Evaluation of TRANSTYPE, a Computer-aided Translation Typing System: A comparison of a theoretical- and a user- oriented evaluation procedures

Philippe Langlais, Sébastien Sauvé, George Foster, Elliott Macklovitch, Guy Lapalme

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

Abstract

We describe and compare two protocols — one theoretical and the other in-situs — for evaluating the TRANSTYPE system, a target-text mediated interactive machine translation prototype which predicts in real time the words of the ongoing translation.

1. Introduction

TRANSTYPE is part of a project set up to explore an appealing solution to Interactive Machine Translation (IMT). The first IMT facility was implemented as part of Kay's MIND system (Kay, 1973), where the user's role was to help the computer to analyse the source text by answering questions about word sense, ellipses, phrasal attachments, etc. Later works (Blanchon, 1991; Brown and Nirenburg, 1990; Maruyama and Watanabe, 1990; Whitelock et al., 1986) mainly have concentrated on lightening the question/answer process (less questions, more friendly ones, etc).

TRANSTYPE originated with the conviction that the problem would be better alleviated if the focus of interaction were shifted from the meaning of the source text to the form of the target text. In TRANSTYPE, a translation emerges from alternating contributions by human and machine, with the translator's inputs serving as progressively informative constraints for the MT component, which would normally respond to each of them with a fresh proposal for all or part of the target text.

This approach, which we call target-text mediated (TTM) IMT, can in principle accommodate a wide range of MT proficiencies and may encompass a number of interesting interaction scenarios. In our current prototype, the machine's task is to try to guess what the translator will type next and display this in the best proposals. A completion menu is proposed after each keystroke. The translator may then choose one item in the menu or continue typing.

Up to now, we have put most of our efforts into devising an adequate statistical translation strategy compliant with strong practical constraints, in order that a completion be proposed after each keystroke typed by a translator (Foster et al., 1997; Langlais and Foster, 2000; Langlais et al., 2000). The result is a prototype which offers a friendly interface to a translator (see figure 1). The current state of TRANSTYPE is described in section 2.

During the development stage, we evaluated several approaches in TRANSTYPE only by considering a theoretical measurement: essentially the number of keystrokes saved by a hypothetical translator which produces the target part of a given test bitext. We describe this issue in section 3.

TRANSTYPE has now reached a stage where it can be evaluated in a more natural and adapted way: that is, asking translators to use it. We have designed an evaluation protocol that encompasses three major steps and which requires around one hour of the subject's time. This protocol is detailed in section 4.

At the time of writing, a group of ten translators have gone through this evaluation protocol. In section 5., we report on both the quantitative and qualitative feedback we gained from analysing this data. In section 6. we compare the theoretical and the in-situs evaluation. Finally, in section 7. we discuss the evaluation we carried out.

2. The TRANSTYPE's prototype

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 $p(t|\tilde{t}, s)$, the probability of a target word t given the preceding target context $\tilde{t}$, and a source segment s. Creating this model means finding some decomposition of $p(t|\tilde{t}, s)$ 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 decoders (Wang and Waibel, 97; Wang and Waibel, 98) 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 $(\tilde{t}, s)$ (Langlais and Foster, 2000). However in the version we used for the present work, the weighting factors have been set up empirically to 0.6 regardless of the context.

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

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

An advantage of our approach is that there are well-established modeling techniques for both distributions in equation 1. 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).

The first few lines of Table 1 give an idea of how TRANSTYPE functions[1]. Let us assume that the user wants to produce the sentence "Ce projet de loi est très semblable au projet de loi que nous avons examiné hier à la chambre des communes" as a translation for the source sentence "*This bill is very similar to its companion bill which we dealt with yesterday in the house of commons*" and suppose that he/she has already typed the first word ce (*this*). The first hypothesis that the system produces before the user enters a character is est (*is*). As this is not a good guess from TRANSTYPE the user types the first character (p) of the words he/she wants as a translation. Taking this new input into account, TRANSTYPE then modifies its proposal so that it is compatible with what the translator has typed. It suggests the desired sequence projet, which the user can simply validate by typing a dedicated key. Continuing in this way, the user and TRANSTYPE aalternately contribute to the final translation.

