Using Context-Dependent Interpolation to Combine Statistical Language and Translation Models for Interactive Machine Translation

Philippe Langlais and George Foster
Université de Montréal
C.P. 6128, succursale Centre-ville
H3C 3J7, Montréal, Québec, Canada
{felipe,foster}@IRO.UMontreal.ca

Abstract

This work is in the context of TRANSTYPE, a system that watches over the user as he or she types a translation and repeatedly suggests *completions* for the text already entered. The user may either accept, modify, or ignore these suggestions. The system's proposals are selected and scored using a linear combination of a trigram language model and a translation model. We investigate the issue of how weights should be assigned to these two models in different contexts.

1 Introduction

TRANSTYPE is part of a project set up to explore an appealing solution to the problem of using *Interactive Machine Translation* (IMT) as a tool for professional or other highly-skilled translators. IMT first appeared as part of Kay's MIND system (Kay, 1973), where the user's role was to help the computer analyse the source text by answering questions about word sense, ellipsis, phrasal attachments, etc. Most later work on IMT, eg (Blanchon, 1991; Brown and Nirenburg, 1990; Maruyama and Watanabe, 1990; Whitelock et al., 1986), has followed in this vein, concentrating on improving the question/answer process by having less questions, more friendly ones, etc. Despite progress in these endeavors, systems of this sort are generally unsuitable as tools for skilled translators because the user serves only as an advisor, and the MT component has overall control of the translation process.

TRANSTYPE originated from the conviction that a better approach to IMT for competent translators would be to shift the focus of interaction from the *meaning* of the source text to the *form* of the target text (Foster et al., 1997). This would relieve the translator of the burden of having to provide explicit analyses of the source text and allow him to translate naturally, assisted by the machine whenever possible.

In this idea, a translation emerges from a series of alternating contributions by human and 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 directly in control of the process: the machine must respect the constraints implicit in his contributions, and he is free to accept, modify, or completely ignore its proposals.

The above description encompasses a number of interesting scenarios for interaction. In our current prototype, we have implemented one of the simplest, where the machine's task is just to try

to guess what the translator will type next. The translator is given access to the system's proposed completions—here limited to a single word or suffix of a word—and can incorporate them into the target text whenever desired. Completions are generated using a linear combination of separate predictions from the target text (a trigram language model) and the source text (a translation model). In this paper, we present a theoretical model of the task addressed by TRANSTYPE and we report on experiments carried out to explore different ways of combining these two predictive sources.

2 TRANSTYPE and its model

2.1 User Viewpoint

Our interactive translation system is illustrated in figure 1 for English to French translation. It works as follows: a translator selects a sentence and begins typing its translation. After each character typed by the translator, the system displays a proposed completion, which may either be accepted using a special key or rejected by continuing to type. This interface is simple and its performance may be measured by the proportion of characters or keystrokes saved in typing a translation.

