Integrating Bilingual Lexicons in a Probabilistic Translation Assistant

Philippe Langlais, George Foster, Guy Lapalme

RALI / DIRO
Université de Montréal
C.P. 6128, succursale Centre-ville
Montréal (Québec)
Canada, H3C 3J7
www-rali.iro.umontreal.ca

Abstract

In this paper, we present a way to integrate bilingual lexicons into an operational probabilistic translation assistant (TransType). These lexicons could be any resource available to the translator (e.g. terminological lexicons) or any resource statistically derived from training material. We describe a bilingual lexicon acquisition process that we developped and we evaluate from a theoretical point of view its benefits to a translation completion task.

Keywords

Translator assistant, probabilistic translation, bilingual lexicon acquisition, bilingual phrasal alignment

Introduction

Translation needs are growing faster than machine translation (MT) technology improves. Therefore, there are more and more situations where MT is just not an acceptable solution, especially when high quality translation is required. This statement is encouraging for (computational) linguists since people realize that despite good translation programs available on the web, more effort must still be invested in MT research. In the meantime, we believe that turning to alternatives to fully automatic translation is a challenging but promising approach.

Among the tools that may make the translator's difficult task a little easier, Foster et al.(1997) have developed TRANSTYPE, a system in which a translation emerges from a series of alternating contributions by the human and the machine. The machine's contributions are basically proposals for parts of the target text, while the translator's can take many forms, including pieces of target text, corrections to a previous machine contribution, hints about the nature of the desired translation, etc. In all cases, the translator remains fully in control of the process: the machine must work within the constraints implicit in the user's contributions, and he or she is free to accept, modify, or completely ignore its proposals.

TRANSTYPE takes the form of a specialized text editor (see Figure 1). Embedded within this editor is a non-intrusive machine translation engine which can provide, at any point of the translation, a ranked list of units (words or sequences of words) that the translator is likely to type. The editor allows for the easy insertion of anyone of these units at a keystroke. These completions are computed according to a translation model and a language model that both take into account the translation already typed. Within such a scenario, we have investigated the possibility of integrating bilingual lexicons and report on our results in this paper. These lexicons could be any resource available to the translator (e.g. terminological lexicons) or any resource statistically derived from training material.

In the next section, we give a brief overview of the TRANSTYPE system and its evaluation by translators. In section three, we describe the strategy we devised to automatically extract bilingual lexicons from training material. In the fourth section, we describe the way we integrated these automatically acquired lexicons within TRANSTYPE's completion mechanism, followed by a performance evaluation of this integration. We sum up by drawing the conclusions of our experiment.

An overview of TRANSTYPE

The core system

The core of TRANSTYPE is a completion engine which comprises two main parts: an *evaluator* which assigns probabilistic scores to completion hypotheses, and a *generator* which uses the evaluation function to select the best candidate for completion.

The evaluator is a function $p(t|t', s)$ which assigns to each target-text unit t an estimate of its probability given a source text s and the tokens t' which precede t in the current translation of s. The approach to modeling this distribution is based to a large extent on that of the IBM group (Brown et al., 1993), but owing to the real-time constraints of our application, it differs in one significant aspect: whereas the IBM model involves a "noisy channel" decomposition, TRANSTYPE uses a linear combination of separate predictions from a language model and a translation model. The language model itself is a classical trigram interpolated model, while the translation model represents a slight modification of an IBM2 model. The two are combined as follows.

$$p(t|t', s) = \underbrace{p(t|t')\ \alpha(t', s)}_{\text{language}} + \underbrace{p(t|s)\ [1 - \alpha(t', s)]}_{\text{translation}} \qquad (1)$$

where $\alpha(t', s) \in [0, 1]$ are context-dependent interpolation coefficients.

