Merging Example-Based and Statistical Machine Translation: An Experiment

Philippe Langlais and Michel Simard

Laboratoire de Recherche Appliquée en Linguistique Informatique (RALI)
Département d'Informatique et de Recherche Opérationnelle
Université de Montréal
C.P. 6128, succursale Centre-ville
H3C 3J7, Montréal, Québec, Canada
http://www.rali-iro.umontreal.ca

Abstract. Despite the exciting work accomplished over the past decade in the field of Statistical Machine Translation (SMT), we are still far from the point of being able to say that machine translation fully meets the needs of real-life users. In a previous study [6], we have shown how a SMT engine could benefit from terminological resources, especially when translating texts very different from those used to train the system. In the present paper, we discuss the opening of SMT to examples automatically extracted from a Translation Memory (TM). We report results on a fair-sized translation task using the database of a commercial bilingual concordancer.

1 Introduction

The past decade witnessed exciting work in the field of Statistical Machine Translation. We are still far however from the point of being able to say that machine translation fully meets the needs of real-life users and it is a well known fact that translators remain reluctant to post-edit the output of a machine translator (statistical or not).

The present work is largely inspired by two studies we have previously conducted. In a first one [6], we investigated how a statistical engine behaves when translating a very domain-specific text far different from the corpus used to train both the translation and language models used by the engine. We measured a significant drop in performances mainly due to *out-of-vocabulary* (unknown) words and specific terminology that the models handle poorly. We proposed to overcome the problem by providing the engine with available (non statistical) terminological resources.

In a second study [13], we investigated how a database of past translations could help a human translator in his work. Such *Translation Memories* (TM) already exist, but typically operate on complete sentences, thus limiting their usefulness. We showed in our study that an impressive coverage of the source-language (SL) text could be obtained (up to 95%) by systematically querying the memory with sub-sentential sequences of words of the text to translate,

from which we may automatically retrieve useful target-language (TL) material. Other works in the same vein reported comparable encouraging results [3].

In the present study, we extend these lines of work by feeding a statistical translation engine with examples automatically extracted from the database of an online bilingual concordancer: TSRALI[1]. This concordancer allows a user to query a large collection of French-English bitexts (more than 100 million words per language), aligned at the level of sentences. A full description is given in [8].

The approach we investigate here involves three main steps which are described in the following sections. The first consists in chunking the SL text to be translated; each identified chunk is then submitted to the TM (Section 2). The second step extracts from all the TL material returned by a query the portions that are likely to be useful for a translation (Section 3). In a third step (Section 4), these pieces of TL text are fed to a statistical engine in order to produce a translation. In Section 5, we report on an experiment we conducted made on a fair-sized corpus. We conclude with a discussion in Section 6.

2 Looking up SL Sequences in tsrali

In this paper, we call *Translation Memory* (TM) a database of existing translations. Conceptually, it can be viewed as a collection of pairs $< S, T >$, where S is a SL segment of text, T is a TL segment, and S and T are translations of one another. In our case, S and T are typically single sentences, although in some cases S or T may be empty ("untranslated sentences") or consist of a short sequence of sentences (anywhere between 2 and 5). We call these pairs *couples*.

Using standard full-text indexation techniques, it is possible to efficiently extract from such a collection all couples that contain some given sequence of word-forms in one language or the other. This is precisely what TSRALI is designed to do. Given a SL sentence $S = s_1...s_m$, our plan is to use such a TM to propose TL translations for partial sequences s_i^j of S.

In previous work using a similar setup [13], we established that concentrating on *syntactically motivated* sequences of S was more productive than looking up all possible sequences. To identify these sequences in S, we employ a *chunker*, i.e. a system that identifies basic syntactic constituents.