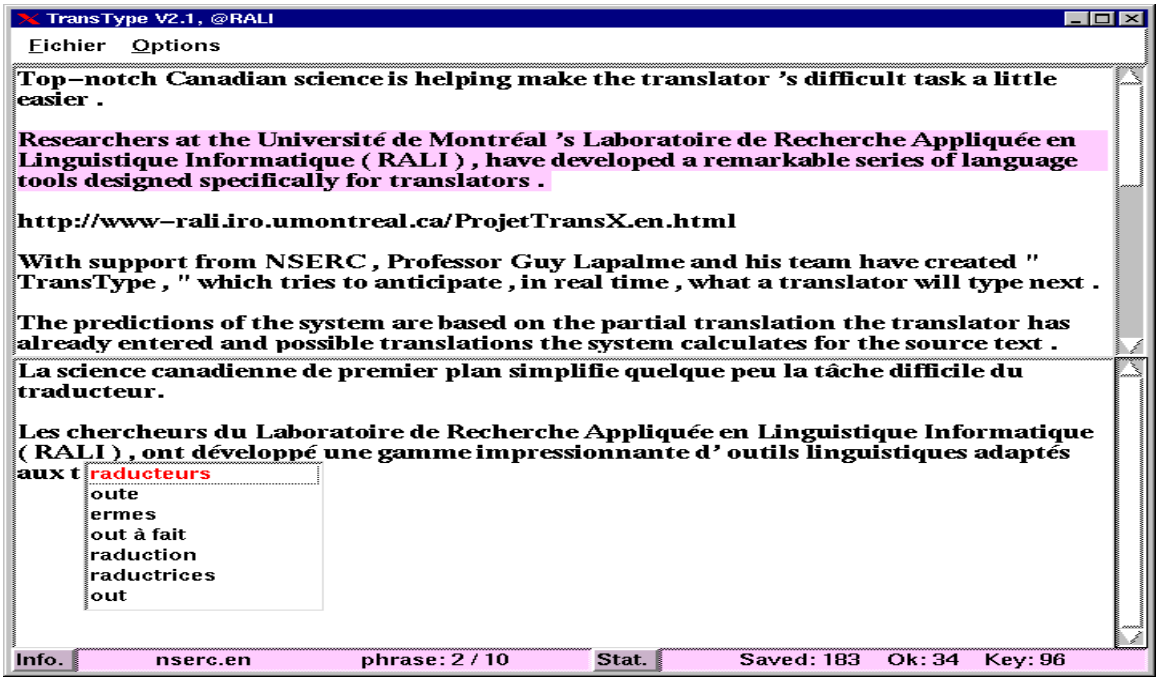

Figure 1: Example of an interaction in TRANSTYPE with the source text in the top half of the screen. The target text is typed in the bottom half with suggestions given by the menu at the insertion point.

Although this form of translation completion is expected to be useful for translators, we have not yet verified this conjecture. The goal of this paper is to show that this kind of interaction is within the reach of current MT technology, and to measure the level of accuracy which is attainable by a simple system. The user-interface design choices and a more formal evaluation within the global task of translation will be the subject of another paper.

2.2 System Viewpoint

To complete words, TRANSTYPE relies on two main components: the *generator* which produces a list of hypotheses that match the current (possibly null) word prefix and the *evaluator* which ranks them.

The generator computes for each source segment (usually a sentence), an *active vocabulary* consisting of the set of words to which the translation model (see below) assigns the highest probabilities, along with a static list of frequent words compiled from a training corpus. 90% of the target tokens of a 30000 word test corpus were covered by this process, with an active vocabulary size of less than 500 words.

The evaluator implements a model which computes an estimate of $p(t|\tilde{\mathbf{t}}, \mathbf{s})$, the probability of a target word t given the preceding target context $\tilde{\mathbf{t}}$, and a source segment $\mathbf{s}$. Creating this model means finding some decomposition of $p(t|\tilde{\mathbf{t}}, \mathbf{s})$ in terms of parameters whose values can be estimated from a training corpus.

They are many ways of accomplishing this, of which the most obvious is the classical noisy channel method. One drawback of a noisy channel approach is that it requires a complex decoding strategy. Although recent methods for efficient dynamic-programming (Tillman et al., 97; Niessen et al., 98) and stack-based (Wang and Waibel, 97; Wang and Waibel, 98) decoders have been proposed, we consider these strategies still too expensive for TRANSTYPE (recall that a completion must be generated after each keystroke).

Thus, for reasons of search efficiency we chose to use separate models to capture predictions from the target and source texts, then combine them into a single global prediction. Our basic method is a linear combination of source and target text models, using some weighting factors λ (see equation 1). Linear combination is a weak technique because it tends to average out the strengths and weaknesses of its components. It always performs at least as well as best of the two, but in practice it usually does not perform much better. For this reason, we investigated weights which depend on the context:

$$p(t|\tilde{\mathbf{t}}, \mathbf{s}) = \lambda(\Theta(\tilde{\mathbf{t}}, \mathbf{s}))\, p(t|\tilde{\mathbf{t}}) + (1 - \lambda(\Theta(\tilde{\mathbf{t}}, \mathbf{s})))\, p(t|\mathbf{s}) \quad (1)$$

where $\Theta(\tilde{\mathbf{t}}, \mathbf{s})$ stands for any function which maps $\tilde{\mathbf{t}}$,$\mathbf{s}$ into a set of equivalence classes. Intuitively, $\lambda(\Theta(\tilde{\mathbf{t}}, \mathbf{s}))$ should be high when $\mathbf{s}$ is more informative than $\tilde{\mathbf{t}}$ and low otherwise.

An advantage of this approach is that there are well-established modeling techniques for both distributions in this equation. Currently, the first distribution is approximated by an interpolated trigram model for French, of the type commonly used in speech recognition (Jelinek, 1990), and the second distribution derives from an IBM-style statistical translation model (1&2) (Brown et al., 1993). Both have been trained on a large portion of the Canadian Hansard corpus (a large collection of texts of Canadian parliamentary debates). Details of the training procedure are given in (Foster et al., 1997).

