Machine Translation **15:** 267–294, 2000.

Unit Completion for a Computer-aided Translation Typing System

PHILIPPE LANGLAIS, GEORGE FOSTER and GUY LAPALME
RALI/DIRO, Université de Montréal, C.P. 6128, succursale Centre-ville, H3C 3J7 Montréal, Canada
(E-mail: {felipe,foster,lapalme}@iro.umontreal.ca)

Abstract. This work is in the context of TRANSTYPE, a system that watches over the users as they type a translation and repeatedly suggests *completions* for the text already entered. The users may either accept, modify, or ignore these suggestions. We describe the design, implementation, and performance of a prototype which suggests completions of units of texts that are longer than one word.

Key words: interactive machine translation, machine-aided human translation, statistical language models, statistical translation models, target-text mediation, word completion

1. Introduction

TRANSTYPE is 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, e.g., Blanchon (1991), Brown and Nirenburg (1990), Maruyama et al. (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 translators of the burden of having to provide explicit analyses of the source text and allow them to translate naturally, assisted by the machine whenever possible.

In TRANSTYPE, 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 translators remain directly in control of the process: the machine must respect the constraints implicit in their contributions, and they are free to accept, modify, or completely ignore its proposals.

In this paper, we treat the problem of finding an appropriate word sequence (called here a "unit") to follow a particular position in the target text given the corresponding source text. The target text is thus the result of a mix of contributions from both the source text and the partial translation already written and/or accepted by the translator. We first present a theoretical model of the task and the research we carried out for modeling units in both source and target texts. We also give the results of an evaluation of the performance of the system implementing this model.

2. TRANSTYPE and Its Model

2.1. USER VIEWPOINT

Our interactive translation system is illustrated in Figure 1 for an 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. Note that, throughout this process, the translator remains in control, and the machine must continually adapt its suggestions to the translator's input. This differs from the usual MT set-ups where it is the machine that produces the first draft which then has to be corrected by the translator.

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 form of target-text mediation is within the reach of current MT technology. The user-interface design choices and a more formal evaluation within the global task of translation will be the subject of another paper.

The first version of TRANSTYPE (Foster et al., 1997) proposed completions only for the current word. This paper deals with predictions which extend to the next several words in the text. The potential gain from multiple-word predictions can be appreciated in the one-sentence translation task reported in Table I, where a hypothetical user saves over 60% of the keystrokes needed to produce a translation in a word-completion scenario, and about 75% in a unit-completion scenario.

2.2. SYSTEM VIEWPOINT

The core of TRANSTYPE is a completion engine which comprises two main parts: an "evaluator" which assigns probabilistic scores to completion hypotheses and