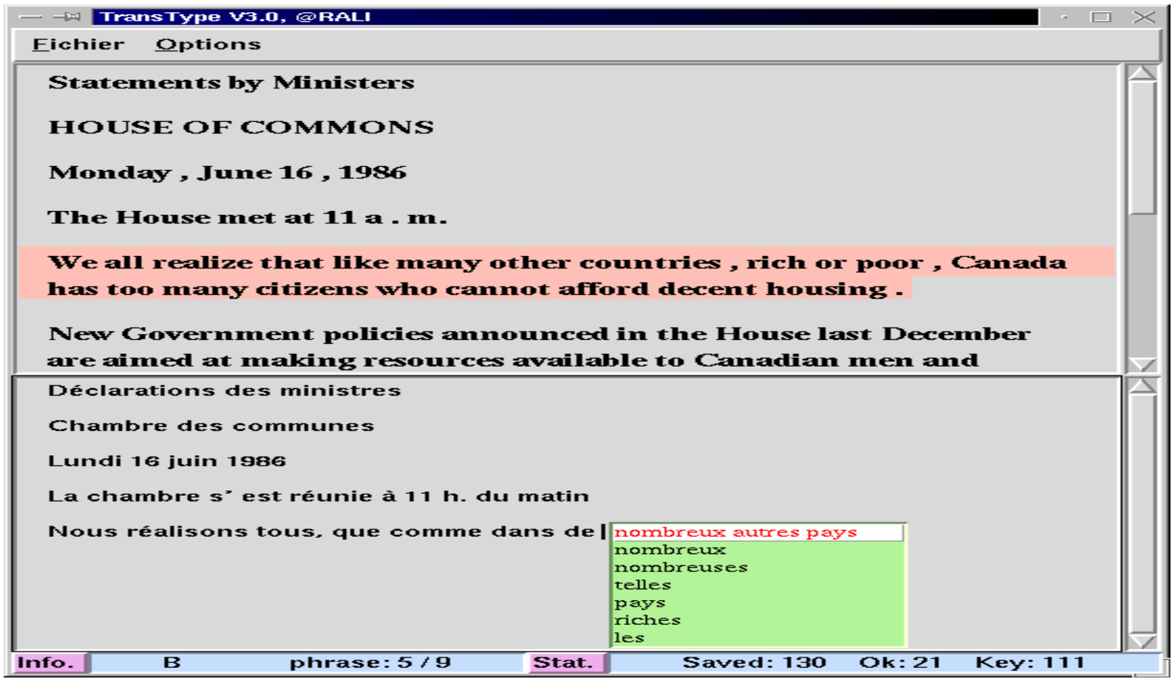

Figure 1: An example of interaction in TRANSTYPE with the source text in the top half of the screen. The target text is typed in the bottom half, with suggestions provided by the menu that appears at the insertion point.

Evaluation

An implementation of TRANSTYPE which allows the completion of words was evaluated in two ways. In a theoretical evaluation, a simulated user generates character by character the target part of a test corpus, accepting as soon as it is helpful the first completion provided by TRANSTYPE. It was shown that under this scenario, a user could save about two thirds of the keystrokes needed to produce a translation (Foster et al., 1997).

An *in-situ* evaluation involving ten translators who were asked to translate the same text using TRANSTYPE has also been carried out (Langlais et al., 2000b). Some interesting observations emerged which motivate the present study. Only one translator actually managed to translate faster using TRANSTYPE; this suggests that even in a very simple scenario, target-text mediated interactive translation is at least viable. Lack of training time is probably one reason for these otherwise disappointing results. The fact that real users do not systematically watch the screen when typing may also account for part of the problem.

A qualitative survey revealed that most users (actually nine out of ten) liked TRANSTYPE and would be eager to try it in their work. However, they expressed the desire for a version of the system which would be able to suggest completions beyond the word level.