Our chunker essentially follows the lines of [11]: it relies on a part-of-speech tagger (in our case, a hidden Markov model rather than a maximum entropy tagger), and proceeds in successive tagging stages, each working on the previous stages' output. The first stage is a standard POS tagger: it associates a POS tag p_i to each word-token of s_i of S. The second stage takes as input a symbol obtained by combining s_i and p_i, and outputs so-called *IOB* tags c_i. These tags take one of the forms: *B-X*: first word of a chunk of type X; *I-X*: non-initial word in a X chunk; *O*: word outside of any chunk.

The last and final stage is designed to provide the chunker with more context: it takes as input a symbol obtained by combining POS and IOB tags p_i, c_i, p_{i+1}

[1] `http://www.tsrali.com`.

and c_{i+1}, and produces "revised" IOB tags c'_i on the output. An example of the resulting bracketing is shown in Figure 1.

[$_{NP}$ The government] [$_{VP}$ is putting] [$_{NP}$ a $2.2 billion tax] [$_{PP}$ on]
[$_{NP}$ Canada] [$_{NP}$'s most vulnerable industry] , [$_{NP}$ the airline industry] .

Fig. 1. Output of the chunker.

We then search our TM for all sequences that begin and end at chunk boundaries (sequences of O tags are viewed as chunks in this process). We also exclude search sequences of less than two words, and sequences exclusively made up of very frequent word-forms (we use a stop-list of the 20 most frequent SL words). Figure 2 shows the matching sequences for the example of Figure 1, and Figure 3 shows a sample matching couple for one of the sequences found in the TM.

The government / The government is putting / is putting / a $2.2 billion tax /
a $2.2 billion tax on / a $2.2 billion tax on Canada / Canada 's most vulnerable
industry / 's most vulnerable industry / , the airline industry / the airline industry

Fig. 2. The 10 sequences found in the translation memory.

Source: Yes **, the airline industry** is an important industry .
Target: Oui , l' industrie aérienne est un secteur important .

Fig. 3. Sample match for sequence "*, the airline industry*".

3 Identifying Potentially Useful TL Units

For each SL sequence s_i^j of S, we extract a (possibly empty) set of couples from the TM. In order to come up with translation proposals for the sequence s_i^j, from each of these couples $< S_k, T_k >$, we must now identify the part of T_k that translates the initial sequence. We make the simplifying assumption that this translation will itself be a sequence of words from T_k (no discontiguous translations).

For this task, we use a sequence alignment method that recursively segments the SL and TL text, each time choosing the segmentation that maximizes an association score between the matched pairs of segments. This scoring function approximates $P(t_k^l|s_i^j)$, the probability of observing the TL sequence t_k^l, given the SL sequence s_i^j:

$$Score(i, j+1, k, l+1) = \delta(j-i|l-k) \prod_{K=k}^{l} \sum_{I=i}^{j} \frac{tr(t_K|s_I)}{j-i} \qquad (1)$$

where the $tr(t|s)$ are the lexical parameters of a statistical translation model (IBM model 1 [2]) and $\delta(m|n)$ represents the probability of observing a sequence of m words as the translation of a sequence of n words. In practice, we also make the simplifying assumption that the δ distribution is uniform over "reasonable" values of m.

Given a pair of sequences $< s_i^{j-1}, t_k^{l-1} >$, the alignment procedure finds optimal segmentation points I and K, and the best way of pairing up the resulting sub-sequences (in *parallel*, or in *reverse*):

$$< I, K, d >= argmax_{I,K,d} \begin{cases} Score(i, I, k, K) \times Score(I, j, K, l) \ (d = parallel) \\ Score(i, I, K, l) \times Score(I, j, k, K) \ (d = reverse) \end{cases}$$

It then proceeds recursively on pairs of sequences $< s_i^{I-1}, t_k^{K-1} >$ and $< s_I^j, t_K^l >$ (or $< s_i^{I-1}, t_K^l >$ and $< s_I^j, t_k^{K-1} >$ if $d = reverse$).
We have found that we can both improve alignment results and significantly reduce the search-space for this procedure by forcing it to consider only "syntactically motivated" segmentation points I and K. To do this, we first run the SL and TL segments of each couple through text chunkers, identical to the one described in Section 2. We then consider as valid only those segmentation points that lie at chunk boundaries. Figure 5 shows sample alignments for two different SL sentences.

