Harnessing the Redundant Results of Translation Spotting

Stéphane Huet, Julien Bourdaillet, Philippe Langlais and **Guy Lapalme**
DIRO - Université de Montréal
C.P. 6128, succursale Centre-ville
H3C 3J7, Montréal, Québec, Canada
{huetstep,bourdaij,felipe,lapalme}@iro.umontreal.ca

Abstract

Translation spotting consists in automatically identifying the translations of a user query inside a bitext. This task, when it relies solely on statistical word alignment algorithms, fails to achieve excellent results. In this paper, we show that identifying the translations of a query during a first translation spotting stage provides relevant information that can be used in a second stage to improve the precision of the results. This method is similar to the relevance feedback used in the information retrieval domain to enhance retrieval.

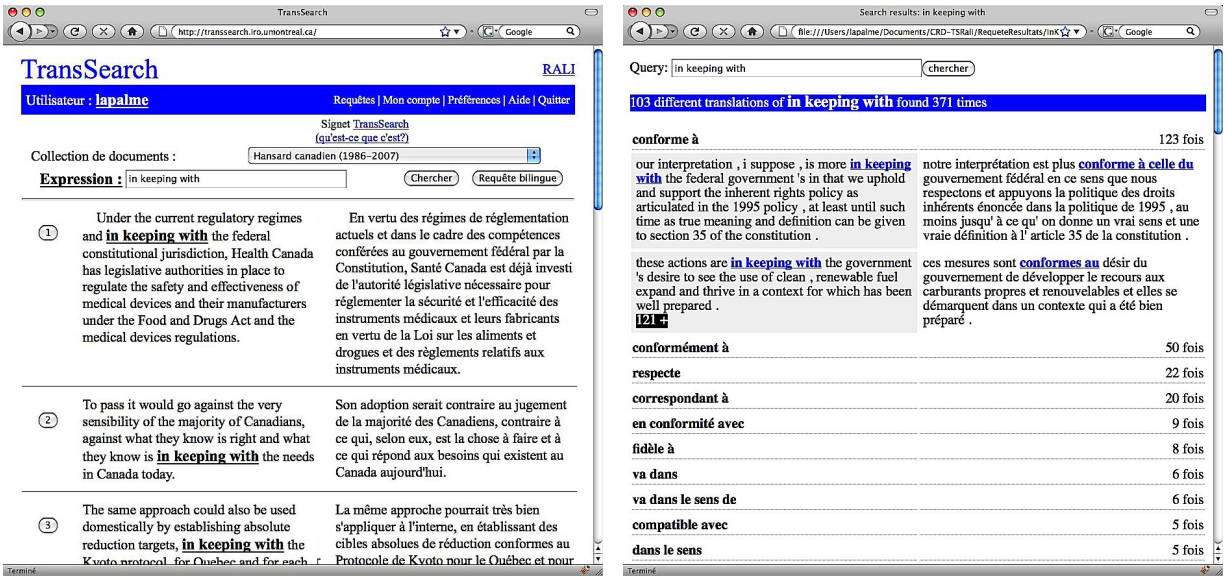
1 Introduction

Although the last decade witnessed an impressive amount of effort devoted to improving the current state of Machine Translation (MT), professional translators still prefer Computer Assisted Translation (CAT) tools, among them *translation memory* (TM) systems and *bilingual concordancers*. Both tools exploit a TM composed of a *bitext*: a set of pairs of units (typically sentences) that are in translation relation. Whereas a TM system is a translation device, a bilingual concordancer is conceptually simpler, since its main purpose is to retrieve from a bitext, the pairs of units that contain a *query* (typically a phrase) that a user manually submits. It is then left to the user to locate the relevant material in the retrieved target units. As simple as it may appear, a bilingual concordancer is nevertheless a very popular CAT tool. In (Macklovitch et al., 2008), the authors report that TRANSSEARCH,[1] the commercial web-based concordancer we focus on in this study, received an average of 177 000 queries a month over a one-year period (2006–2007).

Figure 1(a) shows the first three search results found by the current version of TRANSSEARCH for the English phrase `in keeping with` in a bitext composed of sentences from the Canadian Hansards. Once the system has identified English sentences containing the query, it displays them with the corresponding French sentence. Although the substring of the English sentence can be highlighted easily (here in bold), the corresponding substring in the French sentence is much harder to identify. Professional translators are quick to locate the matches but they must often go through many sentences to find different translations. Currently, the system displays the sentences in reverse chronological order of dates of the documents.

