RALI: SMT shared task system description

Philippe Langlais, Guihong Cao and Fabrizio Gotti
RALI
Département d'Informatique et de Recherche Opérationnelle
Université de Montréal
Succursale Centre-Ville
H3C3J7 Montréal, Canada
http://rali.iro.umontreal.ca

Abstract

Thanks to the profusion of freely available tools, it recently became fairly easy to built a statistical machine translation (SMT) engine given a bitext. The expectations we can have on the quality of such a system may however greatly vary from one pair of languages to another. We report on our experiments in building phrase-based translation engines for the four pairs of languages we had to consider for the SMT shared-task.

1 Introduction

Machine translation is nowadays mature enough that it is possible without too much effort to devise automatically a statistical translation system from just a parallel corpus. This is possible thanks to the dissemination of valuable packages. The performance of such a system may however greatly vary from one pair of languages to another. Indeed, there is no free lunch for system developers, and if a black box approach can sometimes be good enough for some applications (we can surely accomplish translation *gisting* with the French-English and Spanish-English systems we developed during this exercice), making use of the output of such a system for, let's say, quality translation is another kettle of fish (especially in our case with the Finnish-English system we ended-up with).

We devoted two weeks to the SMT shared task, the aim of which was precisely to see how well systems can do across different language families. We began with a core system which is described in the next section and from which we obtained baseline performances that we tried to improve upon.

Since the French- and Spanish-English systems produced output that were comprehensible enough[1], we focussed on the two languages whose translations were noticeably worse: German and Finnish. For German, we tried to move around words in order to mimic English word order; and we tried to split compound words. This is described in section 4. For the Finnish/English pair, we tried to decompose Finnish words into smaller substrings (see section 5).

In parallel to that, we tried to smooth a phrase-based model (PBM) making use of WORDNET. We report on this experiment in section 3. We describe in section 6 the final setting of the systems we used for submitting translations and their official results as computed by the organizers. Finally, we conclude our two weeks of efforts in section 7.

2 The core system

We assembled up a phrase-based statistical engine by making use of freely available packages. The translation engine we used is the one suggested within the shared task: PHARAOH (Koehn, 2004). The input of this decoder is composed of a phrase-based model (PBM), a trigram language model and an optional set of coefficients and thresholds

[1] What we mean by this is nothing more than we were mostly able to infer the original meaning of the source sentence by reading its automatic translation.

pair	WER	SER	NIST	BLEU
fi-en	66.53	99.20	5.3353	18.73
de-en	60.70	98.40	5.8411	21.11
fr-en	53.77	98.20	6.4717	27.69
es-en	53.84	98.60	6.5571	28.08

Table 1: Baseline performances measured on the 500 top sentences of the DEV corpus in terms of WER (word error rate), SER (sentence error rate), NIST and BLEU scores.

which control the decoder.

For acquiring a PBM, we followed the approach described by Koehn et al. (2003). In brief, we relied on a bi-directional word alignment of the training corpus to acquire the parameters of the model. We used the word alignment produced by Giza (Och and Ney, 2000) out of an IBM model 2. We did try to use the alignment produced with IBM model 4, but did not notice significant differences over our experiments; an observation consistent with the findings of Koehn et al. (2003). Each parameter in a PBM can be scored in several ways. We considered its relative frequency as well as its IBM-model 1 score (where the transfer probabilities were taken from an IBM model 2 transfer table). The language model we used was the one provided within the shared task.

We obtained baseline performances by tuning the engine on the top 500 sentences of the development corpus. Since we only had a few parameters to tune, we did it by sampling the parameter space uniformly. The best performance we obtained, *i.e.*, the one which maximizes the BLEU metric as measured by the `mteval` script[2] is reported for each pair of languages in Table 1.