4 Merging Example-based and Statistical MT

4.1 The Translation Engine

We have extended the decoder (statistical machine translator) of [10] to a trigram language model. The basic idea of this search algorithm is to expand hypotheses along the positions of the TL string while progressively covering the SL ones. Every TL word may be associated with any l adjacent SL words; the decoder thus accounts for the notion of *fertility* [2], even if IBM model 2 does not incorporate this notion.

The decoder is a dynamic programming scheme based on a recursion which results from straight manipulations of the following maximization equation, where the SL sentence to translate is s_1^J, and l indicates the fertility of the TL word t_i:

$$\hat{t}_1^{\hat{I}} = \max_I \left[\underbrace{p(J|I)}_{length} \max_{t_1^I} \prod_{i=1}^{I} \left[\underbrace{p(t_i|t_{i-2}t_{i-1})}_{trigram} \max_{j,l} \prod_{\bar{j}=j-l+1}^{j} \left\{ \underbrace{p(i|\bar{j}, J, I)}_{alignment} \underbrace{p(s_{\bar{j}}|t_i)}_{transfer} \right\} \right] \right] \quad (2)$$

We refer the reader to [10] for the formal description of the recursion, and instead give in Figure 4 a sketch of how a translation is built-up.

4.2 Feeding the TL Examples to the Decoder

There are many possible strategies to integrate the contributions of the TM into the decoding process. One of these is to "reward" the decoder whenever it

Input: $s_1 \ldots s_j \ldots s_J$

for all TL position $i = 1, 2, \ldots, I_{max}$ **do**
  for all valid hypothesis at stage $i - 1$ **do**
    for all TL word t_i **do**
      for all free SL position j **do**
        for all fertility l **do**
          Consider t_i to be the translation of the l SL words s_j^{j+l-1}

Fig. 4. Sketch of our decoder

generates hypotheses that contain part or all of a TL example sequence. However, because of the various pruning strategies used to keep the decoding time reasonable, even highly promising TL hypotheses may never be examined.

Instead, we investigated an approach that rests on the assumption that among the TL sequences extracted from the TM, there must be at least one which corresponds to a valid (usable) translation of the associated SL sequence.

In this perspective, the task of the statistical engine is to discover which TM sequences are most likely to be useful, as well as to determine (by optimizing equation 2 over the full sentence) the most likely target positions of these sequences. Hence what we are looking for is the most likely sentence that contains one TL sequence from the TM per matched SL sequence.

In the extreme case, if in the sentence of Figure 1, the only sequence submitted to the TM was *the airline industry*, with only one association returned *l'industrie du transport aérien*, our search algorithm would end up with a translation which contains only this French passage; the position of this sequence in the final translation being a by-product of the maximization operation.

5 Experiment

5.1 Practical Details

The TSRALI database used as TM for extracting sub-sentential translation proposals contains all the debates of the Canadian Parliament (the Hansard), published between April 1986 and December 2001, in all over 100 million words in each language. The French and English documents were aligned at the sentence level using *SFIAL*, a somewhat improved implementation of the method proposed in [12].

The English chunker used for sub-sentential extraction and sequence alignment was composed of three distinct HMM-based taggers ([5]), as discussed in Section 2. All taggers were trained on data from the Penn Treebank, more specifically the training set provided for the CONLL 2000 shared task. Its performance is essentially similar to that of [11].

The architecture of the French chunker used in the sequence alignment procedure is similar to that of the English tagger. The first stage HMM (POS tagger) was trained on a 160 000-word hand-tagged portion of the Hansard. The second

and third stage HMM's were trained on a portion of the *Corfrans* corpus [1], a collection of articles from the French newspaper *Le Monde*, manually annotated for syntax. The overall performance of the French chunker is much worse than the English (we estimate around 70% precision and recall). This is likely attributable to the small size of the training corpus, less than 1 500 sentences in all, compared to over 5 000 sentences for the English chunker.