Identifying the matching substring in the other language enables a better display of the results by grouping them as shown in Figure 1(b). To do so, we collect all these substrings for a given query and merge close variants, such as the inflected forms of the same lemmas, according to the procedure described in (Huet et al., 2009). In the displayed example, the user can easily browse the 103 different French translations identified for `in keeping with` among 371 sentence pairs. The most frequent translations are `conforme à`, `conformément à`, `respecte`, `correspondant à`, etc. Clicking on a translation shows the two most recent occurrences and clicking on the number below the second translation displays all of them.

[1] www.tsrali.com



(a) Current system.

(b) Development version.

Figure 1: Results to the query `in keeping with` with the current version (a) and the future version (b) of TRANSSEARCH. This last version groups the translations before the display; the user can thus know right away the whole gamut of translations that were found in the bitext. For the first suggested translation, the two substrings highlighted in the targer parts were considered as variants of `conforme à`, according to the merging process described in (Huet et al., 2009).

Following (Simard, 2003), we call *translation spotting* or *transpotting* the process of identifying the different translations of a user query in a bitext. The first idea that comes to mind is using word alignment, a common task in Statistical Machine Translation (SMT). Unfortunately, merely relying on *maximal* word alignment scores does not give satisfactory results. While in SMT, word alignment is one component of a complete pipeline, in our application, its results are directly visible by the user. In many cases, they are a *a bit off*: the spotted translation may be incomplete, too long or completely miss the right translation.

In (Bourdaillet et al., 2009), each pair of sentences in which a query occurs is aligned individually—as it is done in SMT. But given the fact that all sentence pairs share the substring containing the query, we decided to make use of this fact to release the current *independence* assumption. In this paper, we describe two methods that use the information provided by the translations found in a first stage to refine the results in a second phase before they are displayed to the user. Both methods lead to significant improvements in transpotting quality (see Section 5.3). These methods are similar in principle to the *relevance feedback* concept from the information retrieval domain because they make use of the initial results of a query to gather initial information before retrieving other results in order to refine a second retrieval stage.

This paper is organized as follows. We first describe in Section 2 the translation spotting technique we implemented. Section 3 shows how the outputs of TRANSSEARCH are refined from the results initially obtained and it introduces the two models we designed. We present in Section 4 the data and the metrics used to evaluate the methods. Experiments are reported in Section 5. We conclude our discussion and propose ongoing avenues in Section 6.

2 Transpotting

Transpotting is the task of identifying the word-tokens in a target-language translation that correspond to the word-tokens of a query in a source language. We call *transpot* the target word-tokens automatically associated with a query in a given

pair of units (sentences). For instance in Figure 1(b), `conforme à` and `respecte` are 2 of the 103 transpots displayed to the user for the query `in keeping with`.

2.1 Word Alignment

Word alignment is a key component of the transpotting task. Given a source sentence $S = s_1...s_n$ and a target sentence $T = t_1...t_m$ in translation relation, an IBM-style alignment $a = a_1...a_m$ connects each target token to a source one ($a_j \in \{1, ..., n\}$) or to the so-called NULL token which accounts for untranslated target tokens, and which is arbitrarily set to the source position 0 ($a_j = 0$).

Several word-alignment models are introduced and discussed in (Brown et al., 1993). They differ by the expression of the joint probability of a target sentence and its alignment, given the source sentence. As we do not want to keep the precomputed alignments of each sentence pairs, we decided to use the IBM model 2 which allows our system to quickly manage on-line hundreds of pairs of sentences retrieved for a given query. This model is expressed by:

$$p(t_1^m, a_1^m | s_1^n) = \epsilon \prod_{j=1}^{m} p(t_j | s_{a_j}) \times p(a_j | j, m, n)$$

where ϵ is the length distribution, the first term inside the product is the transfer or lexical distribution and the second one is the alignment distribution.

Given this decomposition of the joint probability, it is straightforward to compute the Viterbi alignment maximizing the quantity $p(a_1^m | t_1^m, s_1^n)$. This computation can be done efficiently in $O(m \times n)$.

