Experiments with a Probabilistic Translation Assistant: would Statistical Grammars help ?

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

Keywords

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

Abstract

In this paper, we present the latest version of TRANSTYPE, a prototype which implements a novel approach to interactive machine translation; namely Target Text Mediated Interactive Machine Translation. We first give an overview of the core system TRANSTYPE relies on. Then, we summarize the results of an *in-situ* evaluation we have carried out this summer. Finally, we discuss the potential benefits that could be gained by integrating a probabilistic grammar in our approach.

1 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. 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 way we integrated user lexicons within TRANSTYPE's completion mechanism. In the fourth section, we describe the strategy we devised to automatically extract bilingual lexicons from training material. This is followed by the presentation of the 2001 *in situ* evaluation we carried out this summer. Finally, we discuss the benefit we could gain by integrating a probabilistic grammar in our approach.

2 An overview of TRANSTYPE

2.1 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

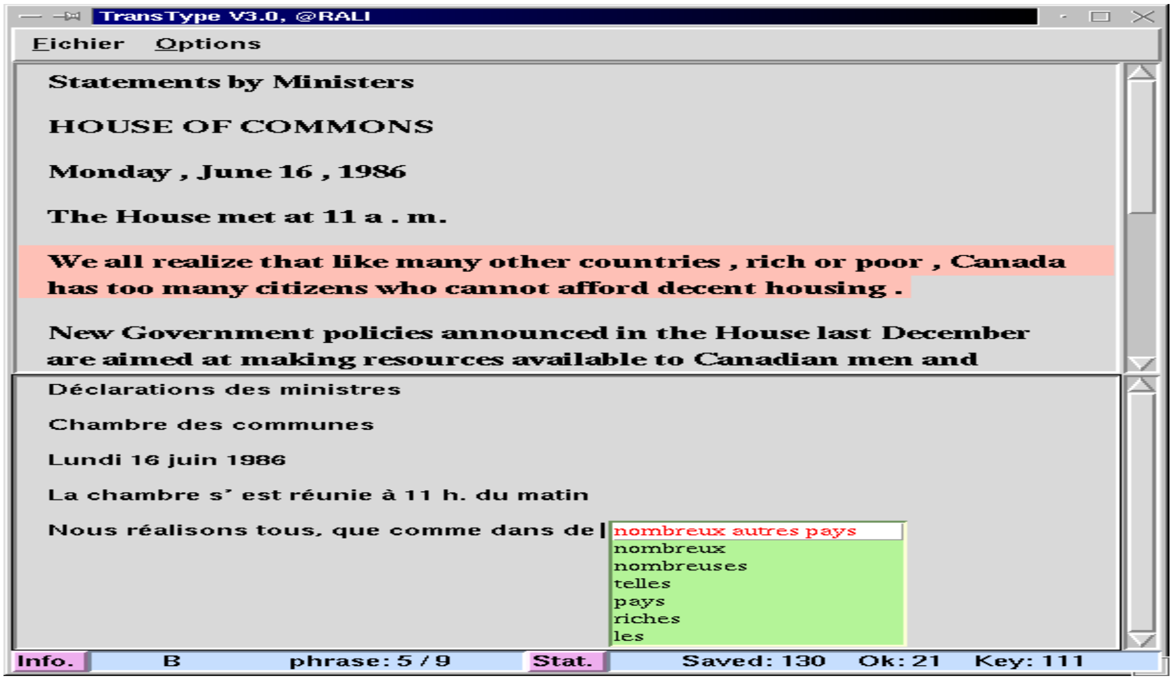

Figure 1: An example of interaction in TRANSTYPE with the source text in the top half of the screen. The target text is typed in the bottom half, with suggestions provided by the menu that appears at the insertion point.

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.

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

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

2.2 Theoretical evaluation

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).

2.3 The 2000 *in-situ* evaluation

An implementation of TRANSTYPE which allowed for the completion of words was evaluated during summer 2000. This *in-situ* evaluation involved ten translators who were asked to translate the same text using TRANSTYPE. The full description of this study may be found in (Langlais et al., 2000b); but some interesting observations emerged which motivated a second evaluation that we recently completed.

First, 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 felt it would be a desirable extension to the system if users were allowed to introduce specific lexicons (hereafter called user lexicon). 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.