From informal discussions with translators, we concluded that an important part of the translation process relies on lexicons. Actually, one of a translator's first tasks is often terminological research; and many translation companies employ specialized terminologists. The need for specialized lexicons becomes even more crucial in a machine translation application. Beyond the infrequent cases where, in a given thematic context, a word is likely to have a clearly preferred translation (*e.g.* bill/*facture vs* bill/*projet de loi*), lexicons are often the only means for a user to influence the translation engine. As TRANSTYPE is deeply user-oriented, we feel it would be a desirable extension to the system if users were allowed to introduce specific lexicons. This extension can be seen as a first step toward an adaptative version of TRANSTYPE, which is a very challenging issue that we hope to study further.

Automatic acquisition of lexicons from bilingual corpora

Many studies have addressed the problem of automatically acquiring bilingual lexicons (see for instance (Melamed, 1997; Ohomori and Higashida, 1999; Rapp, 1999; Tanaka and Matsuo, 1999; Jacquemin, 99) for recent ones). These studies are by nature difficult if not impossible to compare. Therefore, we investigate a simpler version of the approach described in (Langlais et al., 2000a) that basically involves three steps. First, we identify monolingually salient units using various statistical metrics and/or filters. Second, we group together in our training corpus words which belong to the units selected in the previous step in order to train a new translation model where both words and sequences of words (units hereafter) are linked across languages. Last but not least, we clean up the resulting model by filtering out dubious associations.

The motivation behind this process is essentially practical. We do not believe that separating the identification of salient units from their bilingual mapping is a promising approach. It would be much better to look for a translation model which allows $n : m$ associations. Of course, the problem for such an approach is to find a way to cope with the well known malediction of multidimensionality (any group of source words being potentially associated to any target group one). More advanced models such as IBM models 3 to 5 (Brown et al., 1993) which permit $1 : n$ associations may be seen as a step in this direction. More recently, the 2-stage model described by Och (Och and Weber, 98; Och et al., 99) seems to be another alternative — at least in a task comparable to the Verbmobil one — as it allows certain hidden structural information to be captured.

Identifying monolingual salient sequences

Distributional filters

The literature abounds in measures that help to decide whether words that happen to co-occur are linguistically significant or not. In this study, we rated the coherence of any sequence of words seen in a training corpus by means of two measures: a likelihood-based one (Dunning, 93) and an entropy-based one (Shimohata et al., 1997). Observing the output produced by these methods, it is immediatly apparent that neither metric guarantees that the best ranked units are those that we would ourselves manually select as salient. In particular, it is clear that many sequences overlap; which further complicating the selection process. For this reason, we applied a cascade filter to remove well-rated but non-salient units. Below, we report on the results of a filtering process (called DIST) which removes any sequence seen only once or having a likelihood ratio lower than 5.0; DIST also removes some sequences that overlap with others, according to their entropy score.

Linguistically motivated filters

In a second approach to salient sequence selection, we tried several linguistically motivated filters that make use of regular expressions defined on part of speech (POS) tags obtained from a tagger. We experimented with several such filters, but report on the one that yielded the best results (namely SNP, for simple noun phrase). More precisely, we filter out any sequence of words that does not match a regular expression which recognizes any sequence composed of one or more articles, numbers, common or proper nouns, adjectives, and passive or progressive verbal forms (a few constraints were empirically added to this passive regular expression to improve the trade-off between precision and recall in this noun phrase identification task.)

Mapping units between languages

Mapping units across the two languages first requires the grouping into units of the tokens in our training corpus, on the basis of the unit lexicons identified in the previous stage. This step, although easy in principle, conceals rather difficult problems. To begin with, different salient units may contain sequences that partially overlap, even under stringent filtering constraints, and may lead to erroneous tokenizations.