This bill is very similar to its companion bill which we dealt with yesterday in the house of commons		
	pref.	completions
ce	ce+	/loi · c/'
projet	p+	/est · p/rojet
de	d+	/très · d/e
loi	l+	/très · l/oi
est	e+	/de · e/st
très	t+	/de · t/rès
semblable	se+	/de · s/es · se/mblable
au	au+	/loi · a/vec
projet	p+	/loi · p/rojet
de	d+	/loi · d/e
loi	l+	/nous · l/oi
que	qu+	/nous · q/ui · qu/e
nous	+	/nous
avons	av+	/nous · a/vec · av/ons
examiné	ex+	/hier · e/n · ex/aminé
hier	+	/hier
à la	à+	/hier · à/ la
chambre	+	/chambre
des	de+	/communes · d/e · de/s
communes	+	/communes
106 char.	23 pref. + 20 accept. = **43 keystrokes**	

Table 1: A one-sentence session illustrating the word-completion task. The first column indicates the target words the user intends to produce. The next two columns indicate respectively the prefixes typed by the user and the completions made by the system under a word-completion task. The total number of keystrokes is reported in the last line. + 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 $\cdot$.

3. Theoretical Evaluation

Up to now, we have tackled the evaluation problem by automatically measuring the number of keystrokes saved by a hypothetical user producing the target text of a test bitext. We assume a left-to-right mode in which the user is expected to type the translation sentence by sentence, starting from the left to the right. A completion is proposed automatically by the system after each keystroke. Then the user has two choices: 1) accepting the completion by typing an acceptance key, or 2) ignoring the completion by typing the next character of the word under translation. In (Foster et al., 1997), it was assumed that a translator carefully observes each completion proposed by the system and accepts it as soon as it is correct. This is far too strong a hypothesis and this scenario is only valid in the case of a translator typing very slowly. It is however directly reproducible.

To some extent, we can relax the first scenario by introducing some heuristics that attempt to model a user's behavior. In particular, it is likely that a user will accept a completion that is close enough to the desired string, then make minor changes. For instance, in a unit-completion scenario, we may reasonably hypothesize that a user who wants to produce au projet de loi que (*the bill that*) will accept a close enough completion such as au projet de loi sur (*the bill on*).

[1]For the rest of this presentation, we make use of different fonts for differentiating the kinds of input and output: *italics* are used for the source text, sans-serif for characters typed by the user and typewriter-like for characters completed by the system.

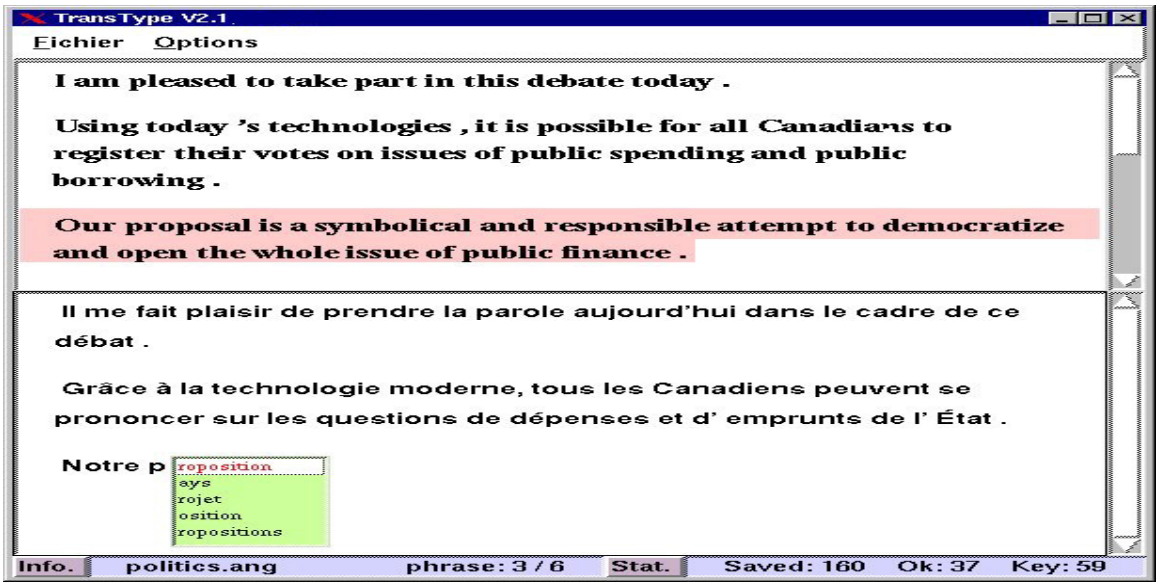

Figure 1: An example of interaction in TRANSTYPE with the source text in the top half of the screen. The target text is typed in the bottom half with suggestions given by the menu at the insertion point.