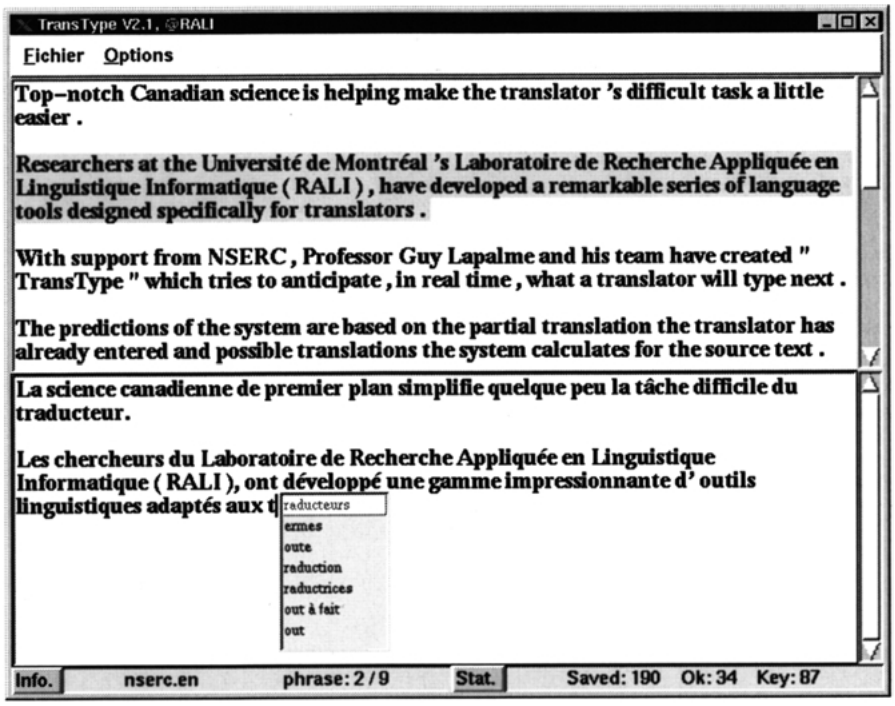

Figure 1. Example of an interaction in TRANSTYPE with the source text in the top half of the screen. The target text is typed in the bottom half with suggestions given by the menu at the insertion point.

a "generator" which uses the evaluation function to select the best candidate for completion.

2.2.1. *The Evaluator*

The evaluator is a function $p(t|t', s)$ which assigns to each target-text unit t an estimate of its probability given a source text s and the tokens t' which precede t in the current translation of s.[1] Our approach to modeling this distribution is based to a large extent on that of the IBM group (Brown et al., 1993), but it differs in one significant aspect: whereas the IBM model involves a "noisy channel" decomposition, we use a linear combination of separate predictions from a language model $p(t|t')$ and a translation model $p(t|s)$. Although the noisy-channel technique is powerful, it has the disadvantage that $p(s|t', t)$ is more expensive to compute than $p(t|s)$ when using IBM-style translation models. Since speed is crucial for our ap-

Table I. A one-sentence session illustrating the word- and unit- completion tasks. We use 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. 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 by the system under a word-completion task. The last two columns provide the same information for the unit-completion task. The total number of keystrokes for both tasks 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 $\cdot$.

This bill is very similar to its companion bill which we dealt with yesterday in the house of commons

	Word-completion task		Unit-completion task	
	Pref.	Completions	Pref.	Completions
ce	ce+	/loi · c/'	c+	/loi · c/e projet de loi
projet	p+	/est · p/rojet	-	
de	d+	/très · d/e	-	
loi	l+	/très · l/oi	-	
est	e+	/de · e/st	e+	/de · e/st
très	t+	/de · t/rès	t+	/de · t/rès
semblable	se+	/de · s/es · se/mblable	se+	/de · s/es · se/mblable
au	au+	/loi · a/vec	a+	/loi · a/u projet de loi sur
projet	p+	/loi · p/rojet	-	
de	d+	/loi · d/e	-	
loi	l+	/nous · l/oi	-	
que	qu+	/nous · q/ui · qu/e	qu+	/nous · q/ui · qu/e
nous	+	/nous	+	/nous
avons	av+	/nous · a/vec · av/ons	av+	/nous · a/vec · av/ons
examiné	ex+	/hier · e/n · ex/aminé	exa+	/à la chambre des communes e/n · ex/istence · exa/miné
hier	+	/hier	h+	/à la chambre des communes h/ier
à la	à+	/hier · à/ la	+	/à la chambre des communes
chambre	+	/chambre	-	
des	de+	/communes · d/e · de/s	-	
communes	+	/communes	-	
106 char.	23	20 accept. **43 keystrokes**	14	11 accept. + 1 correc. **26 keystrokes**

plication, we chose to forego it in the work described here. Our linear combination model is described as in (1),

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

where $\lambda(\Theta(t', s)) \in [0, 1]$ are context-dependent interpolation coefficients. $\Theta(t', s)$ stands for any function which maps t', s into a set of equivalence classes. Intuitively, $\lambda(\Theta(t', s))$ should be high when s is more informative than t' and low otherwise. For example, the translation model could have a higher weight at the start of sentence but the contribution of the language model can become more important in the middle or the end of the sentence.

2.2.2. *The Language Model*

We experimented with various simple linear combinations of four different French language models: a cache model, similar to the cache component in Kuhn and De Mori's (1990) model; a unigram model; a triclass model (Derouault and Merialdo, 1986); and an interpolated trigram (Jelinek, 1990).

We finally opted for an interpolated trigram alone, in which the probability of a token depends directly on the previous two. The trigram was trained on the Hansard corpus, with 75% of the corpus used for relative-frequency parameter estimates, and 25% used to reestimate interpolation coefficients.

