Mood at work: Ramses versus Pharaoh

Alexandre Patry, Fabrizio Gotti and Philippe Langlais

RALI — DIRO
Université de Montréal
{patryale,gottif,felipe}@iro.umontreal.ca

Abstract

We present here the translation system we used in this year's WMT shared task. The main objective of our participation was to test RAMSES, an open source phrase-based decoder. For that purpose, we used the baseline system made available by the organizers of the shared task[1] to build the necessary models. We then carried out a pair-to-pair comparison of RAMSES with PHARAOH on the six different translation directions that we were asked to perform. We present this comparison in this paper.

1 Introduction

Phrase-based (PB) machine translation (MT) is now a popular paradigm, partly because of the relative ease with which we can automatically create an acceptable translation engine from a bitext. As a matter of fact, deriving such an engine from a bitext consists in (more or less) gluing together dedicated software modules, often freely available. Word-based models, or the so-called IBM models, can be trained using the GIZA or GIZA++ toolkits (Och and Ney, 2000). One can then train phrase-based models using the THOT toolkit (Ortiz-Martínez et al., 2005). For their part, language models currently in use in SMT systems can be trained using packages such as SRILM (Stolcke, 2002) and the CMU-SLM toolkit (Clarkson and Rosenfeld, 1997).

[1] www.statmt.org/wmt06/shared-task/baseline.html

Once all the models are built, one can choose to use PHARAOH (Koehn, 2004), an efficient full-fledged phrase-based decoder. We only know of one major drawback when using PHARAOH: its licensing policy. Indeed, it is available for non-commercial use in its binary form only. This severely limits its use, both commercially and scientifically (Walker, 2005).

For this reason, we undertook the design of a generic architecture called MOOD (Modular Object-Oriented Decoder), especially suited for instantiating SMT decoders. Two major goals directed our design of this package: offering open source, state-of-the-art decoders and providing an architecture to easily build these decoders. This effort is described in (Patry et al., 2006).

As a proof of concept that our framework (MOOD) is viable, we attempted to use its functionalities to implement a clone of PHARAOH, based on the comprehensive user manual of the latter. This clone, called RAMSES, is now part of the MOOD distribution, which can be downloaded freely from the page http://smtmood.sourceforge.net.

We conducted a pair-to-pair comparison between the two engines that we describe in this paper. We provide an overview of the MOOD architecture in Section 2. Then we describe briefly RAMSES in Section 3. The comparison between the two decoders in terms of automatic metrics is analyzed in Section 4. We confirm this comparison by presenting a manual evaluation we conducted on an random sample of the translations produced by both decoders. This is reported in Section 5. We conclude in Section 6.

2 The MOOD Framework

A decoder must implement a specific combination of two elements: a model representation and a search space exploration strategy. MOOD is a framework designed precisely to allow such a combination, by clearly separating its two elements. The design of the framework is described in (Patry et al., 2006).

MOOD is implemented with the C++ programming language and is licensed under the Gnu General Public License (GPL)[2]. This license grants the right to anybody to use, modify and distribute the program and its source code, provided that any modified version be licensed under the GPL as well. As explained in (Walker, 2005), this kind of license stimulates new ideas and research.

3 MOOD at work: RAMSES

As we said above, in order to test our design, we reproduced the most popular phrase-based decoder, PHARAOH (Koehn, 2004), by following as faithfully as possible its detailed user manual. The command-line syntax RAMSES recognizes is compatible with that of PHARAOH. The output produced by both decoders are compatible as well and RAMSES can also output its n-best lists in the same format as PHARAOH does, i.e. in a format that the CARMEL toolkit can parse (Knight and Al-Onaizan, 1999). Switching decoders is therefore straightforward.

4 RAMSES versus PHARAOH

To compare the translation performances of both decoders in a meaningful manner, RAMSES and PHARAOH were given the exact same language model and translation table for each translation experiment. Both models were produced with the scripts provided by the organizers. This means in practice that the language model was trained using the SRILM toolkit (Stolcke, 2002). The word alignment required to build the phrase table was produced with the GIZA++ package. A Viterbi alignment computed from an IBM model 4 (Brown et al., 1993) was computed for each translation direction. Both alignments were then combined in a heuristic way (Koehn et al.,). Each pair of phrases in the model is given 5 scores, described in the PHARAOH training manual.[3]

To tune the coefficients of the log-linear combination that both PHARAOH and RAMSES use when decoding, we used the organizers' `minimum-error-rate-training.perl` script. This tuning step was performed on the first 500 sentences of the dedicated development corpora. Inevitably, RAMSES differs slightly from PHARAOH, because of some undocumented embedded heuristics. Thus, we found appropriate to tune each decoder separately (although with the same material). In effect, each decoder does slightly better (with BLEU) when it uses its own best parameters obtained from tuning, than when it uses the parameters of its counterpart.