Selecting all the target tokens aligned with the words of the query is a straightforward transpotting method. In practice, however, this strategy is error prone and better transpotting algorithm must be considered. Figure 2 illustrates a common error that appears when using only a word alignment algorithm. In this example, the identified transpot for the query `in keeping with` is made of two sequences: `mesure` and `conforme à`. Although it may be necessary to choose a non-contiguous phrase, we noticed that such transpots are usually erroneous.

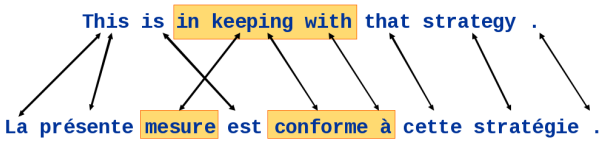


Figure 2: Example of word alignment generated by an IBM model 2 that leads to an erroneous transpot for the query `in keeping with`.

2.2 Transpotting Algorithm

In this work, we implemented a variant of the transpotting algorithm initially proposed by Simard (2003), which shares some similarity with the phrase extraction technique described in (Venugopal et al., 2003). For each pair $\langle j_1, j_2 \rangle \in [1, m] \times [1, m]$, two Viterbi alignments are computed: one between the phrase $t_{j_1}^{j_2}$ and the query $s_{i_1}^{i_2}$, and one between the remaining material in the sentences $\bar{s}_{i_1}^{i_2} \equiv s_1^{i_1 - 1} s_{i_2+1}^n$ and $\bar{t}_{j_1}^{j_2} \equiv t_1^{j_1 - 1} t_{j_2+1}^m$. This method finds the translation of the query according to:

$$\hat{t}_{\hat{j}_1}^{\hat{j}_2} = \underset{(j_1, j_2)}{\text{argmax}} \begin{cases} \max_{a_{j_1}^{j_2}} p(a_{j_1}^{j_2} | s_{i_1}^{i_2}, t_{j_1}^{j_2}) \\ \times \\ \max_{\bar{a}_{j_1}^{j_2}} p(\bar{a}_{j_1}^{j_2} | \bar{s}_{i_1}^{i_2}, \bar{t}_{j_1}^{j_2}) \end{cases}$$

Our implementation of this method when resorting to the IBM model 2 to compute Viterbi alignments has a complexity in $O(m \times n)$. As shown in (Bourdaillet et al., 2009), it significantly reduces the erroneous transpots w.r.t. the use of an IBM 2 word alignment alone.

3 Pseudo Relevance Feedback

Relevance feedback techniques are well studied in the information retrieval domain (Rocchio, 1971; Ruthven and Lalmas, 2003). They rely on human judgments for identifying relevant documents returned in a first stage of retrieval; this information is used for improving a second stage. A variant of this method, known as pseudo relevance feedback, does not require user annotation but assumes that the top ranked documents returned during the first stage are relevant (Croft and Harper, 1979).

We propose to adapt pseudo relevance feedback to the transpotting task. In our case, a first transpotting phase is carried out and the most frequent transpots are considered as relevant. This information is

transpot	frequency
mode de vie	731
du mode de vie	27
façon de vivre	27
style de vie	25
...	
leur mode de vie	9
mode de vie de	8
niveau de vie	7
mode de vie au	7
du mode de vie des	7
la société	7
du style de vie	4
...	
mode de vie qui	1
mode de vie a de	1
nuit au mode de vie	1
bien des gens	1
pratiqué la	1

Figure 3: Subset of the 335 different transpots retrieved for the query `way of life` by the first stage transpotting algorithm.

then used to improve a second stage of transpotting. Overall, we noticed that frequent transpots are likely good translations of a query.

This is illustrated in Figure 3 for the query `way of life`. The correct translation `mode de vie` clearly occurs more frequently than the other transpots. The next candidates also deliver relevant translations, such as `façon de vivre` or `style de vie`. At the end of this list, many transpots, especially hapax ones, correspond to variants of the most frequent translation, like `mode de vie qui`. Furthermore, a proportion of transpots (`bien des gens`, `pratiqué la`) are incorrect.

Using pseudo relevance feedback methods, the first stage will identify the frequent transpots that will be used to correct erroneous transpots during a second stage. We now present two methods we designed based on this principle.