3 The experimental protocol

The modeling strategy we devised, although easy to implement, raises the problem of defining which information in the target prefix $\tilde{\mathbf{t}}$ and in the source text $\mathbf{s}$ could serve as a good predictor of the

relative strength of the two models. This is the problem we address in the following. More specifically, we investigated the usefulness of different functions $\Theta(\tilde{\mathbf{t}}, \mathbf{s})$, some extracting information from the target words already validated by the translator, others depending on simple intrinsic properties of the source segment under translation.

As suggested by the linear form of equation 1, we used the EM algorithm to estimate optimal weighting coefficients for each candidate mapping $\Theta(\tilde{\mathbf{t}}, \mathbf{s})$, maximizing in an iterative process the probability assigned by p to a training corpus.

3.1 The EM-corpus

To estimate the weighting coefficients, we randomly extracted a hundred pairs of files of the Hansard corpus. These pairs of texts were automatically aligned to the sentence level, using the program described in (Langlais and El-Bèze, 97)[1]. Then we filtered out all non one-to-one pairs and removed those containing sentences longer than 40 words. We call this bitext the *EM-corpus*, to distinguish it from the much larger corpus used to train the translation and language models.

3.2 The test corpus

The test corpus consists of three automatically aligned texts not used for training; two coming from different periods of the Hansard corpus, and one from an unrelated corpus (a text on the competitiveness of the Canadian milk and dairy products industry), as shown in table 1.

Corpus	Segments	English tokens	French tokens	1-1 pairs
EM-corpus	154559	2820549	3017942	100 %
A (Hansard-1986)	786	19457	21130	93 %
B (Hansard-1992)	1140	29886	32138	87.7 %
C (non-Hansard)	594	18881	21303	93.9 %

Table 1: Characteristics of the corpora used in this study in terms of number of segments, number of source and target tokens. The last column indicates the percentage of pairs in which one source sentence has been aligned to one target sentence by our automatic aligner.

3.3 Evaluation

We evaluated the performance of different methods for contextual model weighting by measuring the theoretical performance of TRANSTYPE over a pair of translated texts. More precisely we measured the number of keystrokes saved by a hypothetical user producing the target text as a translation of the source text, typing each sentence from left to right. A completion is proposed automatically by the system after each keystroke. The user then has two choices: 1) acceptaing the completion by typing an acceptance key, or 2) ignoring the completion by typing the next character of the word under translation. We assumed that a translator carefully observes each completion proposed by the system and accepts it as soon as it is correct.

[1]For a recent comparison of different alignment techniques, see the ARCADE exercise (Langlais et al., 98)

We do not suggest that this fully automatic evaluation is an accurate reflection of the labour saved by TRANSTYPE. We are currently investigating a user evaluation of our prototype, and will present a comparison of these two evaluation procedures in another paper.

For sake of comparison we report the performance rate obtained by the model described in (Foster et al., 1997). It belongs to the familly of models described by equation 1; where any coefficient λ is set to a single value regardless of the context. This value has been chosen as to optimize the completion performance over a test corpus, that is $\lambda(\Theta(\tilde{\mathbf{t}}, \mathbf{s})) = 0.6$. This is the *baseline* model we considered in this work.

4 Experiments

To gauge how well we can perform by appropriately mixing the language model and the translation model predictions within our linear framework, we ran a fake completion session where the identity of each word under completion was known. The completion proposed after each keystroke for the expected current word t_e was set to t_e if either of the models ranked it first, otherwise to the best token $\hat{t}$ according to the baseline model:

$$\hat{t} = \begin{cases} t_e, \text{ if } \operatorname{argmax}_t p(t|\mathbf{s}) = t_e \text{ or } \operatorname{argmax}_t p(t|\tilde{\mathbf{t}}) = t_e \\ \operatorname{argmax}_t 0.6p(t|\mathbf{s}) + 0.4p(t|\tilde{\mathbf{t}}), \text{ else} \end{cases} \quad (2)$$

This experiment indicated that the global performance of TRANSTYPE can be improved by a maximum of approximately 3.7% over the baseline. This represents a reduction of 12.3% in the number of keystrokes, with better predictions (shorter prefixes required) for 19% of words. Having better predictions for a fifth of all words is an improvement that seems very likely to be noticeable to a user of TRANSTYPE, although we have not yet run tests to establish this.