3 Smoothing PBMs with WORDNET

Among the things we tried but which did not work well, we investigated whether smoothing the transfer table of an IBM model (2 in our case) with WORDNET would produce better estimates for rare words. We adapted an approach proposed by Cao et al. (2005) for an Information Retrieval task, and computed for any parameter (e_i, f_j) belonging to the original model the following approximation:

$$\dot{p}(e_i|f_j) \approx \sum_{e \in \mathcal{E}} p_{wn}(e_i|e) \times p_n(e|f_j)$$

where $\mathcal{E}$ is the English vocabulary, p_n designates the native distribution and p_{wn} is the probability that two words in the English side are linked together. We estimated this distribution by co-occurrence counts over a large English corpus[3]. To avoid taking into account unrelated but co-occurring words, we used WORDNET to filter in only the co-occurrences of words that are in relation according to WORDNET. However, since many words are not listed in this resource, we had to smooth the bigram distribution, which we did by applying Katz smoothing (Katz, 1997):

$$p_{katz}(e_i|e) = \begin{cases} \frac{\dot{c}(e_i,e|W,L)}{\sum_{e_j} c(e_j,e|W,L)} & \text{if } c(e_i,e|W,L) > 0 \\ \alpha(e) p_{katz}(e_i) & \text{otherwise} \end{cases}$$

where $\dot{c}(a,b|W,L)$ is the good-turing discounted count of times two words a and b that are linked together by a WORDNET relation, co-occur in a window of 2 sentences.

We used this smoothed model to score the parameters of our PBM instead of the native transfer table. The results were however disappointing for both the G-E and S-E translation directions we tested. One reason for that, may be that the English corpus we used for computing the co-occurrence counts is an out-of-domain corpus for the present task. Another possible explanation lies in the fact that we considered both synonymic and hyperonymic links in WORDNET; the latter kind of links potentially introducing too much noise for a translation task.

4 The German-English task

We identified two major problems with our approach when faced with this pair of languages. First, the tendency in German to put verbs at the end of a phrase happens to ruin our phrase acquisition process, which basically collects any box of aligned source and target adjacent words. This

[2] `http://www.nist.gov/speech/tests/mt/mt2001/resource`

[3] For this, we used the English side of the provided training corpus plus the English side of our in-house Hansard bitext; that is, a total of more than 7 million pairs of sentences.

can be clearly seen in the alignment matrix of figure 1 where the verbal construction *could clarify* is translated by two very distant German words *könnten* and *erläutern*. Second, there are many compound words in German that greatly dilute the various counts embedded in the PBM table.

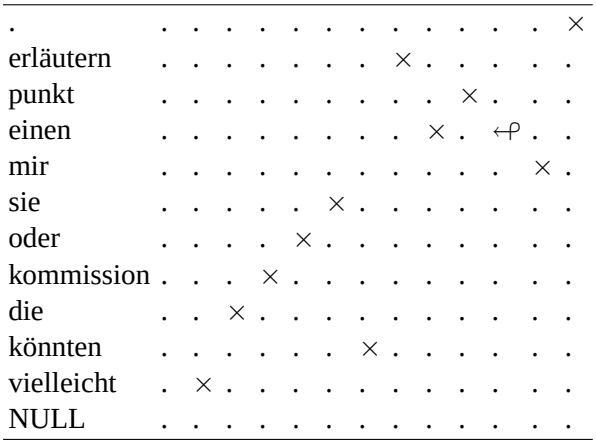

Figure 1: Bidirectional alignment matrix. A cross in this matrix designates an alignment valid in both directions, while the $\leftarrow\!\!\rho$ symbol indicates an uni-directional alignment (`for` has been aligned with `einen`, but not the other way round).

4.1 Moving around German words

For the first problem, we applied a memory-based approach to move around words in the German side in order to better synchronize word order in both languages. This involves, first, to learning transformation rules from the training corpus, second, transforming the German side of this corpus; then training a new translation model. The same set of rules is then applied to the German text to be translated.

The transformation rules we learned concern a few (five in our case) verbal constructions that we expressed with regular expressions built on POS tags in the English side. Once the *locus* e_u^v of a pattern has been identified, a rule is collected whenever the following conditions apply: for each word e in the locus, there is a target word f which is aligned to e in both alignment directions; these target words when moved can lead to a diagonal going from the target word (l) associated to e_{u-1} to the target word r which is aligned to e_{v+1}.

The rules we memorize are triplets (c, i, o) where $c = (l, r)$ is the context of the locus and i and o are the input and output German word order (that is, the order in which the tokens are found, and the order in which they should be moved).

For instance, in the example of Figure 1, the `Verb Verb` pattern match the locus *could clarify* and the following rule is acquired: `(sie einen, könnten erläutern, könnten erläutern)`, a paraphrase of which is: "whenever you find (in this order) the word *könnten* and *erläutern* in a German sentence containing also (in this order) *sie* and *einen*, move *könnten* and *erläutern* between *sie* and *einen*.