Eight coefficents were adjusted this way: five for the translation table (one for each score associated to each pair of phrases), and one for each of the following models: the language model, the so-called word penalty model and the distortion model (word reordering model). Each parameter is given a starting value and a range within which it is allowed to vary. For instance, the language model coefficient's starting value is 1.0 and the coefficient is in the range [0.5–1.5]. Eventually, we obtained two optimal configurations (one for each decoder) with which we translated the TEST material.

We evaluated the translations produced by both decoders with the organizers' `multi-bleu.perl` script, which computes a BLEU score (and displays the n-gram precisions and brevity penalty used). We report the scores we gathered on the test corpus of 2000 pairs of sentences in Table 1. Overall, both decoders offer similar performances, down to the n-gram precisions. To assess the statistical significance of the observed differences in BLEU, we used the bootstrapping technique described in (Zhang and Vogel, 2004), randomly selecting 500 sentences from each test set, 1000 times. Using a 95% confidence interval, we determined that the small differences between the two decoders are not statistically significant, except for two tests. For the direction English to French, RAMSES outperforms PHARAOH, while in the German to English direc-

[2] `http://www.gnu.org/copyleft/gpl.html`

[3] `http://www.statmt.org/wmt06/shared-task/training-release-1.3.tgz`

tion, PHARAOH is better. Whenever a decoder is better than the other, Table 1 shows that it is attributable to higher n-gram precisions; not to the brevity penalty.

We further investigated these two cases by calculating BLEU for subsets of the test corpus sharing similar sentence lengths (Table 2). We see that both decoders have similar performances on short sentences, but can differ by as much as 1% in BLEU on longer ones. In contrast, on the Spanish-to-English translation direction, where the two decoders offer similar performances, the difference between BLEU scores never exceeds 0.23%.

Expectedly, Spanish and French are much easier to translate than German. This is because, in this study, we did not apply any pre-processing strategy that we know can improve performances, such as clause reordering or compound-word splitting (Collins et al., 2005; Langlais et al., 2005).

Table 2 shows that it does not seem much more difficult to translate into English than from English. This is surprising: translating into a morphologically richer language should be more challenging. The opposite is true for German here: without doing anything specific for this language, it is much easier to translate from German to English than the other way around. This may be attributed in part to the language model: for the test corpus, the perplexity of the language models provided is 105.5 for German, compared to 59.7 for English.

D	BLEU	p_1	p_2	p_3	p_4	BP
			es → en			
P	30.65	64.10	36.52	23.70	15.91	1.00
R	30.48	64.08	36.30	23.52	15.76	1.00
			fr → en			
P	30.42	64.28	36.45	23.39	15.64	1.00
R	30.43	64.58	36.59	23.54	15.73	0.99
			de → en			
P	25.15	61.19	31.32	18.53	11.61	0.99
R	24.49	61.06	30.75	17.73	10.81	1.00
			en → es			
P	29.40	61.86	35.32	22.77	15.02	1.00
R	28.75	62.23	35.03	22.32	14.58	0.99
			en → fr			
P	30.96	61.10	36.56	24.49	16.80	1.00
R	31.79	61.57	37.38	25.30	17.53	1.00
			en → de			
P	18.03	52.77	22.70	12.45	7.25	0.99
R	18.14	53.38	23.15	12.75	7.47	0.98

Table 1: Performance of RAMSES and PHARAOH on the provided test set of 2000 pairs of sentences per language pair. **P** stands for PHARAOH, **R** for RAMSES. All scores are percentages. p_n is the n-gram precision and BP is the brevity penalty used when computing BLEU.

5 Human Evaluation

In an effort to correlate the objective metrics with human reviews, we undertook the blind evaluation of a sample of 100 pairwise translations for the three Foreign language-to-English translation tasks. The pairs were randomly selected from the 3064 translations produced by each engine. They had to be different for each decoder and be no more than 25 words long.

Each evaluator was presented with a source sentence, its reference translation and the translation produced by each decoder. The last two were in random order, so the evaluator did not know which engine produced the translation. The evaluator's task was two-fold. (1) He decided whether one translation was better than the other. (2) If he replied 'yes' in test (1), he stated whether the best translation was satisfactory while the other was not. Two evaluators went through the 3×100 sentence pairs. None of them understands German; subject B understands Spanish, and both understand French and English. The results of this informal, yet informative exercise are reported in Table 3.