Formally, let s be the sequence a user wants to produce, p the prefix typed (possibly null), c the completion proposed by the system, y the longest correct part of c and z the incorrect part of c (hopefully null). The cost of the modification of $y.z$ to $p.s$ can be decomposed into two costs: the cost (E) of erasing z, and the cost (A) of adding the missing characters to make s. We designed a special key for deletion whose role is to remove the last word (a sequence of non blank characters) in one keystroke. Table 2 gives some examples of the cost (counted in characters) associated with different completion cases. The rejection of a completion is decided when one of these rules applies:

- the completion length is less than m characters,
- the user has to type more than one word to correct the completion,
- the number of characters to add to the completion is above a threshold M,
- the cost of correcting the completion is higher than the cost of typing the desired completion.

Finally, we evaluate the completion task over a reference bitext of n pairs of sentences $R = \{(R_s^1, R_t^1), (R_s^2, R_t^2), \ldots, (R_s^n, R_t^n)\}$; R_t^i standing for the ith target sentence that contains n_t^i tokens $w_1^i \ldots w_{n_t^i}^i$; by computing the rate:

$$spared = 100 \times \frac{\sum_R (|p| + cost) + accept + sep}{\sum_{i=0}^{n} (\sum_{j=1}^{n_t^i} (|w_j^i| + 1) - 1)} \quad (2)$$

where $|w_j^i|$ stands for the number of characters of the jth token of the ith target sentence to produce, $accept$ stands for the number of times a completion has been accepted (we assume that an acceptance keystroke also adds a separator) and sep is the number of cases where no completion has been proposed for a token that is not the last one in a sentence (the user has to add a separator). $|p|$ is the number of characters typed by the hypothetical user, and $cost$ the cost of possible corrections made on partially good completions that have been accepted.

In the one-sentence session example given in Table 1, $|p| = 23$, $cost = 0$ (all the accepted completions matched exactly the expected words), $accept = 20$ and $sep = 2$ (here, TRANSTYPE did not manage to propose the translations for the words *ce* (this) and *au* (to) before the user finished typing these tokens). The number of characters of the intended target text is 106 (86 plain characters plus 20 separators).

4. The In-situs Evaluation

In the previous section, we presented an automatic evaluation procedure which essentially counts the number of keystrokes saved by a hypothetical translator. This way of gauging our prototype, although easy to run, is somehow questionable. Will a user really read the completions made by TRANSTYPE? If so, will it speed up the translation process? Is speed and/or keystroke saving a good way of evaluating an IMT?

In order to gain a better view of the usability of TRANSTYPE, we decided to pursue its evaluation in a more natural and adapted way: that is, asking translators to use it. In this section, we describe the protocol we have set up as well as the motivations that led us to design it that way.

4.1. The Protocol

We have designed an evaluation protocol that encompasses three major steps and which requires around one hour of the subject's time. This is actually the more we can ask to testers on a voluntary ground.

Before being directly in contact with the interface, the users are first given a demonstration that introduces the

s	p	c	y	z	$cost$
au cours de	a	u cours des cinq	u cours de	s cinq	E(s cinq) = 3
prières	pr	ière	ire	-	A(s) = 1
de la	d	e l'	e l	a	E(') + A(a) = 2
politique	-	politiques	politique	s	E(s) = 1
universités	-	université	université	s	A(s) = 1

Table 2: Examples of costs (counted in characters) associated with partially bad completions. E is the cost of removing z, A the cost of adding the missing characters to produce s.

goals and the major components of TRANSTYPE. General instructions are also given during this introduction. In particular, we emphasize that the translators should not worry about formatting matters, but instead should focus on producing a version which substance would require a normal review. After this 15-minute stage where they can ask any question they want, we assume that the users are ready to use TRANSTYPE by themselves.