2.2.3. *The Translation Model*

Our translation model is based on the linear interpolation given in (2) which combines predictions of two translation models — M_s and M_u — both based on an IBM-like model 2 (see (3)). M_s was trained on single words and M_u, described in Section 3, was trained on both words and units.

$$p(t|s) = \underbrace{\beta.p_s(t|s)}_{\text{word}} + \underbrace{(1 - \beta).p_u(t|\mathcal{G}(s))}_{\text{unit}} \tag{2}$$

where p_s and p_u stand for the probabilities given respectively by M_s and M_u. $\mathcal{G}(s)$ represents the new sequence of tokens obtained after grouping into units the tokens of s. The grouping operator $\mathcal{G}$ is illustrated in Table II and described in Section 3.

Both models are based on IBM translation model 2 (Brown et al., 1993) which has the property that it generates tokens independently. The total probability of the ith target-text token t_i is just the average of the probabilities with which it is generated by each source-text token s_j; this is a weighted average that takes the distance from the generating token into account (3),

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

where $p(t_i|s_j)$ is a word-for-word translation probability, $|s|$ is the length (counted in tokens) of the source segment under translation s, and $a(j|i, |s|)$ is the *a priori* alignment probability that the target-text token at position i will be generated by the source-text token at position j; this is equal to a constant value of $1/(|s| + 1)$ for model 1. This formula follows the convention of Brown et al. (1993) in letting s_0 designate the null state. We modified IBM model 2 to account for invariant entities such as English forms that almost invariably translate into French either verbatim or after having undergone a predictable transformation, e.g., numbers or dates (Foster et al., 1997). These forms are very frequent in the Hansard corpus.

2.2.4. *Mixing Translation and Language Models*

The problem posed by an interpolated model such as (1) is to find features of t', s that indicate which component—language or translation—will be a better predictor of t in a particular context.

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 mock 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 that was chosen so as to optimize the completion performance over a test corpus (that is $\lambda(\Theta(t', s)) = 0.6$) (4).

$$\hat{t} = \begin{cases} t_e, & \text{if } \operatorname{argmax}_t p(t|s) = t_e \text{ or } \operatorname{argmax}_t p(t|\tilde{t}) = t_e \\ \operatorname{argmax}_t 0.6p(t|s) + 0.4p(t|\tilde{t}), & \text{else} \end{cases} \tag{4}$$

The results of this experiment indicated that the global performance of TRANSTYPE[2] can be improved by a maximum of approximately 3.7% over the baseline. For the user, this represents a reduction of 12.3% in the number of keystrokes, with better predictions (shorter prefixes required) for 19% of words. Providing 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.

As suggested by the linear form of (1), we used the EM algorithm to estimate optimal weighting coefficients for each candidate mapping $\Theta(\tilde{t}, s)$ (the two source predictions being constant), maximizing in an iterative process the probability assigned by p to a training corpus (Langlais and Foster, 2000). We tested several source-based and target-based Θ-functions without obtaining a noticable improvement in the baseline performance. The main reason for this disappointing result is that we are faced with a local consistency problem: Lowering the weight on the language model in a specific context may introduce ungrammatical sequences. Conversely, raising the language model weight will favor the tokens that follow more frequently in the training corpus, the conditioning context, even if the source text contains overwelming evidence against them. We found that the more contexts

we consider, the more these consistency breaks arise. This problem can be considered as an over-training one which is largely avoided with the baseline model, where only one weight is assigned globally to balance the two prediction sources.

2.3. THE GENERATOR

The task of the generator is to identify units matching the current prefix typed by the user, and pick the best candidate using the evaluation function. Given the real-time constraints of an IMT system, we designed some special features described by Foster et al. (1997). We focus here on the division of the French vocabulary into two parts: a small "active" component whose contents are always searched for a match to the current prefix, and a much larger "passive" part which comes into play only when no candidates are found in the active vocabulary. Both vocabularies are coded as tries.