3 Plugging user lexicons into TRANSTYPE

This section describes how we integrate non-probabilistic user lexicons within the probabilistic framework of TRANSTYPE. An example of such a lexicon for a text downloaded from the Health Canada web site ("Nutrition for Healthy Infants") is provided in figure 3.

To understand this integration, we need to sketch how TRANSTYPE works basically functions. 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

healthy term infants	nourrissons nés à terme et en santé
dietitians of canada	les diététistes au canada
partly skimmed milk	lait partiellement écrémé
breastfed	nourri au sein

Figure 2: Excerpt of a user lexicon for a text downloaded from the Health Canada web site (“Nutrition for Healthy Infants”).

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.

3.1 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.

3.2 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. (1993), we have:

$$p(t_j|s_1^n) = \sum_{i=0}^{n} t(t_j|s_i)a(i|j,n) \tag{2}$$

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

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 2, 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 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.

3.3 Trace of a translation session

To illustrate the full process, we provide in Table 3.3 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.

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 1: 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.

4 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, 1999) for recent ones). These studies are by nature difficult if not impossible to compare. Therefore, we investigated a simple version which is described in (Langlais et al., 2001) 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, 1998; Och et al., 1999) 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.

4.1 Identifying monolingual salient sequences

4.1.1 Linguistically motivated filters

The literature abounds in statistically minded measures that help to decide whether words that happen to co-occur are linguistically significant or not (see for instance (Dunning, 1993; Shimohata et al., 1997)). We have tried several of them with different amount of succes (Langlais et al., 2000a). In this study, we tried instead several linguistically motivated filters that make use of regular expressions defined on part of speech (POS) tags obtained from a tagger. 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.)

4.2 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,

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.

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. We have described in (Langlais et al., 2001) a way to overcome the tokenization issue.

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).

4.3 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.

4.4 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.

5 The 2001 *in situ* evaluation

We ended up with a new prototype that we decided to test with real users. We asked nine translators with various work experience and expertise to use TRANSTYPE in a controlled setting. Following the first *in situ* evaluation carried out in summer 2000, we took for granted that their translations would be acceptable. We wanted to evaluate our system, not the translators !

The data analysis of this evaluation will be reported elsewhere; here, we provide a summary of the main results.

- If we define the **productivity** as the ratio of the number of characters in the final text over the time it took to produce the text, then we observed an average decrease of the user's productivity of about 10% (this is actually an improvment over last year decrease of 35% !). However, about half of the translators were under the impression that they were going faster with TRANSTYPE.

- All the translators were very happy with the possibility of adding their own lexicon. For four of them, this could even justify adopting TRANSTYPE in their daily work.

- Several users pointed out that it would be better not to be disturbed by short completions (say, 3 characters or less). We actually measured that suggesting short completion always ended up in a loss of time (time to read them and to decide whether they are worthwhile or not). Actually, 51% of the displayed suggestions had less than four characters !

- Despite the observed decrease in terms of productivity, it is worth noting that the completions made by TRANSTYPE could have allowed a careful user to save the entering of more than two thirds of the translation produced.

- Looking at the reaction time and the length of the completions proposed, we conclude that when a user accepts a completion, it is usually at the start of a word. Thus, a possible way to improve the use of the suggestions would be to convince the translators that they should look at the suggestions very soon in their typing process. As looking at these suggestions and deciding if they are worthwhile takes time and can in a way distract the user, suggestions must be valuable. This means, at least, that they should be long enough. We furthermore think that syntactically minded suggestions (for instance, the suggestion of a full noun phrase) would help to convince the user, a point that we develop in the next section.

6 How could a probabilistic grammar help ?

As we just saw, suggesting longer completions seem to be a key point to a successful use of TRANSTYPE. This is actually a tricky situation because the longer the units are, the more time a user will need to process them. The problem certainly involves ergonomic considerations, but intuitively it seems, that suggesting completions that span full syntactic groups (*e.g. the agreement between the parties*, or *the agreement between the parties is that*, etc.) is likely to be a good strategy.