The first step of the protocol puts the subjects directly in contact with the text-editor implemented in TRANSTYPE. This editor offers all the standard operations (cut & paste, delete, etc.) that a computer-familiar person may expect. During this period, TRANSTYPE works in a silent mode (i.e. it does not propose anything) and the user only uses the editing functionalities of the prototype. We expect that this 5-minute stage will make the user familiar with the few specific commands TRANSTYPE requires (e.g. selecting a new source sentence to translate).

In the second step, TRANSTYPE is switched to its normal mode, that is, proposing after each keystroke the completion of the current word. This stage which takes 20 minutes is divided in two parts (nominally 2a and 2b). In the first one, we observe the strategy the user is developing using the prototype. In the second part, when appropriate, we give him/her advice for improving his/her use of TRANSTYPE.

The third and last step of the protocol (which takes 5 minutes) is intended to measure how a user may perceive TRANSTYPE if it is able to predict the next several words instead of the only current one. Automatically finding the accurate translations of a group of words is still an open problem that we have partially addressed within this project (Langlais et al., 2000). We did not however included this functionality within TRANSTYPE yet, as we have to think about an adequate way of proposing units to the user without being intrusive. In this stage, we manually introduced sequences of words (called *briskels*) that a translator will likely want to use in its translation. These briskels are provided in a special area of the interface, once after the user selects the source sentence to translate. A briskel may be integrated in the translation simply by clicking on it. An example of briskels is given in figure 3. The evaluation protocol ends up with a 10-minute feed-back survey to collect the subject's feelings and suggestions.

4.2. The material

We put together a corpus of about one hundred isolated sentences chosen from the Hansard corpus. We excluded the sentences that had been used during the training of the language and the translation models and also removed sentences that were too long, contained too many complicated proper names or numbers, etc. Finally, we inspected the selected sentence in order to remove those that we found ambiguous or difficult to translate without larger context (e.g. sentences with ellipses, etc).

le discours du trône
(the throne speech)
trois programmes importants
(three major programs)
le ministère des affaires autochtones
(the aboriginal affairs department)

Table 3: Briskels proposed for the source sentence: *The throne speech presented three major programs that will be undertaken by the aboriginal affairs department.*

Users were asked what they felt about the selected material. All thought that it was globally easy to translate and representative of a realistic translation task, although they sometimes missed the context in which the sentences appeared.

5. TRANSTYPE and its users

In this section we describe and comment on the data we collected from ten users who went through the entire protocol described in the previous section.

5.1. The users

We made use of word of mouth to enroll the volunteers. Thus, all of the users we found have a special interest in testing MT or IMT prototypes. Four of them were either professional translators or professors from the University of Montreal actively involved in teaching translation. The other six are graduate students engaged in a translation program. All of them are very familiar with computers. The testing was carried out over a period of three weeks at the RALI.

5.2. The qualitative Survey

A set of open questions were asked in order to get qualitative feedback from the users. We summarize their answers in the next paragraphs.

1. Is the text-editor used in TRANSTYPE friendly enough?

   The motivation behind this question was to know if any user was disturbed by the text-editor proposed in

TRANSTYPE, thus introducing a bias in the evaluation. None of the users answered no to this question, although four of them mentioned that it was disturbing not to be allowed to erase a selection simply by typing erase[2].

2. Are you satisfied with TRANSTYPE, and would you use it in your day-work?

   This question is important to us, as we implemented only one of the possible scenarios that a TTM IMT prototype may offer to a user. One user said clearly she hates TRANSTYPE and that she would never use such a tool in her work (subject 9 in the following). The nine others expressed in various ways that they liked it and would enjoy using it in their work. They did however mention some points that are developed in the next paragraphs.

3. Do you find that the proposals made by TRANSTYPE disturb you in your translation?

   Three of the nine satisfied users answered negatively. The six others said that the pop-up menu output after each keystroke is somewhat intrusive; especially when they reformulate part of a sentence, in which case they would prefer a dumb prototype [3]. These six users also mentioned that it is difficult to simply ignore the pop-up menu and continue typing the intended translation. They felt however, that the suggestions were "logical" and of great help in special situations (e.g. where they do not know how to translate a word or a term). Furthermore, they also mentioned that being disturbed by TRANSTYPE is not necessarily a drawback: according to some users, it happens often that TRANSTYPE has a positive impact on the quality of the translation, notably by proposing a word that they were not thinking of, or by encouraging the translator to validate when appropriate full words instead of abbreviations they would otherwise use.