4.1 Frequency-based functions

The target words already validated by the translator may serve as a predictor of the weights to assign to both models. The first set of functions we looked at extract the frequency—as counted in the corpus used to train the language model—of the two last completed target tokens (t', t''). This idea is widely used to smooth the distribution of unseen or rare trigrams using the conditioning bigram and unigram frequencies (Chen and Goodman, 1996).

Θ_{f_2} and Θ_{cf_2} extract the frequency of the bigram conditioning the token under completion t, while Θ_{f_1} and Θ_{cf_1} focus on the unigram frequency. Θ_{cf_2} and Θ_{cf_1} cluster the frequency on a logarithmic scale. More precisely, if $f(x)$ stands for the frequency (as measured in the training corpus) of the event x, and Int for the function returning the integer part of its argument, then:

$$\Theta_{cf_1}(\tilde{\mathbf{t}}, \mathbf{s}) = Int(10 \times log_{10} f(t'')) \qquad \Theta_{f_1}(\tilde{\mathbf{t}}, \mathbf{s}) = f(t'')$$
$$\Theta_{cf_2}(\tilde{\mathbf{t}}, \mathbf{s}) = Int(10 \times log_{10} f(t't'')) \quad \Theta_{f_2}(\tilde{\mathbf{t}}, \mathbf{s}) = f(t't'')$$

From the results presented in table 6, we can make several observations: a) the best of these four functions do not improve much the results obtained with the baseline; b) clustered frequency functions always outperform their non clustered counterparts; and c) unigram frequency information seems a slightly better predictor than bigram frequency.

The first observation is unfortunatly a constant in our experiments: improving the baseline evaluator is not an easy task. Actually, the present situation differs slightly from a smoothing problem as the two components we want to combine behave differently. The language model captures local grammatical constraints, but displays a high degree of lexical uncertainty, while the translation model has a good idea of which words should appear in the target text but a poor notion of where to put them.

As pointed out in (Foster et al., 1997), we are faced with a local consistency problem. Lowering the weight on the language model in a specific context may introduce ungrammatical sequences. The eighth line 8 in table 3 (see section 4.2) shows such a situation. The weight given to the language model to complete a word after the high-entropy French bigram *et les* (*and the*) is almost null, which reflects the fact that many words may follow the bigram. Thus, the prediction relies mostly on the translation model whose first hypothesis is the word *et*, the correct translation of the word *and* in *Minister and Hon.*. This leads to a totally non grammatical target sequence *et les et*.

Conversely, raising the language model weight will favour the tokens that more frequently follow, in the training corpus, the conditioning context, even if there are overwelming evidences from the source part against them. An illustration of this can be seen in the twelveth line of table 3 where the weight given to the translation model in order to predict the word following the source sequence *s' intéressent à* (*are interested in*) is almost null. This favours the language model's prediction (*l'*, which is incorrect) at the expense of the expected one (*ces*) that was correctly proposed by the translation model.

In the following, we refer to these local consistency problems as **sequence breaks**. The two last points we observed from the results may be seen as a side effect of the remark we just made: the more contexts we consider, the more probable sequence breaks become. In this respect, we can see this problem as an over-training one that is largely avoided with the baseline model, as only one weight is assigned globally to balance the two prediction sources.

Looking more closely at the coefficients estimated by the EM algorithm, we observe several interesting things. In the EM-corpus, there are more than 443,000 different bigrams whose frequencies determine a set of more than 3500 values (*vs* 51 in the clustered version). A small portion of very frequent bigrams are fully identified by the number of times they occurred in the training corpus. For instance the bigram $nil1\,nil2$, the one indicating the beginning of a sentence, is the only one in the training corpus to occur 1,619,682 times (in fact, the number of sentences used to train the language model).

A few of these are reported in table 2 with the estimated weight that should be given to the translation model for the next completion. We observe that high-entropy bigrams tend to favour the translation model. For instance to complete a word at the very beginning of a sentence, weighting the translation model highly seems the best strategy. The language model, in this situation, always predicts the most frequent word of the training corpus, that is, *le* (the). Conversely, in a low entropy context such as *Monsieur le*, which is followed 99% of the time by the word *Président* in our EM-corpus, TRANSTYPE will rely mostly on the language model, even if there is overwhelming evidence against it from the source segment.

Frequency	EM-frequency	λ	associated bigram	following forms
1619682	154559	0.91	nil1 nil2	2093
176609	11479	0.88	de l'	951
121127	8839	0.55	le gouvernement	784
127511	12602	0.89	nil2 le	872
120753	11528	0.43	nil2 je	315
109648	7267	5.78e-05	monsieur le	12
105923	7492	0.99	nil2 m.	695
85646	7531	0.92	de loi	767
83799	7331	0.01	projet de	72

Table 2: Relation between some high bigram frequencies, and the weight assigned to the translation model after few EM iterations. The last column indicates the number of the different forms that follow the bigram in the EM-corpus.