To get around the tokenization problem, we use a dynamic programming scheme optimizing a length-based measure ($\mathcal{C}_l$) over the full sentence w_1^I, as described in equation 2. The segmentation is found by keeping track of the indexes that yield the highest $B(I)$ value. The evidence in favor of this criteria compared to others we tried is not overwhelming, but this is the criteria which empirically yielded to the best results. An exemple of the output of this process is reported in Table 1 for a pair of sentences from the Hansard corpus.

$$B(i) = \begin{cases} 0 \;\; if \;\; i = 0 \\ \underset{I \in [1,i[\;,\; w_{i-I}^{i} \in \mathcal{S}}{\operatorname{argmax}} \begin{pmatrix} \mathcal{C}_l(w_{i-I}^{i}) \\ + \\ B(i-I-1) \end{pmatrix} \end{cases} \tag{2}$$

$$\text{with: } \mathcal{C}_l(w_i^j) = \begin{cases} 0 & if \;\; j <= i \\ j - i + 1 & else \end{cases}$$

SRC: *from time to time , · mr. speaker , · the rcmp · launches · investigations · in canada*
TGT: de temps à autre , · monsieur le président , · la gendarmerie royale du canada · fait · des enquêtes · au canada .

Table 1: Output of the segmentation process for a pair of sentences from the Hansard corpus. The segments are separated by the separator · .

More importantly, there is no guarantee, even if we properly tokenize, that the monolingual groups of words will match across the two languages. For the kind of texts we used in this study, this assumption is however, not too compromising.

Finally, mapping the identified units (tokens or sequences) to their equivalents in the other language is achieved by training a new translation model (IBM 2) using the EM algorithm as described in (Brown et al., 1993).

Tidying up the models

At this stage of the process, we obtain a unit model (M_u) which is fairly noisy, in part because of the reasons explained above, in part because grouping words together also reduces the number of times those particular words occur in isolation, thus lowering the accuracy of their association through the training process.

This makes it worthwhile to filter out spurious units using a word-to-word model M_w (for example, the core model used within TRANSTYPE). We therefore applied an algorithm which basically removes any association of two units, the source words of which are not well associated with the target words, under the word model, and vice versa.

The reduction in the total number of parameters obtained by means of this filter can be very high, depending on the values of the few parameters that control the process. For instance, the SNP model described above inityally produced 10,038,770 pairs of units. Filtering these by only considering the 20-best translations of each source word (according to the word model) that have a probability higher than 0.05 reduces the number of admissible paired units to 50,000, which constitutes a reduction by a factor of 200.

Of course, the more we filter a model, the more we lower its potential coverage. Table 2 gives a few associations generated by an SNP filtered model. A quick glance confirms that the associations are fairly correct. Some of them are compositional (such as *rights of women*/droit des femmes), many others are not. Several associations may be only partly correct such as *boom*/explosion démographique, although we may need the context to decide with certainty.

Application-independent evaluation

In order to gauge the quality of the automatically acquired associations, we asked three judges to review a random selection of 1000 source units with 1135 target associations, and to distinguish those that they felt were good, bad and partially correct. We did not provide judges with a clear definition of these terms. At the time of this writing, only one judge had gone through all one thousand source units.
Over the 1135 associations, this judge evaluated 49 as bad (4.3%), 108 partially good (9.5%), while all the others were marked as good. Around 70% of the bad associations could have been avoided, as they resulted from a bug in our post-filtering stage. It is also worth noting that in 31 cases (around 20% of the non perfect associations), the judge felt the need to see additional context in which the associations occurred. Considering that partially good associations remain useful within an application like TRANSTYPE, these results suggest a fairly high precision rate for our lexicon acquisition process.