4. Do you feel that TRANSTYPE helps you to type faster?

   Five out of ten users answered positively to this question. Two were doubtful, two answered no, but pointed out that it surely is a matter of adaptation to the tool. Subject 9 answered clearly negatively. Interestingly, except for one user (subject 7 in the following), none of the users managed to type faster using the completions. We postpone the discussion of this important point to the next subsection.

5. Do you have any suggestions that would make TRANSTYPE indispensable in your work?

   The users suggested numerous points that could improve the current version of TRANSTYPE. Many of these are just interface considerations that do play a role in TRANSTYPE but which are not crucial from a scientific point of view. Among them, some users suggested that short suggestions (e.g articles, pronouns, etc) should not appear in the pop-up menu.

   Others suggestions were more linguistically motivated. For instance some users noticed that TRANSTYPE does not systematically proposes all the inflections of a given form, thus sometime missing the good one.

   Another problem that some users mentioned derives from the specificity of the Hansard corpus we used to train our models and which has the tendency — according to our users — to contain many anglicisms and calques. TRANSTYPE inherits these problems and therefore tends to induce poor translations.

   Last but not least, all the users (even the one that disliked TRANSTYPE) agreed that they did like the stage where they were given some briskels once a sentence is selected. They indicated however that the best place for these suggestions would naturally be in the pop-up menu. This last observation encourages us in the work we are currently doing to extend the predictions of our translation model (Langlais et al., 2000).

5.3. A quantitative analysis data

All interactions between the user and TRANSTYPE was recorded in a log file during the test. This allows us to get a fairly detailed view of how the user really interacts with TRANSTYPE.

In order to appreciate how TRANSTYPE influences the work of the subjects, we computed two measurements: the **productivity** and the **effort**. The productivity is computed as the typing speed of a subject, that is, the ratio of the characters produced in the translation over the time spend to accomplish it. Especially long pauses (more than 30 seconds without any interaction with TRANSTYPE) were automatically removed from the duration we used in this computation. These long pauses generally occurs when subjects asked questions during the test. In practice, we removed 34 pauses for a total duration of around 27 minutes over a total duration of approximately 6 hours. The effort is the ratio of any action (keystrokes or mouse click) produced over the time spent to translate.

These two rates, measured for each stage of the protocol and for each subject, are reported in figure 2. A first observation we can make from this graph is that all the subjects are not equally fast. During the first stage (that is, without any suggestion), the slowest subject produces less than 68 characters per minute while the fastest subject types his/her translations twice as fast (144 characters per minute). This may reflect both the different typing skills of the translators but also the different translation habits they have.

What also appears on this graph is the clear separation between the first stage and the others. Not surprisingly, each user in step 1 is over the diagonal that we would observe if a translator were translating from left to right without changing the intended version and without producing any typing mistakes. Actually this is partly the assumption we made in the theoretical evaluation protocol we described in section 3..

Somewhat deceptively, this figure also tells us that globally, all the users perform worse when they use TRANSTYPE in its normal mode, that is proposing completions for the current word (stage 2a and 2b). As a matter of

[2] TRANSTYPE now includes this command.

[3] This is of course something easy to implement.

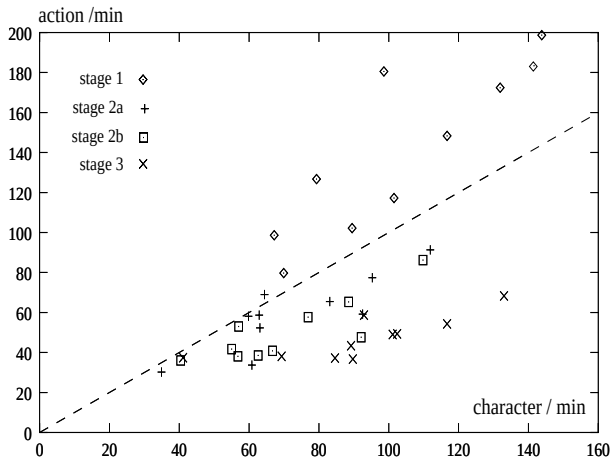

Figure 2: Productivity versus effort of each subject over each stage of the protocol. The x-axis indicates the productivity, that is: the number of characters produced by unit of time (here a minute). The y-axis (the effort) indicates the number of keystrokes (or mouse clicks) produced on average each minute.

fact, the scatter plot which represents the subjects in steps 2 and 3 is globally shifted to the left of the graph.

