongoing)

Table 8: BLEU Rescoring results for primary system.

Our final experiments are aimed at determining the effect of the named-entity translation rules on the operation of the primary system. Results are shown in **Table 9** and **Table 10**. **Table 9** shows that the named-entity rules increase BLEU by about 0.5 on "eval" data. We had speculated that it was the presence of these rules that gave us high weighted N-gram (WNM) scores on the evaluation, but the result doesn't seem to be clearcut: the rules apparently lower WNM/Recall (WNM/R), while raising WNM/F-measure (WMN/F). Note that we were not able to reproduce exactly the official WNM results, presumably owing to minor features of the tokenization used for evaluation.

Table 10 is related to the results shown in **Table 8** – it shows how much higher the BLEU scores shown in that table are compared to what they would have been without the named-entity rules. For instance, one can deduce from **Table 8** and **Table 10** that the score of the baseline system with no rescoring and no named-entity rules on "eval" is 13.37-1.23 = 12.14 BLEU. Note that the presence of the named-entity rules always leads to an improvement on "eval", ranging from an improvement of +2.05 BLEU to +0.49 BLEU. These rules were developed on the "test" corpus, so the improvements there are expected.

System	BLEU	WNM/R	WNM/F
Baseline	12.43	0.644	0.620
NE rules	12.91	0.621	0.633

Table 9: Effect of using Named-Entity translation rules according to different metrics on "Eval" (primary system)

Nbest size	Feature set	Test	Eval
1	---	+1.41	+1.23
200	Base	+0.57	+0.74
200	Extended	+2.05	+0.69
500	Base	+0.77	+0.49
500	Extended	+1.52	(ongoing)
1000	Base	+1.30	+0.96
1000	Extended	(ongoing)	(ongoing)

Table 10: BLEU improvements due to inclusion of Named-Entity rules for primary system

5 Discussion

From the point of view of the NRC group, one of the most interesting aspects of the TC-STAR results was the high scores obtained by PORTAGE (especially the primary submission) on the "weighted Ngram / Recall" and "weighted Ngram / F-measure" (WNM/RECALL and WNM/F-measure) metrics. This was certainly not due to tuning the system to these metrics, with which the NRC group was completely unfamiliar prior to the evaluation; all PORTAGE tuning used BLEU.

According to the program documentation and a paper describing these WNM metrics (Babych and Hartley, 2004), they are extensions of BLEU that weight N-grams with statistical salience scores (S-scores). These S-scores are very similar to the tf.idf scores used in information retrieval to assess information content. Thus, the WNM metrics assign greater importance to words that bear the most information than BLEU does; (Babych and Hartley, 2004) show that they correlate better with human judgments of adequacy and fluency, particularly the former, than BLEU does. We had speculated that the good performance of our system according to these metrics was due to our including a kind of named-entity translation module in our system (even if it was of the most primitive possible kind, consisting merely of a small list of name translations) since names are presumably a highly salient type of word. This hypothesis does not seem to be confirmed by **Table 9**. We are grateful that the TC-STAR evaluation exposed us to these WNM metrics, which seem interesting and valuable; we will continue to study them.

6 References

Agbago, A., Kuhn, R., and Foster, G. (2005). Truecasing for the Portage System. In *Int. Conf. on Recent Advances in Natural Language Processing (RANLP).* Borovets, Bulgaria: pp. 25-31.

Babych, B., and Hartley, A. (2004). Extending BLEU MT Evaluation Method with Frequency Weighting. In *Proc. of 42nd Annual Meeting of Assoc. for Computational Linguistics (ACL).* Barcelona, Spain.

Chen, S., and Goodman, J. (1999). An Empirical Study of Smoothing Techniques for Language Modeling. In *Computer Speech and Language*. Oct. 1999, V. 13, : pp. 359-394.

Church, K., and Gale, W. (1991). A Comparison of the Enhanced Good-Turing and Deleted Estimation Methods for Estimating Probabilities of English Bigrams. In *Computer Speech and Language.* V. 5, no. 1: pp. 19-54.

Goodman, J. (2001). A Bit of Progress in Language Modeling (extended version). *Microsoft Research Technical Report 2001-72*. Downloadable from research.microsoft.com/~joshuago/publications.htm

Johnson, H., Sadat, F., Foster, G., Kuhn, R., Simard, M., Joanis, E., and Larkin, S. (2006). PORTAGE: with Smoothed Phrase Tables and Segment Choice Models. Accepted for publication in *NAACL 2006 Workshop on Statistical Machine Translation (WMT06)*. To be held in New York City, June 2006.

Koehn, P., Och, F.-J., and Marcu, D. (2003). Statistical Phrase-Based Translation. In *Proc. of Human Language Technology Conf. of North American Chapter of Assoc. for Computational Linguistics (HLT/NAACL).* Edmonton, Alberta, Canada: pp. 127-133.

Koehn, P., Axelrod, A., Mayne, A., Callison-Burch, C., Osborne, M., Talbot, D., and White, M. (2005). Edinburgh System Description for the 2005 NIST MT Evaluation. In *Proc. Of Machine Translation Evaluation Workshop.*

Kuhn, R., Yuen, D., Simard, M., Foster, G., Paul, P., Joanis, E., and Johnson, H. (2006). Segment Choice Models: Feature-Rich Models for Global Distortion in Statistical Machine Translation. Accepted for publication in *Proc. of Human Language Technology Conf. of North American Chapter of Assoc. for Computational Linguistics (HLT/NAACL).* To be held in New York City, USA, June 2006.

Och, F.-J. (2003). Minimum Error Rate Training for Statistical Machine Translation. In *Proc. of 41st Annual Meeting of Assoc. for Computational Linguistics (ACL).* Sapporo, Japan.

Sadat, F., Johnson, H., Agbago, A., Foster, G., Kuhn, R., Martin, J. and Tikuisis, A. (2005). Portage: A Phrase-

based Machine Translation System. In *Proc. of ACL 2005 Workshop on Building and Using Parallel Texts: Data-Driven Machine Translation and Beyond*. Ann Arbor, Michigan, USA.

Stolcke, A. (2002). SRILM - An Extensible Language Modeling Toolkit. In *Int. Conf. Spoken Language Processing (ICSLP02)*. Denver, Colorado, USA: pp. 901-904.

Zens, R. and Ney, H. (2004). Improvements in Phrase-Based Statistical Machine Translation. In *Proc. of Human Language Technology Conf. of North American Chapter of Assoc. for Computational Linguistics (HLT/NAACL)*, Boston, Massachusetts, USA.