It is less obvious how to interpret directly the contexts defined by frequencies that characterize many bigrams. We can however observe from our data that the less frequent a conditioning bigram is (especially for those occurring less than 500 times), the less weight is given to the translation model ($\lambda < 0.3$).

4.2 Target token based functions

Instead of reducing the knowledge carried by a conditioning context to just its frequency, we can make the weighting function consider the tokens directly. We tried two functions that consider, respectively, the conditioning bigram and unigram: $\Theta_{bi}(\tilde{\mathbf{t}}, \mathbf{s}) = t't''$, and $\Theta_{uni}(\tilde{\mathbf{t}}, \mathbf{s}) = t''$.

As can be seen from table 6 these contexts give slightly worse results than their frequency counterpart and never outperform the baseline. This confirms the remark we made in the previous section: the more contexts we consider, the more sequence breaks are likely to occur.

We also report in table 6 the performance of a function (pos), that was set up to handle more simply the special situation of completions at the beginning of a sentence, by considering the number of tokens already completed: $\Theta_{pos}(\tilde{\mathbf{t}}, \mathbf{s}) = |\tilde{\mathbf{t}}|$.

4.3 Contexts based on POS

A way of reducing the number of different contexts, and thus lowering the number of potential sequence breaks, without losing all the linguistic information of the target text already validated is to consider part-of-speech (POS) information. Although there are obviously better ways to integrate the POS information in TRANSTYPE[2], we tried to gauge its potential within the linear combination currently used in TRANSTYPE.

We first tried to make the interpolation coefficient of equation 1 dependent on the POS tag of the word being considered for completion. For obvious reasons, we can't assume that equation 1 still sums up to 1 over all French words[3]. We used a statistical tagger described in (Foster, 1991), but reduced its set of tags to 15 macro-classes by removing information such as gender or number.

[2]for instance a word-class mechanism such as: $p(t|\tilde{\mathbf{t}}, \mathbf{s}) = \sum_c\ p(t|c, \tilde{\mathbf{t}}, \mathbf{s})\ p(c|\tilde{\mathbf{t}}, \mathbf{s})$ is a more appealing one

[3]Indeed, we verified that the sum of $p(t|\tilde{\mathbf{t}}, \mathbf{s})$ over the French vocabulary of the EM-corpus is slightly less than unity

	λ	oracle	prefix	prediction	TM first	LM first
1	0.91	Je	+	/Je	Je	Le
2	0.43	sais	+	/sais	que	ne
3	0.034	que	+	/que	que	que
4	0.42	la	la+	/le l/es	ministre	le
5	0.81	ministre	+	/ministre	ministre	chambre
6	0.41	et	+	/et	ministre	a
7	0.47	les	le+	/ministre l/e le/s	et	le
8	0.96	députés	d+	/et d/éputés	et	autres
9	0.39	s’	s’+	/de s/ont	députés	de
10	0.02	intéressent	in+	/en i/l in/téressent	députés	en
11	0.01	à	+	/à	ces	à
12	0.03	ces	ce+	/l’ c/ette ce/s	ces	l’
13	0.74	questions	q+	/ces q/uestions	ces	gens
14	0.63	pratiques	pr+	/ces p/ratique pr/atiques	ces	et
15	0.40	.	.+	/qui	pratiques	qui

Table 3: A sample completion session for the English source sentence *I know that the Minister and Hon. Members are interested in these practical issues.*. The first column contains the translation model weight, the second one the French target sentence; the third, the prefix typed by the translator; the fourth, the record of the successive proposals for each token, with a slash separating the prefix from the proposed completion. The two last columns indicate the best hypotheses made respectively by the translation model (TM) and the language model (LM) alone.

Table 4 reports the coefficients assigned to the translation model by the EM algorithm. As expected, the translation model is weaker on frequent function words such as articles (Dete), prepositions (Prep), pronouns (Pron) or even punctuation (Punc). It is stronger on quantities (Quan, Ordi) and proper names (NomP). As can be seen on table 5, a small excerpt of the translation matrix associated to model 1, high-entropy words e, such as *the*, are characterized by a flat probability distribution $p(f|e)$, where f stands for any French word.