Looking at the productivity rates of each subject in detail, it turns out that except for subject 7 who managed to outperform stage 1 using the completions, all the other subjects were less productive. The decline in productivity is either smooth or drastic, as for instance for subject 9 whose typing speed was reduced by half. The differences between the two stages (2a and 2b) are either positive or negative, depending on how the users reacted to the instructions we gave them. For instance, some subjects were really disturbed when asked to use the mouse instead of the keyboard to accept the completions. In a way, this also tell us that twenty minutes may be too short to be able to use TRANSTYPE efficiently. For the rest of the data analysis, we decided to regroup stage 2a and 2b in a single stage (stage 2).

We must confess that we were not totally surprised by the foregoing conclusions. First of all, we do not think that the word-completion scenario we tested is the best we can do; rather, we think that a unit-scenario in which phrases would be proposed would be a more natural and more efficient TTM interaction scenario. It is interesting at this point to note that the typing speed measured in the third stage (that is, the one where we simulated a unit completion scenario) is encouraging: 3 subjects outperformed stage 1 in terms of productivity, only a few were significantly slower. We are currently studying such a scenario within TRANSTYPE (Langlais et al., 2000).

What Figure 2 also teaches us is that although the subjects are globally less productive using TRANSTYPE, they are nonetheless spared some effort to produce a translation. Ideally, we would like the dots representing each subject to be shifted to the lower-right corner of the graph. This would indicate that a user is still as productive as usual (or even better) but with less effort. What we do observe here is that in steps 2 and even more in step 3, the users are typing less and less to produce the translation.

Let us define the **efficiency** of a user as the ratio of his/her productivity over the effort. Table 4 reports the average efficiency (all subjects taken together) measured for each stage. It appears that the average gain in efficiency between stage 1 and stage 2, but also between stage 2 and stage 3, is around 0.6. What this means is simply that to produce a translation of let say one hundred characters, a user requires on average 143 actions (keystrokes or mouse clicks) in stage 1. In stage 2, only about 77 actions only are required to produce the same translation. Finally, in stage 3, the user only requires 53 actions.

stage	productivity	effort	efficiency
1	102.1	139.1	0.7
2	72.4	56.4	1.3
3	91.1	47.0	1.9

Table 4: Average productivity, effort and efficiency of all subjects for each stage of the protocol.

5.4. Interpreting the data analysis

Several hypotheses come to mind for explaining the typing speed reduction measured in stages 2 and 3. One plausible explanation may lie in the fact that a user has to perform a task that he/she does not have to do in the first stage, that is, reading the completions! This may disturb the work of a translator more than we expected. Such an assertion is difficult to analyse accurately. We saw however in the previous section, that most users felt TRANSTYPE was helping them to translate faster.

Table 5 helps provide more insight on the interactions between the user and TRANSTYPE. The first columns tell us how often each subject (s) accepted the completions given by TRANSTYPE either by using the validation key or by using the mouse. On average, users accept a word in 28% of the completion menus that the system proposes. The way they validate a completion is variable. For instance, subject 6 mostly used the validation key (88% of the validations) while subjects 1 and 7 never did and preferred mouse clicks. They tend to validate completions that are at least 5-characters long and do so at the very beginning of a word. Actually, most of the time, they do not accept a word proposed if they have already typed its first characters.

Subject 7 (the one who did manage to type faster using TRANSTYPE) had the following strategy: accepting long-enough completions using the mouse. This was in fact the instruction we gave to every one in stage 2b. It allows us to envisage the possibility that enough time, and probably some interface modifications, a user could train and become efficient with TRANSTYPE.