Integrating structural information into probabilistic applications is a challenging research topic. Recently, Chelba et al. (1997) have shown that a structured language model could significantly improve the performance of a speech recognition system. Thanks to the availability of the well known Penn TreeBank (Marcus et al., 1993), it is now possible to produce statistical parsers with good level of performance (Collins, 1996; Charniak, 2001). Charniak (2001) recently proposed a lexicalized statistical parser (SLP) which reduces by 24% the perplexity of a baseline language model (a classical trigram language model), when both the SLP and the baseline model are combined. These studies all suggest that the technology is now mature enough to be successfully integrated within a particular application.

If we consider the way non-probabilistic MT engines have been designed (see for instance Hutchins and Somers (1992) for a description of the major translation systems), they almost invariably follow what we call the ATG approach, an acronym for Analysis, Transfer and Generation. That is, a parser first analyzes the source sentence to be translated. A set of transfer rules are then applied to choose the syntactical structure of the translation. Last, a generation module produces the final translation according to the previously instanciated constraints.

We feel that it may be possible to efficiently implement this approach within a statistical framework, following for instance the cascade model presented in equation 3; where s is the source sentence to translate, $\hat{c}$ is the translation proposed; a_s and a_c are the syntactic trees of respectively the source and the target sentences. In this equation, P_a plays the role of the syntactical source parser, P_t is the transfer model and P_g is the generative model. Such an architecture is illustrated on a simple sentence in figure 3.

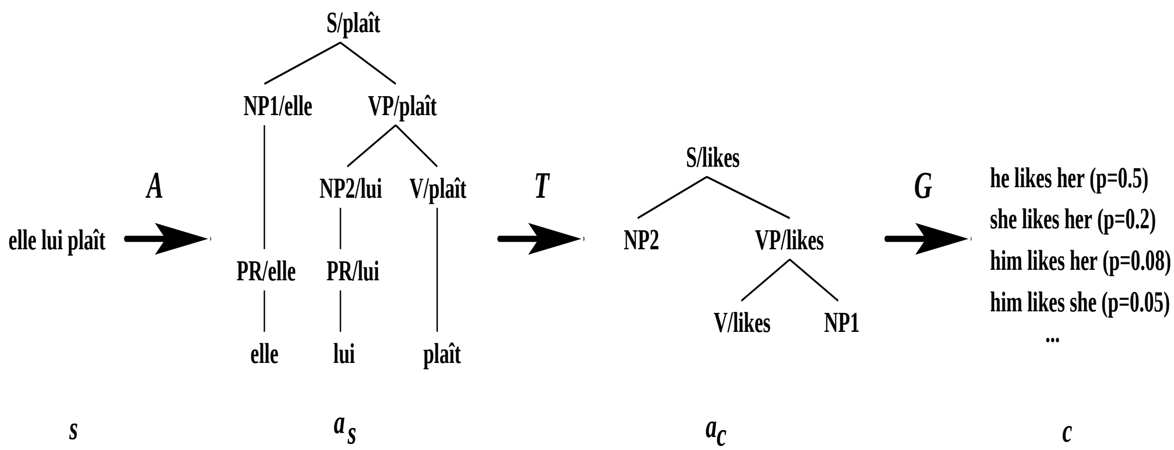

Figure 3: Sketch of the probabilistic ATG approach for the translation of the source sentence *Elle lui plat* into English (He likes her). It is interesting to note, that the translation provided by Babelfish (`http://world.altavista.com`) for this small example is totally sensless: It to him likes.

$$\hat{c} = \operatorname*{argmax}_{c} P_a(a_s|s).P_t(a_c|a_s,s).P_g(c|a_c,a_s,s) \tag{3}$$

There are some tractability questions raised by such a cascade of models. First, the number of syntactical parses for a given sentence grows exponentially with the length of that sentence. Of course, well known techniques such as dynamic beam search or stack decoding may be applied here. Second, it is not clear that designing a transfer model separately is an easy task. Elements of a solution may be found in the work of (Wu and Wong, 1998) where the authors proposes to adopt the noisy channel approach in order to integrate a statistical parser. This approach strongly relies on information expressed in the form of lexical rules such as `Noun-> cat/chat`. However, we hope to design a transfer model that does not rely on such a fine level of granularity, but instead focusses more on matching syntactic constructions (this construction may involve general information such as the head word which dominates each node in the structure). This would offer a great deal of generalization, and would also simplify the adaptation of a translation engine to a new pair of languages.