Plugging lexicons into TRANSTYPE

Collecting lexicons (automatically or not) is rarely an end in itself; for this reason, it makes sense to evaluate the quality of a bilingual lexicon through the tasks the lexicon is designed for. TRANSTYPE lends itself perfectly to this sort of evaluation, since it is a strongly user-oriented prototype and since real users suggested that being able to integrate user lexicons within TRANSTYPE would be an attractive benefit.
In the following experiment, we simply ignore the probability attached to each pair within a unit model, thus considering a unit model as a pure bilingual lexicon. This in fact corresponds to a real situation, in which a user would provide TRANSTYPE with a personal lexicon. The remainder of this section describes how we integrated of this non-probabilistic resource within the probabilistic framework of TRANSTYPE.
To understand this integration, we need to briefly sketch how TRANSTYPE works. The first step consists in computing, once a source sentence is selected by a user, a set of words which are likely to occur in the translation of that sentence. We call this set the **active vocabulary**. Foster et al. (1997) has shown that using an IBM1-like model to compute the 500 most likely words yields an active vocabulary with an average coverage of about 96%[1]. The second step involves – in turns – the interaction of the user and TRANSTYPE's generator; which role is to identify the words in the active vocabulary which match the current prefix (possibly empty) that the user has typed and to pick the best candidate proposed by the evaluator.
Because TRANSTYPE has a very simple decoder (see equation 1) in which a new prediction does not depend on any of the previous decoder states, it turns out to be fairly easy to integrate non-probabilistic resources such as lexicons in the process. In fact, all we have to do is: 1) extend the active vocabulary with those units belonging to the lexicon which are likely to occur in the translation; and 2) provide the evaluator with a way to rate those units.

[1]This step is fast enough so that a user won't notice it on a recent enough computer.

Extending the active vocabulary

If we assume that the lexicon we want to integrate is nearly noiseless (we saw in the previous section that this is a reasonable assumption), then any target unit associated in our lexicon with a source unit which is part of the sentence under translation is potentially a good candidate. Therefore it can be safely added to the active vocabulary.

Rating units

The only question that remains to be settled is how to rate a given unit belonging to the active vocabulary. Our implementation is based on the idea that predicting a unit would be greatly simplified if we knew exactly which part of the source sentence is under translation. In practice, we do not explicitly have such information; however, we do know the contribution of each source word the sentence being translated (s_1^n) to the prediction of a given target word (t_j) at the target position j. In the implementation of our translation model, and following Brown et al. (Brown et al., 1993), we have:

$$p(t_j|s_1^n) = \sum_{i=0}^{n} t(t_j|s_i)a(i|j,n) \quad (3)$$

where $t(t_j|s_i)$ stands for the transfer probability (that is, the probability that the word t_j is the translation of s_i), and $a(i|j,n)$ stands for the so-called alignment probability (here, the probability that a source word at position i will be associated with the target word at position j, knowing the number of words n of the source sentence under translation).
From the individual contributions $t(t_j|s_i)a(i|j,n)$, some information is available which can help to track the source portion of the sentence being translated. In the present study, we applied the following heuristic: if one source token s_d dominates the sum of equation 3, then we can assume that if the user wants to type the target word t_j, this is because he or she wants to translate the source word s_d. Therefore, if this word lies within a source unit belonging to the lexicon, it is likely that the user will type one of the target associations which belong to the active vocabulary.
We control the validity of this heuristic via a single threshold which fixes the minimum value of the ratio of the next best source contribution to the best one. We found experimentally that a ratio of more than 0.8 often allows us to determine the source segment under translation.
Once we have decided, using the word model, that a target unit should be proposed, we merely have to favor the unit against its first word by adding to the word probability a very small quantity that will not disturb the relative ranking between words. By so doing, however, we no longer

boom → prospérité,0.32 essor,0.27 explosion démographique,0.2 explosion,0.11 vague de prospérité,0.11	
fbdb → banque fédérale de développement,1	
rights of women → droits des femmes,1	
canadian aviation safety board → bureau canadien de la sécurité aérienne,1	
office of the superintendent of financial institutions → bureau du surintendant des institutions financières,1	
newfoundland unemployment → taux de chômage à terre-neuve,1	
small craft harbours → ports pour petits bateaux,0.53 ports pour petites embarcations,0.47	
airline industry → industrie du transport aérien,0.73 secteur du transport aérien,0.13 industrie aérienne,0.13	
food processing industry → secteur de la transformation des aliments,1	
ordinary canadians → canadiens ordinaires,0.72 canadiens moyens,0.19 simples canadiens,0.082	