To train our statistical translation engine, we assembled a bitext composed of 1 639 250 automatically aligned pairs of sentences. In this experiment, all tokens were folded to lower case before training. The inverted translation model (French-to-English) we used in equation 2 is essentially an IBM model 2. The language model is an interpolated trigram trained on the English sentences of our bitext.

The test corpus for our experiments comes from recent transcripts of the Hansard (March 2002), from which we extracted a passage of 1260 sentences, with an average SL length of 19.4 words. For the purposes of this experiment, English was taken as the source language and the French Hansard translation served as the "oracle".

5.2 Example-based Sequences

More than 22 000 queries were successfully submitted to the TSRALI TM; the average length of successful queries was 4.6 words (the longest one was 17-word long). Of the 1 260 sentences, 12% did not generate any successful query, and less than 4% were found verbatim in the TM, which reinforces the claim that sentence-based TM systems are only useful in very specific tasks (revisions of previously translated documents, very repetitive sub-domains, etc.). We excluded these sentences from our test corpus. Successful queries produced more than 1.2 million TL examples, for an average of 56 examples per query.

In [13], we proposed a coarse evaluation of this example extraction process by assuming a user who tries to produce the oracle translation by juxtaposing pieces of the proposed TL examples. Clearly, a system that proposes a multitude of TL examples is more likely to cover the oracle translation, but at the cost of an increased burden for the user. We therefore evaluated this process in terms of *precision* (the quantity of useful TL material proposed) and *recall* (the proportion of the oracle translation covered by the proposed material). These figures were computed under various user-scenarios.

One somewhat unrealistic scenario assumed that the user constructed his translation by cutting and pasting freely (even single words) from the proposed TL examples. This corresponds to the ratios reported in the first line of Table 1. If we only allowed the user to paste entire TL examples, as proposed by our system, the ratios dropped by more than half (line 3 of Table 1). Line 2 in that table is an in-between scenario where we allow the user to grab sequences of at least two words from the TL proposals[2].

[2] All these ratio have been measured after applying the *cover* filter described in [13].

user-scenario	precision	recall	f-measure
cut&paste 1	50.5%	36.6%	42.4%
cut&paste 2	20.8%	27.4%	23.6%
paste-only	14.5%	20.0%	16.8%

Table 1. Results of using TM examples to assist a human translator. The f-measure is the harmonic average of precision and recall.

What these results indicate is that, when the user is only allowed to juxtapose entire pieces of the proposed TL material, one out of every 7 TL sequences retrieved from the TM is useful to produce 20% of the oracle translation.

5.3 Translating

At the time of writing, we have only translated SL sentences that contain at most 30 words (actually more than 90% of the sentences of the test corpus). We tested our translation engine with and without the addition of the extracted TL examples. The performance of our engine was evaluated in terms of *word error rate* (WER) with regard to a single oracle translation. The word error rate is computed as a Levenstein distance (counting the same penalty for *insertion*, *deletion* and *substitution* operations).

With our current decoder implementation, decoding over the full search space becomes impractical as soon as the sentence to translate contains more than 10 words. Therefore we resorted to several pruning strategies (the description of which is irrelevant in this context), yielding a configuration that translates reasonably fast enough without too many detrimental effects on the quality.

We ran seven translation sessions corresponding to different ways of selecting among the TL proposals. The results of these translation sessions are summarized in Table 2. In this table, MERGE-Fn corresponds to a translation session where the n most frequent TL proposals returned by a given query are considered; MERGE-Sn corresponds to a session where the n best-ranked alignments (scored by equation 1) are considered; SMT corresponds to a session in which the statistical engine operated alone without extracted examples. In this experiment, we tested three values of n: 3, 5, and 10.

system	WER	system	WER	system	WER
SMT	68.9%				
MERGE-F3	73.9%	MERGE-F5	74.2%	MERGE-F10	74.2%
MERGE-S3	75.4%	MERGE-S5	74.9%	MERGE-S10	74.4%

Table 2. Translation performance of the SMT engine alone (line 1) and TM examples under different scenarios.