Overall, in many cases (64% and 48% for subject A and B respectively), the evaluators did not prefer one translation over the other. On the Spanish- and French-to-English tasks, both subjects slightly preferred the translations produced by RAMSES. In about one fourth of the cases where one translation was preferred did the evaluators actually flag the selected translation as significantly better.

6 Discussion

We presented a pairwise comparison of two decoders, RAMSES and PHARAOH. Although RAMSES is roughly twice as slow as PHARAOH, both de-

Test set		[0,15]	[16,25]	[26,∞[
en → fr	(**P**)	33.52	30.65	30.39
en → fr	(**R**)	33.78	31.19	31.35
de → en	(**P**)	29.74	24.30	24.76
de → en	(**R**)	29.85	23.92	23.78
es → en	(**P**)	34.23	28.32	30.60
es → en	(**R**)	34.46	28.39	30.40

Table 2: BLEU scores on subsets of the test corpus filtered by sentence length ([min words, max words] intervals), for **P**haraoh and **R**amses.

	`Preferred`			`Improved`	
	P	R	No	P	R
			es → en		
subject A	13	16	71	6	1
subject B	23	31	46	3	8
			fr → en		
subject A	18	19	63	5	3
subject B	20	21	59	8	8
			de → en		
subject A	24	18	58	5	9
subject B	30	31	39	3	3
Total	128	136	336	30	32

Table 3: Human evaluation figures. The column `Preferred` indicates the preference of the subject (**P**haraoh, **R**amses or **No** preference). The column `Improved` shows when a subject did prefer a translation and also said that the preferred translation was correct while the other one was not.

coders offer comparable performances, according to automatic and informal human evaluations.

Moreover, RAMSES is the product of clean framework: MOOD, a solid tool for research projects. Its code is open source and the architecture is modular, making it easier for researchers to experiment with SMT. We hope that the availability of the source code and the clean design of MOOD will make it a useful platform to implement new decoders.

Acknowledgments

We warmly thanks Elliott Macklovitch for his participation in the manual annotation task. This work has been partially funded by an NSERC grant.

References

P.F. Brown, S.A. Della Pietra, V.J. Della Pietra, and R.L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311.

P. Clarkson and R. Rosenfeld. 1997. Statistical language modeling using the CMU-cambridge toolkit. In *Proc. of Eurospeech*, pages 2707–2710, Rhodes, Greece.

M. Collins, P. Koehn, and I. Kucerova. 2005. Clause restructuring for statistical machine translation. In *Proc. of the 43rd ACL*, pages 531–540, Ann Arbor, MI.

K. Knight and Y. Al-Onaizan, 1999. *A Primer on Finite-State Software for Natural Language Processing*. www.isi.edu/licensed-sw/carmel.

P. Koehn, F. Joseph Och, and D. Marcu. Statistical Phrase-Based Translation. In *Proc. of HLT*, Edmonton, Canada.

P. Koehn. 2004. Pharaoh: a Beam Search Decoder for Phrase-Based SMT. In *Proc. of the 6th AMTA*, pages 115–124, Washington, DC.

P. Langlais, G. Cao, and F. Gotti. 2005. RALI: SMT shared task system description. In *2nd ACL workshop on Building and Using Parallel Texts*, pages 137–140, Ann Arbor, MI.

F.J. Och and H. Ney. 2000. Improved Statistical Alignment Models. In *Proc. of ACL*, pages 440–447, Hongkong, China.

D. Ortiz-Martínez, I. Garciá-Varea, and F. Casacuberta. 2005. Thot: a toolkit to train phrase-based statistical translation models. In *Proc. of MT Summit X*, pages 141–148, Phuket, Thailand.

A. Patry, F. Gotti, and P. Langlais. 2006. MOOD a modular object-oriented decoder for statistical machine translation. In *Proc. of LREC*, Genoa, Italy.

A. Stolcke. 2002. SRILM - an Extensible Language Modeling Toolkit. In *Proc. of ICSLP*, Denver, USA.

D.J. Walker. 2005. The open "a.i." kit^{TM}: General machine learning modules from statistical machine translation. In *Workshop of MT Summit X, "Open-Source Machine Translation"*, Phuket, Thailand.

Ying Zhang and Stephan Vogel. 2004. Measuring confidence intervals for the machine translation evaluation metrics. In *Proc. of the 10th TMI*, Baltimore, MD.