The passive vocabulary is a large dictionary containing over 380,000 word forms. The active part is computed dynamically when a new sentence is selected by the translator. It relies on the fact that a small number of words account for most of the tokens in a text. It is composed of a few entities (tokens and units) that are likely to appear in the translation. Formally, if $s = s_1 \dots s_n$ is the sequence of the n source tokens to be translated, and $s' = \mathcal{G}(s) = s'_1 \dots s'_{n'}$ the same sequence of words recast by the grouping operator $\mathcal{G}$, the active vocabulary A is computed by keeping the N best target words (argmax_{N}) and the N' best target ($\text{argmax}_{\text{N}'}$) without any contribution from the alignment probabilities (that is, considering both M_s and M_u as IBM-like model 1s) (5).

$$\begin{aligned} A &= \operatorname*{argmax}_{\text{N}}{}_{t\in\tau} \sum_{j=1}^{n} p(t|s_j) \bigcup \operatorname*{argmax}_{\text{N}'}{}_{t\in\tau'} \sum_{j=1}^{n'} p(t|s'_j) \\ \tau &= \{t \in V_s : \exists\, i \in [1,n] / p(t|s_i) \neq 0\} \\ \tau' &= \{t \in V_u : \exists\, i \in [1,n'] / p(t|s'_i) \neq 0\} \end{aligned} \tag{5}$$

where τ (τ') stands for the set of all possible target words (units) that have a non-null translation probability of being translated by some source token (unit).

In practice, we found that keeping 500 words and 50 units yields good performance. Table II presents the first ten tokens and the first ten units of the active vocabulary computed from a source sentence.

3. Modeling Unit Associations

The main drawback of our translation model is the independence assumption: i.e. the generation of a target word does not depend on the previously generated words. Therefore designing a translation model that partially overcomes this assumption is of interest. In this section, we report on the experiments we performed to address this problem.

Table II. An illustration of the role of the generator for a pair of sentences t being the translation of s in our corpus. $\mathcal{G}(s)$ is the sequence of source tokens recast by the grouping operator $\mathcal{G}$. A_s indicates the ten best tokens according to the word model, A_u the ten best units according to the unit model.

s	*that · is · what · the · prime · minister · said · , · and · i · have · outlined · what · has · happened · since · then · .*
t	c' · est · ce · que · le · premier · ministre · a · dit · , · et · j' · ai · résumé · ce · qui · s' · est · produit · depuis · .
$\mathcal{G}(s)$	*that is what · the prime minister said · , and i · have · outlined · what has happened · since then · .*
A_s	, · . · est · ce · ministre · que · et · a · premier · le
A_u	ce qui s' est produit · et je · c' est ce que · voilà ce que · qu' est · c' est · , et · le premier ministre disait

Automatically identifying which source words or groups of words will give rise to which target words or groups of words is a fundamental problem which remains open. In this work, we decided to proceed in two steps: (a) monolingually identifying groups of words that would be better handled as units in a given context, and (b) mapping the resulting source and target units.

We do not claim that this is the best technique, but it will serve as a baseline for more elaborate approaches discussed in Section 6. Furthermore, this approach could be reused for other applications such as finding correspondances in non-parallel corpora (Tanaka and Iwasaki, 1996; Rapp, 1999; Tanaka and Matsuo, 1999) or in non-aligned parallel corpora (Ohomori and Higashida, 1999).

3.1. THE TRAINING CORPUS

For the rest of this work, we used a segment of the Hansard corpus consisting of 15,377 pairs of sentences, totaling 278,127 English tokens (13,543 forms) and 292,865 French tokens (16,399 forms).

3.2. FINDING MONOLINGUAL UNITS

Finding relevant or salient units in a text has been explored in many areas of natural language processing. Our approach relies on distributional and frequency statistics computed on each sequence of words found in a training corpus. For the sake of efficiency, we used the suffix array technique to get a compact representation of our training corpus. This method allows the efficient retrieval of arbitrary length n-grams (Nagao and Mori, 1994; Haruno et al., 1996; Ikehara et al., 1996; Shimohata et al., 1997; Russell, 1998).

The literature abounds in measures that can help to decide whether word co-occurrences are linguistically significant or not. In this work, we used a likelihood-based test which is more reliable than other metrics when faced with rare events (Dunning, 1993).