We did not run any performance tests with this way of extracting POS information, as in our prototype computing the tag of each word candidate is a time consuming process[4] that would make the interface impractical. Instead, we considered the non-optimal function Θ_{cl} which computes the

[4]Especially when the word expected by the translator is not in the active vocabulary.

POS	λ	nb	POS	λ	nb	POS	λ	nb
Dete	0.11	193315	Pron	0.29	107514	Ltre	0.39	588
Prep	0.13	117198	AdjQ	0.31	61423	NomC	0.45	245199
Verb	0.25	184980	ConC	0.38	22200	Quan	0.59	32732
ConS	0.27	32131	Inte	0.39	1483	Ordi	0.62	2500
Punc	0.28	128762	Adve	0.39	63891	NomP	0.64	28447

Table 4: Relation between the POS tag of the token under completion and the translation weighting coefficient. nb is the number of observations of each POS tag in the EM-corpus.

most likely POS tag, given the preceding two last tokens $t't''$ validated by the translator, that is:

$$\Theta_{cl}(\tilde{\mathbf{t}}, \mathbf{s}) = \operatorname{argmax} c \;\; p(c|c'c'')\, p(c'c''|t't'')$$

As can be observed in table 6, this is the function that gives the best results on text B.

source	5-best target associations $(word, probability)$									
canada	canada	0.62	du	0.11	au	0.11	le	0.05	pays	0.02
give	donner	0.26	aux	0.07	de	0.05	à	0.05	accorder	0.04
safety	sécurité	0.64	la	0.23	de	0.04	matière	0.02	des	0.01
say	dire	0.36	que	0.21	dit	0.05	dis	0.04	disent	0.03
take	prendre	0.17	de	0.08	pour	0.08	.	0.04	le	0.03
the	le	0.17	la	0.15	de	0.11	l'	0.07	les	0.06

Table 5: Excerpt of the translation matrix. The five most likely French words are reported eith their associated association probability.

4.4 Source based functions

A way to reduce the sequence breaks rate is to consider information contained in the source segment under translation. It is however not obvious how to define a function that would be useful for our combination task without including information that could be handled in a more satisfactory way directly by the translation model.

We made several simple trials that we briefly comment upon. A first function we looked at is the length (counted in tokens) of the source text under translation (namely lg in table 6). The intuition behind this choice is that the translation model may be less accurate for predicting target words when faced with long sentences. As a matter of fact, when assigning the probability $p(t_i|\mathbf{s})$ to a target token t_i, knowing the source segment $\mathbf{s}$, the weighted contribution of each source word is summed up to give the final estimation :

$$p(t_i|\mathbf{s}) = \sum_{j=0}^{S} p(t_i|s_j)\, a(j|i, S, T) \tag{3}$$

where in model 1, the weights $a(j|i, S, T)$ are uniformly distributed over the number of source tokens, and therefore are equal to $1/(S+1)$[5], while in model 2, $a(j|i, S, T)$ is higher when i and j are closer.

As can be seen in table 6, lg gives better results on text C which is the only non-Hansard corpus, but provides worse performances on A and B. Our first thought is however confirmed. Looking at figure 2 we do see a relation between the weight assigned by the EM-algorithm to the translation model and the number of tokens of the source segment. Note that very short sentences (less than 4 words) are mostly titles which occur frequently and should be better considered as single units in TRANSTYPE.

[5] $S+1$ because of the token s_0 added to handle insertion events.