Table 5 also illustrates something interesting regarding the impact TRANSTYPE has on the way a user translates. During stage 1, that is without any completion proposed, the ratio (over the time) of the number of moves or erased characters within the already typed translation is greater than the ratio observed during stage 2. In other words,

	usefulness					extra key	
S	m	%	$\%kb$	c_{lg}	p_{lg}	$r1$	$r2$
1	21.9	33.8	0	5.3	.1	15.4	5.4
2	26.8	31.3	48.0	4.7	.8	32.6	7.9
3	47.2	23.6	48.2	5.2	.6	20.4	9.1
4	31.6	18.2	86.5	5.8	.7	15.4	10.1
5	22.2	43.7	72.2	4.7	.4	6.0	6.6
6	46.3	18.9	88.0	4.8	.8	22.9	5.7
7	54.4	15.7	0	5.4	.3	60.0	11.2
8	33.0	31.7	40.3	4.9	.7	27.0	8.1
9	17.2	29.9	53.6	5.0	0	4.9	6.2
10	27.5	30.6	60.1	4.8	.3	7.8	8.5

Table 5: Usefulness of the completion menu. m is the number of menus proposed per minute, % the acceptance rate of a completion %, $\%kb$ is the percentage of times the user S accepted a completion using the keyboard (vs clicking with mouse). c_{lg} stands for the average length (counted in characters) of the accepted completions and p_{lg} their average prefix length. The two last columns indicate the rates of extra keys (erase, arrows, etc.) used respectively in stage 1 and 2.

TRANSTYPE seems to induce a more left-to-right translation mode. One possible explanation could be that when using a completion, a user does not commit any spelling mistakes, thus lowering the need for revisions and therefore the moves within the current version.

We also analyzed the distribution of the time (counted in seconds) it takes to a user to react (either with the keyboard or the mouse) to a popup menu just output. It is fairly difficult to interpret precisely the figures because we did not trace milliseconds in our log files. But the main observation is that almost all subjects behave on average the same. In less than half the cases, it does not take time before the user reacts. This may indicate that in these cases the user does not consider at all the proposed completions. The others cases show a small delay in response (on average less than 2 seconds). Table 6 reports the average delay between any action (keyboard or mouse) during stage 1 ($m1$) and during stage 2 ($m2$). $m3$ stands for the average response time after a popup menu is output. Normally, a popup menu is output after each keystroke, but it sometimes happens that the system does not have any suggestion to propose, or simply that the user used a keystroke to navigate in the current translation, in which case a popup menu is not systematically proposed.

6. Comparing theoretical and in-situs Evaluation

One goal of our investigation was to measure how representative the theoretical evaluation protocol we described in section 3. is. In other words, does it correlate with the in-situs observations we collected?

Table 7 describes the theoretical gain (as computed in section 3.) that a hypothetical user will obtain if he or she carefully watches every suggestion made by TRANSTYPE and the system tries to produce the translation that each subject really produced in stage 2. For comparison purposes, we also report the in-situs efficiency measured for each subject in stage 2.

S	$m1$	$m2$	$m3$	S	$m1$	$m2$	$m3$
1	0.6	1.1	2.7	6	0.3	0.6	1.1
2	0.5	0.9	1.8	7	0.4	0.5	1.2
3	0.4	0.5	1.3	8	0.3	0.6	1.4
4	0.4	0.9	1.8	9	0.8	1.6	2.5
5	0.8	0.9	2.5	10	0.5	0.5	1.6

Table 6: Average delay between any action (keyboard or mouse) during stage 1 ($m1$) and during stage 2 ($m2$). $m3$ stands for the average response time (counted in seconds) after a popup menu is output.

	theoretical				in situs		
S	tok	$char$	tp	$spare$	pro	e_1	e_2
1	149	795	318	60.0	59.5	35.3	1.7
2	162	821	364	57.7	62.8	48.4	1.3
3	362	1787	744	58.4	103.9	73.6	1.4
4	205	1023	456	55.4	58.5	55.7	1.0
5	191	905	362	60.0	64.1	49.4	1.3
6	258	1213	532	56.1	85.6	65.5	1.3
7	290	1478	587	60.3	98.5	79.4	1.2
8	335	1629	686	57.9	85.0	58.5	1.5
9	91	495	203	59.0	37.8	33.2	1.1
10	239	1163	443	61.9	59.9	55.9	1.1

Table 7: Theoretical versus in-situs data. tok is the number of target tokens produced by each subject S, $char$ is the length counted in characters of the translation produced. tp stands for any keystroke needed to produce the translation within the theoretical scenario and $spared$ indicates the percentage of keystrokes saved. The three last columns indicates respectively the productivity (pro), the effort (e_1) and the efficiency (e_2) really observed.