More precisely, the score associated with a sequence of words $w_1^n = w_1, \ldots, w_n$ is computed as the minimum value of the likelihood test obtained by considering all possible binary cuts in the string, as described in (6), where ℓ stands for the likelihood ratio given by (6). Intuitively, parts of a sequence of words that should be considered as a whole should not appear often by themselves.

$$\begin{aligned} \rho(w_1, \ldots, w_n) &= \operatorname*{argmin}_{i \in [1,n]} \ell(w_1^i, w_{i+1}^n) \\ \ell(x, y) &= h(a) + h(b) + h(c) + h(d) + h(n) - h(a+b) \\ &\quad -h(a+c) - h(b+d) - h(c+d) \\ h(k) &= k \log(k) \end{aligned} \tag{6}$$

where a, b, c, d and n are the cells of the classical contingency table representation, that is:

- a is the number of times y follows x (in the training corpus)
- b is the number of times x is not followed by y
- c is the number of times y is not preceded by x
- d is the number of times neither x nor y appear
- n is the number of words in the training corpus (normalized)

It takes only a few seconds to compute the likelihood scores of all the sequences found in the training corpus. Table III reports the ten best-rated sequences, the last four, and 20 that are in between. Clearly not all highly ranked sequences are relevant from a linguistic point of view. Many sequences overlap with each other, and the likelihood test alone does not handle them properly. This is partly due to the fact that long sequences are penalized by the minimization operation over all the binary cuts in the sequence: the longer the sequence, the more likely a cut will lower the likelihood value.

Fortunately, other measurements may be considered to filter out sequences automatically. Intuitively, the strength of a sequence of words may be assessed by the fact that a salient unit should appear in various contexts. Therefore, following Shimohata (1997), the computation of an entropy-based measure at the left and right boundaries of a unit should provide a clue for filtering purposes (7),

$$\begin{aligned} e(w_1^n) &= \frac{(e_{\text{left}}(w_1^n) + e_{\text{right}}(w_1^n))}{2} \\ e_{\text{left}}(s) &= \sum_{w | ws \in T} h\left(\frac{\text{freq}(ws)}{\text{freq}(s)}\right) \\ e_{\text{right}}(s) &= \sum_{w | sw \in T} h\left(\frac{\text{freq}(sw)}{\text{freq}(s)}\right) \end{aligned} \tag{7}$$

Table III. A few of the 81,974 sequences appeared at least two times in the training corpus. The first and last divisions show repectively the ten best and the last five sequences, as ranked by the likelihood test; the middle part gives 20 in-between units. The last line indicates the average and standard deviation of each metric.

Rank	$\rho(s)$	$e(s)$	$f(s)$	$l(s)$	$s = w_1^l$
1	3959.51	0.79	936	2	mr. speaker
2	2968.54	1.85	607	2	hon. member
3	2877.13	3.07	717	2	) :
4	2403.29	1.88	789	3	mr. speaker ,
5	2268.73	1.63	855	2	speaker ,
6	2163.60	0.00	376	3	some hon. members
7	2115.39	0.79	559	3	: mr. speaker
8	2044.73	2.14	424	4	) : mr. speaker
9	1980.20	1.98	559	4	: mr. speaker ,
10	1854.59	0.68	423	2	hon. members
772	113.32	0.23	16	2	aviation safety
803	109.19	1.14	15	3	aviation safety board
970	95.87	2.47	37	3	there will be
971	95.76	1.08	17	2	age pension
972	95.65	2.8	23	3	has not been
972	95.56	3.34	95	3	the government is
974	95.51	1.08	28	2	to amend
975	95.48	1.69	14	3	private members' business
1017	92.02	0.24	15	3	canadian aviation safety
1121	85.69	1.13	14	4	canadian aviation safety board
2925	45.52	0.00	6	9	deputy prime minister , president of the privy council
2926	45.51	0.00	5	2	ronald j.
2927	45.51	0.00	33	3	united states .
2928	45.48	2.62	22	4	, and that is
2929	45.45	2.01	15	2	our children
2930	45.43	1.28	5	3	chamber of commerce
5026	32.62	1.99	13	5	the canadian aviation safety board
11941	17.95	0.80	4	3	i am wondering
42931	7.02	0.00	2	8	my question is addressed to the minister
52932	5.56	0.69	2	5	the lack of commitment to
81969	$1.60e-06$	0.69	2	3	the government were
81970	$1.24e-06$	1.38	7	2	the work
81972	$6.92e-07$	1.61	7	2	canada for
81973	$5.28e-07$	0.69	2	2	on other
81974	$1.97e-07$	0.69	2	2	for china
μ	13.3	0.30	4.9	4.2	
ρ	46.6	0.40	18.3	3.0	