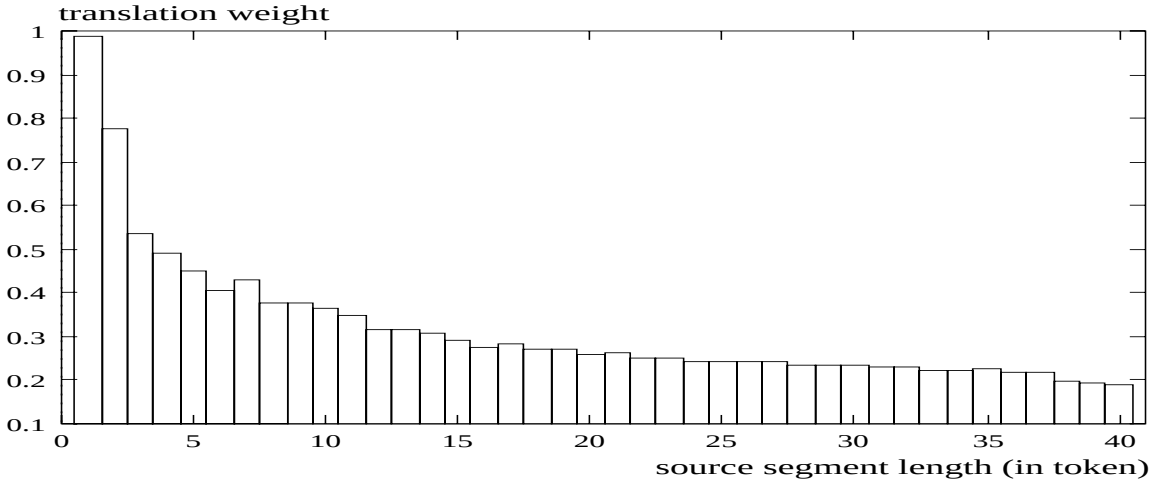

Figure 2: Translation weight as a function of the source segment length (counted in tokens).

The last approach we considered, with the hope of improving the completion accuracy at the beginning of sentences, consisted in two functions, namely $first_1$ and $first_2$, which respectively associate with any context $(\tilde{\mathbf{t}}, \mathbf{s})$, the first source token and the first two source tokens. Both functions slightly improve the baseline performance.

5 Results

Table 6 summarizes the performance of the functions we have described. Results for the non-Hansard text C are substantially worse than for either of the two Hansard texts. This may be seen as a lower bound on TRANSTYPE's performance when faced with texts unrelated to the ones used for the training stage, especially if we consider that none of the models used in the current version of the prototype have any capacity to adapt dynamically.

The results are discussed individually in the previous section. Figure 3 provides more clues about how TRANSTYPE performs. The left plot reports the average number of choices that the evaluator has to rank, as a function of the length of the prefix typed by the translator. Without any prefix typed, the system has to select a word among 492 ones (actually the average size of the active vocabulary associated to a given segment), when the translator types the first character, the number of choices falls to around 35. The right figure plots the percentage of words that have been completed (using the baseline model) versus the prefix length. 21% of words are successfully completed by TRANSTYPE without any prefix, 55% with a prefix less than two characters.

6 Conclusion

We have described experiments for assigning context-dependent weights to the translation model and the language model within an interpolated combination used for TRANSTYPE. It turns out there are only a few sources of information which are reliable indicators of relative performance and which can also be extracted efficiently. Chief among them is the fact that, at the beginning of a sentence, the translation model achieves better completions. None of the combinations we tried yielded significant improvements despite an estimated maximum potential gain of 3.7%. Interesting

	A		B		C	
Context	nb	%	nb	%	nb	%
tm	54961	58.121	84298	57.0186	53030	49.8196
lm	64823	68.5501	100386	67.9004	64751	60.831
bi	66256	70.0655	102437	69.2877	66033	62.0354
f_2	66332	70.1458	102837	69.5582	66217	62.2083
uni	66454	70.2748	102955	69.6381	66312	62.2975
f_1	66476	70.2981	103024	69.6847	66404	62.384
lg	66454	70.2748	103035	69.6922	*66540*	*62.5117*
base	**66547**	**70.3732**	**103105**	**69.7395**	**66444**	**62.4216**
cf_2	66471	70.2928	*103140*	*69.7632*	*66501*	*62.4751*
$first_2$	66499	70.3224	*103109*	*69.7422*	*66545*	*62.5164*
cf_1	66501	70.3245	*103160*	*69.7767*	*66579*	*62.5484*
$first_1$	66511	70.3351	*103149*	*69.7693*	*66642*	*62.6076*
pos	66516	70.3404	*103161*	*69.7774*	*66661*	*62.6254*
cl	66522	70.3468	*103234*	*69.8268*	*66611*	*62.5784*

Table 6: Completion results for different contexts. Results are presented in increasing order of keystrokes saved (absolute counts and percentages) in the three corpora (A, B, and C). The baseline performance is indicated in bold, the contexts which outperform the baseline on a given corpus are indicated in italics.