First, we observe that on average TRANSTYPE is fairly stable over the different translations the subjects produced: in a perfect world, a user would have to type only slightly more than a fourth of the characters he/she planned to type. In practice, things turn out differently and the correlation we measure between the theoretical gain (x) and the efficiency measurement (y) is only of 0.12.

This discrepancy may be explained in many ways. First, the left-to-right scenario is not entirely realistic even if TRANSTYPE tends to induce that way of translating. Second, our prototype offers the user the possibility of selecting (by a mouse-click) a prediction which is not rated first but is still well ranked (the seven first completions are proposed). This is not something our theoretical scenario accounts for. If it did, the gain in terms of actions would be much more significant. More importantly, we feel after the data analysis we carried out, that it is currently too difficult to find a model that can predict whether or not a user will accept a completion. Many factors influence this decision including some some the implementation choices we made in our prototype that disturb a novice user, but also the fac-

t that the user did not really have time to become familiar with the interface, and so could not develop a good working strategy.

7. Discussion

We have presented two protocols to evaluate TRANSTYPE, a target-text mediated interface. The theoretical one — for a given bitext — measures the potential gain of the keystrokes needed to type a translation. It relies on a simplistic model of a user who carefully observes each completion and accepts it when appropriate.

Second, we set up an in-situs protocol involving translators. The goal was twofold: first, to verify that our theoretical evaluation holds and second, to have some concrete feedback on how TRANSTYPE is perceived by the users it is designed for.

Several points emerged from this study. First, we were pleased to see than 9 out of 10 subjects really liked TRANSTYPE and would like to work with it in they daily work. They made some useful suggestions, many of which are implementation considerations. Although they feel they translate faster using TRANSTYPE, our prototype has a negative impact on the productivity of the subjects we tested. The fact that at least one subject still managed to use the tool without losing time allows us conjecture that this could be due to a under-training problem. We also noticed that when we simulated a unit completion scenario, the users did become more productive.

The decrease observed in productivity is probably due to the fact that the user is burdened by the popup menu, which is not so easy to process efficiently. On the other hand, as expected, the number of actions (keystrokes or mouse clicks) needed to produce a given translation is significantly reduced when using TRANSTYPE: users do use the proposed completions. However, it is difficult at this stage to clearly identify the strategies that they developed and the one that would most increase productivity. Obviously, we need more data from subjects who are using TRANSTYPE intensively to be conclusive.

The comparison of the two evaluation protocols (theoretical and in-situs) shows that there is not a strong correlation between the two, the main reason for this being that it is not easy to model when a user will accept a proposed completion. Many factors influence this decision, one being that users do not always watch the screen while typing. Extending the predictive power of TRANSTYPE to longer units will surely rationalize its use and should therefore reduce the gap between the two evaluation protocols.

Acknowledgments

TRANSTYPE is a project funded by the Natural Sciences and Engineering Research Council of Canada. We are indebted to Pierre Isabelle for the fruitful orientation he gave to this work. We also want to warmly thank the subjects who have participated in this study and from whom we have gained much insight.

8. References

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

Brown, Peter F., 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.

Brown, Ralf D. and Sergei Nirenburg, 1990. Human-computer interaction for semantic disambiguation. In *Proceedings of the International Conference on Computational Linguistics (COLING)*. Helsinki, Finland.

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

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

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

Langlais, Ph. and G. Foster, 2000. Using context-dependent interpolation to combine statistical language and translation models for interactive mt. In *Content-Based Multimedia Information Access (RIAO)*. Paris, France.

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

Maruyama, Hiroshi and Hideo Watanabe, 1990. An interactive Japanese parser for machine translation. In *Proceedings of the International Conference on Computational Linguistics (COLING)*. Helsinki, Finland.

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

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

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

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

Whitelock, P. J., 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 *Proceedings of the International Conference on Computational Linguistics (COLING)*. Bonn, West Germany.