Much to our disappointment, all the attempts to merge the extracted examples into the decoder resulted in an increase of the overall WER (around 5%).

It is not easy to evaluate whether this drop in performance also reflects a significant loss in the quality of the translation. Figure 5 gives two examples where we observed an improvement in WER after merging[3]

These examples call for some comment. The first one illustrates the situation where some words are not known to the statistical engine (here the person name *raymonde folco*), but present in the translation memory. Clearly, this is a situation where our approach should yield noticeable improvements.

The second example may help explain the measured degradations. First, there are some examples that are irrelevant to the translation (*e.g. but no authority/le front des soins*), obviously a bad alignment. Second, some examples are only partially good (*e.g. all the responsibilities / détient toutes les responsabilités*), however, our merging strategy only considered the complete TL examples. Last but not least, there are some SL sequences that we may not want to consider, as for example the sequence *they have* for which we only obtained vague translations. A simple filter could reject such undesired queries and hopefully improve the results.

6 Discussion

Although the results presented in the above evaluation are somewhat disappointing, we feel that there are positive aspects to our experiments. The output of the translation sessions shows many cases where the translation obtained by merging the extracted examples with the decoder clearly improved the results obtained by the engine alone. One possible explanation is that an evaluation based on the WER metric and single oracle translations might not fully do justice to the real contribution of the TM.

Yet, it is legitimate to ask whether the approach presented here does not involve a vicious circle, since both the extraction of TL examples from the TM and the translation engine rely on similar types of statistical translation models, essentially trained and used on the same material. In this regard, it is interesting to note that in the TM matching phase, the statistical models are used to perform "translation analysis", while the decoder does "translation generation". As they currently stand, statistical language and translation models are very crude devices. One of the assumptions underlying this work is that given their inherent weaknesses, the former task (analysis) is easier in practice than the latter (generation).

As Daniel Marcu points out [9], improving SMT outputs with TM examples is only possible insofar as it compensates for the imperfections of existing models and decoders. It supposes that the TM contains good translations for SL sequences, which the decoder would not normally produce, either because of the reduced search-space within which it operates, or because of these translations' low frequency in the training corpus. But whether or not the decoder is able to take advantage of the better translations contained in the TM crucially depends on several aspects.

[3] The full translation sessions are available at
`http://www.iro.umontreal.ca/~felipe/ResearchOutput/AMTA2002`.

SRC ms. raymonde folco (parliamentary secretary to the minister of human resources development , lib .)
REF mme raymonde folco (secrétaire parlementaire de la ministre du développement des ressources humaines , lib .)
MERGE-F3 mme raymonde folco (secrétaire parlementaire de la ministre du développement des ressources humaines .) , [wer=15.7%]
SMT mme UNKNOWN clark (secrétaire parlementaire du ministre des finances et des ressources humaines .) [wer=47.4%]

EXAMPLES *ms. raymonde folco (parliamentary secretary to the minister of human resources development/* `mme raymonde folco ( secrétaire parlementaire de la ministre du développement des ressources humaines`

SRC they have all the responsibilities but no authority .
REF ils ont toutes les responsabilités , mais aucune autorité .
MERGE-F3 ils se faire le tour des responsabilités , mais peu de pouvoirs . [wer=61.5%]
SMT ils ont tous la responsabilité d' emprunt non . [wer=72.7%]

EXAMPLES
all the responsibilities / `détient toutes les responsabilités`
all the responsibilities / `faire le tour des responsabilités`
all the responsibilities / `les communications`
they have / `ils se`
they have / `ils ont réussi`
they have / `ils ont passé`
but no authority / `le front des soins`
but no authority / `, mais peu de pouvoirs`

Fig. 5. Translation outputs and matching examples. SRC designates the SL sentence; REF indicates the oracle translation and EXAMPLES indicates the examples given to the decoder.