3.1 Procedural Relevance Feedback

Based on the observation that frequent transpots are likely the good ones, we build the set of the most frequent transpots for each query and try to modify the rare transpots into an element of this set. We call this method the procedural relevance feedback, or PRF for short (see Algorithm 1). The decision to consider a transpot as rare is based on two parameters α and β, which respectively fix an absolute and a relative threshold. For example, the values $\alpha = 5$ and $\beta = 0.02$ lead to regard as rare all the transpots that occur less than 5 times and that are found in less than 2 % of all the retrieved sentence pairs. Rare transpots are then modified if a frequent target one is found in the sentence under consideration.

Input: $\mathcal{S}_{\text{in}}$: set of 1st stage transpotted pairs of sentences,
α: absolute threshold,
β: relative threshold
Output: $\mathcal{S}_{\text{out}}$: set of 2nd stage transpotted pairs of sentences
$\mathcal{S}_{\text{rare}} \longleftarrow \{$pair (s,t) associated with a transpot whose frequency in $\mathcal{S}_{\text{in}}$ is $\leq \alpha$ and $\leq \beta \times$ Card($\mathcal{S}_{\text{in}}$) $\}$
$\mathcal{S}_{\text{out}} \longleftarrow \mathcal{S}_{\text{in}} - \mathcal{S}_{\text{rare}}$
$\mathcal{L}_r \longleftarrow$ list of transpots found in $\mathcal{S}_{\text{out}}$ and sorted in decreasing frequency order
foreach *pair* $(s,t) \in \mathcal{S}_{rare}$ **do**
  foreach $r \in \mathcal{L}_r$ **do**
    if r *occurs in* t **then**
      turn the transpot of (s,t) into r
      $\mathcal{S}_{\text{out}} \longleftarrow \mathcal{S}_{\text{out}} \cup \{(s,t)\}$
      break
    end
  end
end

Algorithm 1: PRF method for a given query

3.2 Statistical Relevance Feedback

One drawback of the PRF method is that it relies on a threshold based decision for identifying good transpots during the first stage. We propose an alternative relevance feedback method, named SRF, which computes a *local* statistical transfer model using the transpots found during the first stage, with the hope it can be improved in a second stage word alignment.

To train a local model, we build for each query a parallel corpus made of this query and all the transpots found in the retrieved pairs of sentences during

the first stage. We take for granted that this short parallel corpus contains information that is more specific to the translation of the query than the very large training corpus used to build the main alignment model.

This specific corpus is used to compute the probabilities $P_{\text{loc}}(t_j|s_i)$ of a local transfer model[2] which are linearly interpolated with the probabilities $P_{\text{glob}}(t_j|s_i)$ of the global transfer model initially used by the word aligner algorithm. This idea shares some commonality with the cache model used in language modeling (Kuhn and De Mori, 1990). Since the specific corpus only provides information about the use of the words of the query, the modifications of the transfer model are limited to those words. Therefore, the new probabilities $P_{\text{SRF}}(t_j|s_i)$ used during the second transpotting stage are:

$$\begin{array}{ll} \lambda P_{\text{glob}}(t_j|s_i) + (1-\lambda)P_{\text{loc}}(t_j|s_i) & \text{if } s_i \in q \\ P_{\text{glob}}(t_j|s_i) & \text{otherwise} \end{array} \quad (1)$$

where λ is a coefficient to optimize.

We noticed that the use of the local transfer model tended to extend the initial transpots with grammatical words (determiners, pronouns, prepositions, conjunctions and auxiliary verbs). To overcome this problem, these grammatical words were filtered out before training local models.

4 Evaluation setup

Evaluating transpotting algorithms in a system such as TRANSSEARCH is challenging and is still an open question. It requires a reference corpus large enough to judge the quality of the results and a set of metrics that exhibit the adequacy of the outputs with the goals of the users.

4.1 Reference corpus

To study the behavior of our methods on "real" queries, we extracted from a log file the most frequent English queries that were submitted to TRANSSEARCH. We reserved 247 queries for development purpose (DEV) and considered 2 110 ones for testing our methods (TEST). For each query, up to 5 000 sentence pairs were retrieved from the parallel text of the Canadian Hansards, the largest collection in TRANSSEARCH. For our experiments, we indexed with Lucene[3] a TM aligned at the sentence-level and comprising 8.3 million pairs of French-English sentences. Of these pairs, 3.3 million were used to train the statistical word-alignment model.