A set of 124 271 rules have been acquired this way from the training corpus (for a total of 157 970 occurrences). The most frequent rule acquired is `(ich herrn, möchte danken, möchte danken)`, which will transform a sentence like "*ich möchte herrn wynn für seinen bericht danken.*" into "*ich möchte danken herrn wynn für seinen bericht.*".

In practice, since this acquisition process does not involve any generalization step, only a few rules learnt really fire when applied to the test material. Also, we devised a fairly conservative way of applying the rules, which means that in practice, only 3.5% of the sentences of the test corpus where actually modified.

The performance of this procedure as measured on the development set is reported in Table 2. As simple as it is, this procedure yields a relative gain of 7% in BLEU. Given the crudeness of our approach, we consider this as an encouraging improvement.

4.2 Compound splitting

For the second problem, we segmented German words before training the translation models. Empirical methods for compound splitting applied to

system	WER	SER	NIST	BLEU
baseline	60.70	98.40	5.8411	21.11
swap	**60.73**	**98.60**	**5.9643**	**22.58**
split	60.67	98.60	5.7511	21.99
swap+split	60.57	98.40	5.9685	23.10

Table 2: Performances of the swapping and the compound splitting approaches on the top 500 sentences of the development set.

German have been studied by Koehn and Knight (2003). They found that a simple splitting strategy based on the frequency of German words was the most efficient method of the ones they tested, when embedded in a phrase-based translation engine. Therefore, we applied such a strategy to split German words in our corpora. The results of this approach are shown in Table 2.

Note: Both the swapping strategy and the compound splitting yielded improvements in terms of BLEU score. Only after the deadline did we find time to train new models with a combination of both techniques; the results of which are reported in the last line of Table 2.

5 The Finnish-English task

The worst performances were registered on the Finnish-English pair. This is due to the agglutinative nature of Finnish. We tried to segment the Finnish material into smaller units (substrings) by making use of the frequency of all Finnish substrings found in the training corpus. We maintained a suffix tree structure for that purpose. We proceeded by recursively finding the most promising splitting points in each Finnish token of C characters F_1^C by computing $split(F_1^C)$ where:

$$split(F_i^j) = \begin{cases} f(F_i^j) & \text{if } j-i<2 \\ \text{argmax}_{c\in[i+2,j-2]}\, f(F_i^c)\times \\ \quad split(F_{c+1}^j) & \text{otherwise} \end{cases}$$

This approach yielded a significant degradation in performance that we still have to analyze.

6 Submitted translations

At the time of the deadline, the best translations we had were the baselines ones for all the language pairs, except for the German-English one where the moving of words ranked the best. This defined the configuration we submitted, whose results (as provided by the organizers) are reported in Table 3.

pair	BLEU	$p_1/p_2/p_3/p_4$
fi-en	18.87	55.2/24.7/13.1/7.1
de-en	22.91	58.9/29.0/16.8/10.3
es-en	28.49	62.4/34.5/21.9/14.4
fr-en	28.89	62.6/34.7/22.0/14.6

Table 3: Results measured by the organizers for the TEST corpus.

7 Conclusion

We found that, while comprehensible translations were produced for pairs of languages such as French-English and Spanish-English; things did not go as well for the German-English pair and especially not for the Finnish-English pair. We had a hard time improving our baseline performance in such a tight schedule and only managed to improve our German-English system. We were less lucky with other attempts we implemented, among them, the smoothing of a transfer table with WORDNET, and the segmentation of the Finnish corpus into smaller units.

References

G. Cao, J. Nie, and J. Bai. 2005. Integrating Word relationships into Language Models. In *to appear in Proc. of SIGIR*.

S. Katz. 1997. Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. *IEEE Transactions on Acoustics Speech and Signal Processing*, 35.

Philipp Koehn and Kevin Knight. 2003. Empirical Methods for Compound Splitting. In *EACL*, Budapest, Hungary.

P. Koehn, F.J. Och, and D. Marcu. 2003. Statistical Phrase-Based Translation. In *Proceedings of HLT*, pages 127–133.

P. Koehn. 2004. Pharaoh: a Beam Search Decoder for Phrase-Based SMT. In *Proceedings of AMTA*, pages 115–124.

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