Table 2: Excerpt of a filtered unit translation model trained on nominal groups (SNP). See the full trace of this model at *www-rali.iro.umontreal.ca/ttype-unit.html*. Note that *fdbd* is an acronym for *Federal Business Development Bank*, for which the translation in our training corpus is almost always the one reported.

have a probabilistic engine, since the scores of all the possible completions do not sum to unity. But because of our decoding strategy, this does not pose a major problem.

Trace of a translation session

To illustrate the full process, we provide in Table 3 a one-sentence session using a lexicon containing the associations produced by the filtered SNP model for which we have removed the probabilities. This session is fairly instructive and warrants some explanation. The source sentence to translate is *I shall return to this point in a few moments*, in which only one words group is found in the lexicon (*few moments*) with three likely translations (quelques minutes, quelques instants and quelques moments). Before the user types anything, TRANSTYPE proposes the target word Je, this is what the user expected, and therefore he accepts this proposal (which is indicated by a + in the second column).

The second token proves more problematic and clearly shows the weakness of mixing the predictions of the language and the translation models. The machine's first proposal is le, which is not the word the user is looking for; thus he is forced to type its first letter. TRANSTYPE adjusts to the userls input by proposing in turn several forms of the word retour (return).

The session ends with TRANSTYPE proposing several target units as likely translations for the source unit *few moments*. Actually, although all of the translations proposed by TRANSTYPE are good ones, the one which the translator decided to use is the last TRANSTYPE proposed. This suggests that evaluating TRANSTYPE on a single translation of a given source text is not really fair, especially within the unit lexicon scenario.

Evaluation

The training corpus

To train our unit models, we used a segment of the Hansard corpus consisting of 174,200 pairs of sentences, totaling 33,000 English forms and 43,000 French ones. About a third of these forms occur only once in the corpus.

The test corpus

We ran a theoretical evaluation of TRANSTYPE by counting the number of keystrokes saved by a user who carefully observes every completion and accepts the first one that corresponds to the associated target sentence. For this evaluation, we randomly selected 1000 sentence pairs from the Hansard corpus, none of which were used in the training.

TRANSTYPE's task was to produce verbatim the target sentence, given the source one. We report two measures of the number of keystrokes saved: **first** correspond to a scenario where only one completion (the best according to the generator) is proposed at a time; and **7-best**, which is the actual way TRANSTYPE is implemented. In the latter, a pop-up menu proposes the seven most likely completions at a given time and the user selects (at a cost of one keystroke[2]) the longest unit which matches the one he is looking for.

Results

The results of our test are summarized in Table 4. For comparison purposes, we also report the results of a baseline approach which proposes no unit. Notice, first of all, that integrating a lexicon into TRANSTYPE slightly improves on the baseline, where the lexicon we automatically extracted focused on noun phrase only. The improvement is very modest, since only 184 target sequences were accepted — under the first scenario — over a thousand pair of sentences. It may be helpful, however, to compare to the overall coverage of the lexicons we obtained. In the SNP-lexicon, there are only 470 target units which altogether appear in the test corpus. This is not anormalous, since user-lexicons are also likely to have poor coverage. Furthermore, as we mentioned above, there are many cases where the predictions made are correct, although they do not correspond exactly to the one that was used by the translator in the test corpus.

Discussion

In this paper, we have described a way to automatically acquire bilingual lexicons based on simple distributional properties of n-grams and on simplistic linguistic knowledge. We have shown that, using a fairly simple filtering method, we can obtain lexicons that have a fairly high level of precision. Evaluating such lexicons is slightly more difficult.

In the second part of the paper, we have described how they have been integrated within TRANSTYPE's completion task. The main conclusion of this task-oriented eval-

[2]For instance by a mouse click.