The manual evaluation of the transpots suggested for a given pair of sentence is a long and often difficult task. To build a large reference for the numerous sentence pairs we collected in our corpus, we used an in-house bilingual-phrase lexicon we collected over various projects. Among the retrieved pairs of sentences, we kept only those whose source part contained the query and the target part one of the sanctioned translations. This resulted in a reference with 150 400 pairs of sentences with an average of 3.6 translations per query for DEV and 1 416 000 pairs of sentences with 3.9 translations on average for TEST. For instance, 3 reference translations are available for the query `way of life` of Figure 3: `mode de vie`, `genre de vie` and `train de vie`.

4.2 Metrics

The new TRANSSEARCH prototype achieves two related tasks that deserve their own evaluation: the transpotting and the translation tasks. As shown in Figure 1(b), the new version highlights in each displayed sentence pair the words that are the transpots of the query—this corresponds to the transpotting task. From all the substrings matching the query in the target sentences, it suggests different translations—this corresponds to the translation task. In the example of Figure 1(b), 103 different translations were identified (only 10 are displayed in the figure) and we see two sentence pairs where variants of `conforme à` occur. We describe now the metrics we designed to measure these two tasks.

Transpotting The transpotting task evaluates the capacity of an algorithm to retrieve the reference transpot in a target sentence. Following the previous work of Simard (2003), the relevance of a transpot r for a given sentence pair (s, t) can be measured in terms of precision and recall when comparing r

[2] Because the local bitext is very short, training the local model is a very fast operation.

[3] `lucene.apache.org`

with the reference transpot $\hat{r}$. Scores are computed as follows:

$$\text{recall}_{\text{ts}}(s,t) = |r \cap \hat{r}|/|\hat{r}|$$
$$\text{precision}_{\text{ts}}(s,t) = |r \cap \hat{r}|/|r|$$

where $\cap$ returns the longest common contiguous subsequence of tokens between r and $\hat{r}$. To get recall and precision on the overall reference corpus, the scores computed for the sentence pairs are averaged independently for each query; the scores so obtained are then averaged over all the queries. These two levels of average enables us to alleviate the fact that some queries are associated with more numerous sentence pairs.

From the users' perspective, highlighting the transpots enables them to quickly identify the translation of their query in the target sentence, even if a mistake was committed on the boundary of the transpot. Thus, a transpotting algorithm that seldom highlights an erroneous word is more interesting than another that does not miss words but suggests too large transpots. For this reason, we mainly focus on the precision metrics for the transpotting task.

Translation The translation task reflects the ability of an algorithm to find the different translations of a given query in the retrieved pairs of sentences. This task is essential for our system since the results must be displayed with a limited size in the screen, which requires them to be both correct and diversified. To measure this, we compare, for each query q, the set of transpots retrieved (HYP_q) with the set of reference translations (REF_q). Recall measures the number of different reference translations found by the transpotting algorithm for a given query, *i.e.* it evaluates the variety of the suggested translations. Precision indicates the proportion of correct translations. As both metrics seem to be equally relevant in the context of our application, we employ the F-measure as a criterion when optimizing our relevance feedback methods for the translation task.

The following ratios are finally computed for each query q:

$$\text{recall}_{\text{tl}}(q) = |\text{HYP}_q \cap \text{REF}_q|/|\text{REF}_q|$$
$$\text{precision}_{\text{tl}}(q) = |\text{HYP}_q \cap \text{REF}_q|/|\text{HYP}_q|$$

where $\cap$ stands for the usual set intersection. This means that a transpot $r \in \text{HYP}_q$ must match exactly one of the reference translation $\hat{r} \in \text{REF}_q$. This makes this task clearly harder than the transpotting one, for which partial matches are credited.

Similarly to the transpotting task, recall and precision are deduced at the level of the reference corpus by averaging values computed for each query.

5 Experiments

In this section, we present a series of experiments we conducted in order to show the interest of relevance feedback for transpotting. Due to the difficulties of evaluation of our application, we are still currently exploring the best ways to evaluate the quality of the results of TRANSSEARCH. In particular, it is still not clear whether the transpotting task should be favored w.r.t. the translation one. That is why two systems were built by optimizing independently the relevance feedback models for these two tasks.