First, looking up SL sequences *verbatim* is admittedly a rather simplistic scheme – one we essentially viewed as a starting point to gauge the potential of the approach. Macklovitch and Russell [7] convincingly argue in favor of performing more "linguistically informed" searches, for instance taking inflectional morphology and syntax into account, or dealing with named entities and numerical expressions in a sensible way. In a more general way, much work in EBMT could be of use in a setup such as ours (for example, see [4]).

Also, a close inspection of TL examples reveals that incorrect alignments are often to blame for bad translations. In particular, such imprecise alignments as those in Figure 5 exacerbate the *boundary friction* problem, well-known in EBMT circles. We are currently experimenting with more elaborate alignment techniques.

Both our strategies of selecting TM examples on the basis of their frequency or alignment score typically prevent the decoder from picking low-frequency examples, in favor of more literal ones. Alternative ranking techniques are needed to prevent this kind of systematic behavior. For one thing, we do not currently force the bracketing of SL sequences found in the TM to match that of the

input sequence. This would probably help filtering out irrelevant or "syntactically incompatible" matches. In the same vein, we could favor TM couples that either globally resemble the input sentence, or that have "syntactic similarities" around the boundaries of the matching sequence.

Finally, and as mentioned earlier, there are many more ways of feeding the TM examples to the translation engine. In short, the time to throw in the towel has not yet come.

Acknowledgments

We would like to thank Elliott Macklovitch and George Foster for the fruitful comments they made on this work. The statistical models used in this work were built using software written by George Foster.

References

1. Anne Abeillé, Lionel Clément and Alexandra Kinyon. Building a treebank for French. *International Conference on Language Resources & Evaluation* (LREC), Athens, Greece (2000).
2. Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra and Robert L. Mercer. The Mathematics of Machine Translation: Parameter Estimation. *Computational Linguistics*, 19-2 (1993) 263–311.
3. Ralf D. Brown. Example-Based Machine Translation in the Pangloss System. *International Conference on Computational Linguistics (COLING)*, Copenhagen, Denmark, (1996) 169–174.
4. Michael Carl and Silvia Hansen. Linking Translation Memories with Example-Based Machine Translation. *Machine Translation Summit VII*, Singapore (1999) 617–624.
5. George F. Foster. *Statistical Lexical Disambiguation*. MSc thesis, McGill University, School of Computer Science (1991).
6. Philippe Langlais. Opening Statistical Translation Engines to Terminological Resources. *7th International Workshop on Applications of Natural Language to Information Systems* (NLDB). June 27-28, 2002. Stockholm, Sweden (2002).
7. Elliot Macklovitch and Graham Russell. What's been Forgotten in Translation Memory. *The Association for Machine Translation in the Americas* (AMTA-2000), Cuernavaca, Mexico (2000).
8. Elliott Macklovitch, Michel Simard and Philippe Langlais. TransSearch: A Free Translation Memory on the World Wide Web. *International Conference on Language Resources & Evaluation* (LREC), Athens, Greece (2000) 641–648.
9. Daniel Marcu. Towards a Unified Approach to Memory- and Statistical-Based Machine Translation. *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, Toulouse, France (2001) 378–385.
10. S. Niessen, S. Vogel, H. Ney and C. Tillmann. A DP based Search Algorithm for Statistical Machine Translation. *COLING/ACL* (1998) 960–966.
11. Miles Osborne. Shallow Parsing as Part-of-Speech Tagging. *Proceedings of CoNLL*, Lisbon, Portugal (2002).
12. Michel Simard, George Foster and Pierre Isabelle. Using Cognates to Align Sentences in Bilingual Corpora. *Conference on Theoretical and Methodological Issues in Machine Translation* (TMI), Montréal, Québec (1992) 67–82.
13. Michel Simard and Philippe Langlais. Sub-sentential Exploitation of Translation Memories. *MTSummit-VIII*, Santiago de Compostela, Spain, (2001) 335–340.