Source sentence:	*I shall return to this point in a few moments*	
Target sentence:	Je reviendrai sur ce point dans quelques moments	
In the lexicon:	*few moments* → quelques minutes/ quelques instants/ quelques moments	

target tokens	typed	best completions in turn
Je	+	/Je
reviendrai	revi+	/le · r/etour · re/venir · rev/iens · revi/endrai
sur	+	/sur
ce	+	/ce
point	+	/point
dans	d+	/de · d/ans
quelques	que+	/le · q/uelques instants · qu/elques minutes · que/lques moments
moments	-	

Table 3: A one-sentence session illustrating the completion tasks. The first column indicates the target words the user is expected to produce. The next two columns indicate respectively the prefixes typed by the user and the completions made – in turn – by the system under a lexicon-completion task. + indicates the acceptance key typed by the user. A Completion is denoted by α/β where α is the typed prefix and β the completed part. Completions for different prefixes are separated by $\cdot$. See *www-rali.iro.umontreal.ca/ttype-proto.en.html* for an animated screen dump of a short translation session.

	first scenario		7-best scenario	
model	Spared (%)	$nb.$	Spared (%)	$nb.$
SNP	55.82	184	64.57	354
DIST	54.74	335	64.04	1049
baseline	55.78	—	64.38	—

Table 4: Results of TRANSTYPE translation sessions on a test corpus consisting of 1000 pair of sentences. $Spared$ is the percentage of keystrokes saved over the session; and nb is the number of sequences that have been proposed during the session.

uation is that lexicons do improve the performance of TRANSTYPE and are therefore of benefit to the user, although this last statement has yet to be confirmed in further user evaluations.

References

Peter F. Brown, Stephen A. Della Pietra, Vincent Della J. Pietra, and Robert L. Mercer. 1993. The mathematics of machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–312, June.

Ted Dunning. 93. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74.

George Foster, Pierre Isabelle, and Pierre Plamondon. 1997. Target-text Mediated Interactive Machine Translation. *Machine Translation*, 12:175–194.

Christian Jacquemin. 99. Syntagmatic and pragmatic representations of term variation. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 341–348, College Park, Maryland.

Ph. Langlais, G. Foster, and G. Lalpalme. 2000a. Unit completion for a computer-aided translation typing system, applied natural language processing. In *Applied Natural Language Processing (ANLP)*, Seattle, Washington, May.

Ph. Langlais, S. Sauvé, G. Foster E. Macklovith, and G. Lapalme. 2000b. Evaluation of transtype, a computer-aided translation typing system: A comparison of a theoretical- and a user- oriented evaluation procedures. In *Second International Conference On Language Resources and Evaluation*, pages 641–648, Athens, Greece, June.

I. Dan Melamed. 1997. Automatic discovery of non-compositional coumpounds in parallel data. In *Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing*, pages 97–108, Providence, RI, August, 1st-2nd.

Franz Josef Och and Hans Weber. 98. Improving statistical natural language translation with categories and rules. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*, pages 985–989, Montréal, Canada.

Franz Josef Och, Christoph Tillman, and Herman Ney. 99. Improved alignment models for statistical machine translation. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 20–28, College Park, Maryland.

Kumiko Ohomori and Masanobu Higashida. 1999. Extracting bilingual collocations from non-aligned parallel corpora. In *8th International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 88–97, Chester, England, August 23 - 25.

Reinhard Rapp. 1999. Automatic Identification of Word Translations from Unrelated English and German Corpora. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 519–526, College Park, Maryland.

Sayori Shimohata, Toshiyuki Sugio, and Junji Nagata. 1997. Retrieving collocations by co-occurrences and word order constraints. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 476–481, Madrid Spain.

Takaaki Tanaka and Yoshihiro Matsuo. 1999. Extraction of translation equivalents from non-parallel corpora. In *8th International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 109–119, Chester, England, August 23 - 25.