5.1 Optimization for PRF

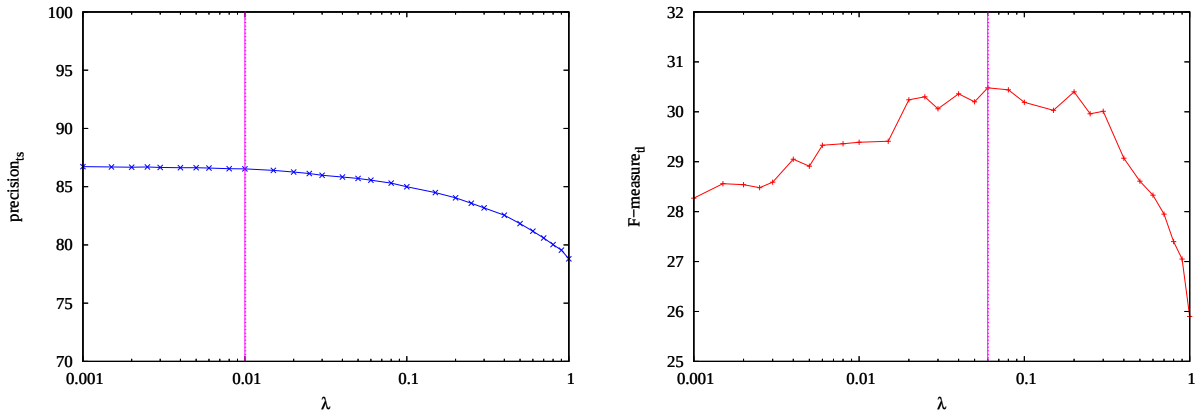
The two threshold values of the PRF method were optimized on the DEV corpus. This optimization was carried out a first time w.r.t. the precision for the transpotting task, and a second time w.r.t. the F-measure for the translation task. For this first task, α and β are respectively fixed to 300 and 0.1. This means that the transpots that are found in less than 300 pairs of sentences and 10 % of the pairs retrieved for the corresponding query are considered as rare. For the second task, these values are set to 100 and 0.03.

5.2 Optimization for SRF

The use of SRF for transpotting requires the optimization of the parameter λ that controls Equation 1. Figure 4 presents how the evaluation metrics vary on the DEV corpus under a logarithmic scale according to λ. For the transpotting task, Figure 4(a) exhibits a plateau in $\text{precision}_{\text{ts}}$ around the value $\lambda = 0.01$ before decreasing. We retained this value as the best one for the transpotting task. For the translation task, the maximum value for $\text{F-measure}_{\text{tl}}$ is reached on Figure 4(b) between 0.02 and 0.3. λ was eventually fixed to 0.06 for this task.

5.3 Evaluation of Relevance Feedback Methods

Table 1 presents the results obtained with SRF on TEST, for some values of λ selected on the DEV cor-



(a) Optimization for the transpotting task.

(b) Optimization for the translation task.

Figure 4: Optimization on DEV of λ according to $\text{precision}_{\text{ts}}$ for the transpotting task (a) and according to $\text{F-measure}_{\text{tl}}$ for the translation task (b). The vertical lines present the optimal λ values retained for SRF for each task.

pus. $\lambda = 0.01$ and $\lambda = 0.06$ are the values optimized respectively for the transpotting and translation tasks. $\lambda = 1$ corresponds to the case where the local transfer model is not used, and therefore SRF remains equivalent to transpotting without relevance feedback. Besides, $\lambda = 0.5$ is the lowest value which allows for the improvement of all the metrics on DEV. On the TEST corpus, this last configuration still leads to the same observation, even if the enhancements in recall are smaller.

Table 2 shows the results of the 3 transpotting methods (without relevance feedback, with PRF and with SRF) measured on TEST. For the transpotting task, these values exhibit an improvement in terms of $\text{precision}_{\text{ts}}$ for both PRF and SRF, with SRF being 2 points superior to PRF. A detailed examination of transpotting results shows that SRF reduces the number of too long transpots, as well as missed transpots.

For the translation task, PRF significantly outperforms SRF with a superiority of 18 points in $\text{precision}_{\text{tl}}$. Results also show that $\text{precision}_{\text{tl}}$ is increased by 22 points w.r.t. the method without relevance feedback, while $\text{recall}_{\text{tl}}$ decreases by 10 points. A closer examination of the transpots returned by the PRF method shows that the large gain in $\text{precision}_{\text{tl}}$ is explained by a reduction of the number of returned transpots. Since only the most frequent transpots are used for relevance feedback, this method reduces the number of transpots and lowers diversity. This has two consequences: some bad transpots are removed (for example, from Figure 3: `nuit au mode de vie`, `tout un mode de vie`), which increases $\text{precision}_{\text{tl}}$, but it occasionally discards some good ones which decreases $\text{recall}_{\text{tl}}$.