Our faith in the appropriatness of the approach is confirmed by a recent study (Yamada and Knight, 2001) in which the authors propose a model in which the noisy-channel takes as input a parsed sentence rather than simple words. This model incorporates three kinds of operations on this syntaxic tree, each being orchestered by an automatically learned probabilistic model. The first operation consists of *reordering* the direct daughters of a given node. In a translation task from English to Japanese, the authors report that the consulted model has learned for instance that an English sentence composed of a preposition, a first verbal group and a second verbal group ($PREPVB1VB2$) is likely to present an inversion of the two verbal groups ($PREPVB2VB1$). The second operation consists of *inserting* extra words at each node of the tree; this is done by consulting a model conditioned on the identity of the two direct ancestors of the node under inspection. Finally, the third operation consists of *translating*, as a normal word-model would do, the words of the transformed tree. These operations allow us to model the translation of languages with different word orders. The authors report encouraging performance of the resulting translation engine, but further evaluations are still needed to be conclusive.

7 Discussion

In this paper, we have presented the current version of TRANSTYPE which prompts the translator with suggestions of both single words and sequences of words. The prototype also allows a translator to take advantage of her/his personal bilingual lexicons.

An algorithm to automatically acquire bilingual lexicons has been presented and positively evaluated by an expert. The main results of a recent *in situ* evaluation of this prototype has been reported, showing that the system slightly reduces the productivity of translators, and this despite the fact that the system was suggesting more that two thirds of the total material to be entered by the translator. The time required to adequately master TRANSTYPE may be part of the explanations for this disappointing result.

We finally discussed how a probabilistic parser could be integrated into TRANSTYPE, by reformulating in a statistical framework an approach that has inspired the design of many traditional translation engines.

Acknowledgments

TRANSTYPE is a project which has been funded by the Natural Sciences and Engineering Research Council of Canada. I am indebted to Elliott Macklovitch for the fruitful orientation he gaves to this work. I also want to warmly thank Sébastien Sauvé and Marie Loranger who organized the two evaluation sessions of TRANSTYPE. Last, I would like to thank the subjects who have participated in these evaluations and from whom we have gained much insight.

References

[Brown et al.1993] 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.

[Charniak2001] Eugene Charniak. 2001. Immediate-head parsing for language models. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 116–124, Toulouse. Morgan Kaufmann Publishers.

[Chelba et al.1997] Ciprian Chelba, David Engle, Frederick Jelinek, Victor M. Jimenez, Sanjeev Khudanpur, Lidia Mangu, Harry Printz, Eric Ristad, Ronald Rosenfeld, Andreas Stolcke, and Dekai Wu. 1997. Structure and performance of a dependency language model. In *Proc. Eurospeech '97*, pages 2775–2778, Rhodes, Greece.

[Collins1996] Michael John Collins. 1996. A new statistical parser based on bigram lexical dependencies. In Arivind Joshi and Martha Palmer, editors, *Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics*, pages 184–191, San Francisco. Morgan Kaufmann Publishers.

[Dunning1993] Ted Dunning. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74.

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

[Hutchins and Somers1992] W. John Hutchins and Harold L. Somers. 1992. *An Introduction to Machine Translation*. Academic Press, New York.

[Jacquemin1999] Christian Jacquemin. 1999. 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.

[Langlais et al.2000a] 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.

[Langlais et al.2000b] 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.

[Langlais et al.2001] Ph. Langlais, G. Foster, and G. Lapalme. 2001. Integrating bilingual lexicons in a probabilistic translation assistant. In *MT Summit VIII*, Santiago de Compostela, Spain, September.

[Marcus et al.1993] M. Marcus, S. Santorini, and M. Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.

[Melamed1997] 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.

[Och and Weber1998] Franz Josef Och and Hans Weber. 1998. 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.

[Och et al.1999] Franz Josef Och, Christoph Tillman, and Herman Ney. 1999. 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.

[Ohomori and Higashida1999] 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.

[Rapp1999] 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.

[Shimohata et al.1997] 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.

[Tanaka and Matsuo1999] 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.

[Wu and Wong1998] Dekai Wu and Hongsing Wong. 1998. Machine translation with a stochastic grammatical channel. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*, pages 1408–1414, Montréal, Canada.

[Yamada and Knight2001] Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 523–530, Toulouse, France, jul.