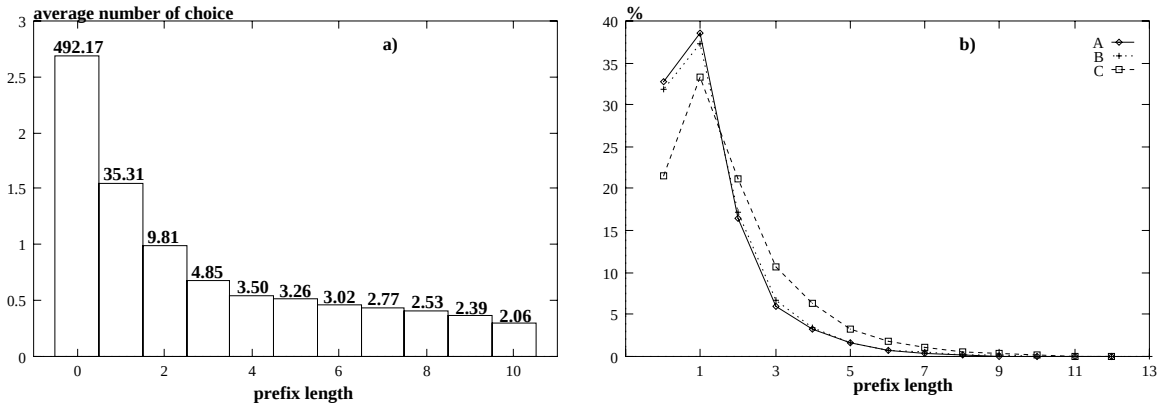

Figure 3: a) Average number of choices (on a 10-based logarithm scale), as function of prefix length (counted in characters). b) Percentage of completions achieved by the baseline system versus the length (counted in characters) of the prefix under completion.

features have however been observed and discussed.

There are numerous ways of extending the current capabilities of TRANSTYPE. We are currently working on two main avenues. The first one consists in modeling the translation and the language sources with a maximum entropy approach. This should provide a principled way to mix the translation and the language sources. We are also attacking the completion of unit larger than a singe word, a challenging issue that should reveal more of the potential of our approach. We conducted a first encouraging experiment that will be integrated into the demonstration prototype we intend to present at the conference.

Acknowledgements

TransType is a project funded by the Natural Sciences and Engineering Research Council of Canada NSERC. It also benefits from an industrial partnership with Machina Sapiens.

References

Hervé Blanchon. 1991. Problèmes de désambiguïsation interactive et TAO personnelle. In *L'environnement Traductionnel*, Journées scientifiques du Réseau thématique de recherche "Lexicologie, terminologie, traduction", pages 31–48, Mons, April.

Ralf D. Brown and Sergei Nirenburg. 1990. Human-computer interaction for semantic disambiguation. In *International Conference on Computational Linguistics (COLING)*, pages 42–47, Helsinki, Finland, August.

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.

Stanley F. Chen and Joshua Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 310–318, Santa Cruz, California.

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

George F. Foster. 1991. Statistical lexical disambiguation. Master's thesis, McGill University, School of Computer Science.

Frederick Jelinek. 1990. Self-organized language modeling for speech recognition. In A. Waibel and K. Lee, editors, *Readings in Speech Recognition*, pages 450–506. Morgan Kaufmann, San Mateo, California.

Martin Kay. 1973. The MIND system. In R. Rustin, editor, *Natural Language Processing*, pages 155–188. Algorithmics Press, New York.

Philippe Langlais and Marc El-Bèze. 97. Alignement de corpus bilingues :algorithmes et évaluation. In *1ères Journées Scientifiques et Techniques du réseau FRANCIL*, pages 191–197, Avignon, France, Avril.

Philippe Langlais, Michel Simard, and Jean Véronis. 98. Methods and practical issues in evaluating alignement techniques. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistic*, Montréal, Canada.

Hiroshi Maruyama and Hideo Watanabe. 1990. An interactive Japanese parser for machine translation. In *International Conference on Computational Linguistics*, pages 257–262, Helsinki, Finland, August.

S. Niessen, S. Vogel, H. Ney, and C. Tillman. 98. A dp based search algorithm for statistical machine translation. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*, pages 960–966, Montréal, Canada.

C. Tillman, S. Vogel, H. Ney, and A. Zubiaga. 97. A dp based search using monotone alignments in statistical translation. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 289–296, Madrid, Spain.

Ye-Yi Wang and Alex Waibel. 97. Decoding algorithm in statistical machine translation. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 366–372, Madrid, Spain.

Ye-Yi Wang and Alex Waibel. 98. Fast decoding for statistical machine translation. In *proceedings of the 5th International Conference on Spoken Language Processing*, Sydney, Australia.

P. J. Whitelock, M. McGee Wood, B. J. Chandler, N. Holden, and H. J. Horsfall. 1986. Strategies for interactive machine translation: the experience and implications of the UMIST Japanese project. In *International Conference on Computational Linguistics*, pages 329–334, Bonn, West Germany.