6 Conclusion and future works

In this paper, we described two methods based on relevance feedback for enhancing the transpotting algorithm embedded in TRANSSEARCH. We have shown that relevance feedback clearly improves precision scores for the transpotting and the translation tasks, two metrics we consider as important in our application. More specifically, the SRF method has a better ability to spot a translation in a given pair of sentences, while the PRF method tends to reduce the number of suggested translations to a query.

Albeit these results are encouraging, we are facing an evaluation problem inherent to interactive applications. Ultimately, this will involve the development of test cases with real users of the application.

We are currently experimenting the use of HMM alignment models (Vogel et al., 1996) in our transpotting algorithms as alternative to the IBM model 2. Finally, our approach might be applied to the more general task of acquiring a phrase table in SMT. As a matter of fact, a phrase in such a system plays a similar role to a query in our setting. Therefore, ap-

λ	0		0.01		0.06		0.5		1	
score	rec	prec	rec	prec	rec	prec	rec	prec	rec	prec
transpotting	72.6	85.6	77.6	**85.7**	80.5	84.8	**83.9**	81.3	83.7	78.7
translation	61.9	18.0	74.9	**18.5**	79.9	18.2	**82.6**	16.2	**82.6**	14.7

Table 1: Scores of the SRF method for the transpotting and translation tasks according to the parameter λ on the TEST corpus. $\lambda = 1$ corresponds to a system without relevance feedback.

task	Transpotting		Translation	
score	recall	precision	recall	precision
w/o relevance feedback	**83.7**	78.7	**82.6**	14.7
PRF	83.5	83.5	72.3	**36.9**
SRF	77.6	**85.7**	79.9	18.2

Table 2: Scores for the transpotting and translation tasks for the three transpotting algorithms, the parameters being optimized independently for each task.

plying relevance feedback to the phrase acquisition process is an interesting prospect.

Acknowledgements

This research is being funded by an NSERC grant in collaboration with Terminotix.[4] The authors would like to thank Fabrizio Gotti and Elliott Macklovitch for their contribution to this work.

References

J. Bourdaillet, S. Huet, F. Gotti, G. Lapalme, and P. Langlais. 2009. Enhancing the bilingual concordancer TransSearch with word-level alignment. In *22nd Conference of the Canadian Society for Computational Studies of Intelligence*, pages 27–38, Kelowna, Canada.

P. Brown, V. Della Pietra, S. Della Pietra, and R. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19(2):263–311.

W. Croft and D. Harper. 1979. Using probabilistic models of information retrieval without relevance information. *Journal of Documentation*, 35(4):285–295.

S. Huet, J. Bourdaillet, and P. Langlais. 2009. TS3: an improved version of the bilingual concordancer TransSearch. In *13th Annual Conference of the European Association for Machine Translation (EAMT)*, pages 20–27, Barcelona, Spain.

R. Kuhn and R. De Mori. 1990. A cache-based natural language model for speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(6):570–583.

E. Macklovitch, G. Lapalme, and F. Gotti. 2008. TransSearch: What are translators looking for? In *18th Conference of the Association for Machine Translation in the Americas (AMTA)*, pages 412–419, Waikiki, Hawai'i, USA.

J. Rocchio, 1971. *Relevance feedback in information retrieval*, chapter 14, pages 313–323. Prentice-Hall Inc.

I. Ruthven and M. Lalmas. 2003. A survey on the use of relevance feedback for information access systems. *Knowledge Engineering Review*, 18(2):95–145.

M. Simard. 2003. Translation spotting for translation memories. In *HLT-NAACL 2003 Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and beyond*, pages 65–72, Edmonton, Canada.

A. Venugopal, S. Vogel, and A. Waibel. 2003. Effective phrase translation extraction from alignment models. In *41st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 319–326, Sapporo, Japan.

S. Vogel, H. Ney, and Tillmann C. 1996. HMM-based word alignment in statistical translation. In *16th Conference on Computational Linguistics*, pages 836–841, Copenhagen, Denmark.

[4] www.terminotix.com