where $e_{\text{left}}(s)$ ($e_{\text{right}}(s)$) is null when only one form follows (precedes) s in all possible occurrences of s in T. It is maximal when there are exactly freq(s) forms following (preceding) the freq(s) occurrences of s, where freq(s) stands for the frequency of the sequence s in the training corpus. This is the metric reported in the third column of Table III.

Using this metric, the sequence *aviation safety* can be removed from the list, considering that the sequence *aviation safety board*, which is rated worse by the likelihood test, has a higher entropy score. As a matter of fact, *aviation safety* appears 16 times in our training corpus, but only once not followed by the word *board*, thus indicating that *aviation safety board* may better be considered as a single unit.

We implemented a cascade filtering strategy based on the likelihood score ρ, the frequency f, the length l and the entropy value e of the sequences. A first filter ($\mathcal{F}_1(l_{\min}, f_{\min}, \rho_{\min}, e_{\min})$) removes any sequence s for which $l(s) < l_{\min}$ or $\rho(s) < \rho_{\min}$ or $e(s) < e_{\min}$ or $f(s) < f_{\min}$. A second filter ($\mathcal{F}_2$) removes sequences that are included in preferred ones.

Precision and recall of this filtering procedure cannot easily be measured as we do not have a reference (i.e., a bilingual unit lexicon) for that. However in Section 4.2, we discuss their impact on the completion task within TRANSTYPE.

In terms of sequence reduction, applying $\mathcal{F}_1(2, 2, 5.0, 0.2)$ on the 81,974 English sequences of at least two tokens seen at least twice in our training corpus, less than 50% of them (39,093) were filtered: 17,063 (21%) were removed because of their low entropy value, 25,818 (31%) because of their low likelihood value.

3.3. THE MAPPING

We collected automatically and monolingually a list of units for each language. The mapping of the identified units is achieved by the same EM algorithm used for training our translation model (Brown et al., 1993). This first requires merging the words of our training corpus into sequences, called the sequence-based corpus. This step would be straightforward if the list of salient sequences obtained in the previous step did not contain overlaps. Dealing with overlaps requires some way of defining automatically what constitutes a good unit.

3.3.1. *The Grouping Operator*

The transformation of the initial corpus into a sequence-based corpus has been achieved using a dynamic programming scheme optimizing a criteria $\mathcal{C}$ given by (8).

$$\begin{aligned} \text{Best}(i) &= \underset{I \in [1,i] \,/\, w_{i-I}^{i} \in \mathcal{S}}{\text{argmax}} \left(\mathcal{C}(w_{i-I}^{i}) + \text{Best}(i - I - 1)\right) \\ \text{Best}(0) &= 0 \end{aligned} \tag{8}$$

where $\mathcal{S}$ is the set of all units collected for a given language plus all single words.

We investigated several $\mathcal{G}$-operators (9)–(12), and we found (10) the most satisfactory (this observation was correlated by results obtained on the unit-completion task described in the next section). Table IV reports several outputs of the grouping process, when considering different $\mathcal{G}$-operators: $\mathcal{G}_f$, $\mathcal{G}_l$, $\mathcal{G}_\rho$ and $\mathcal{G}_n$, which are respectively frequency (9), unit-length (10), likelihood-based (11) and number-based (12) operators. For instance, $\mathcal{G}_n$ will group words into the maximal number of units, given a unit lexicon. For comparison, we also report a simple grouping scheme, which groups the largest unit in the lexicon by a left-to-right processing.

$$\mathcal{G}_f(w_i^j) = \begin{cases} \text{freq}(w_i^j) \text{ if } j > i \\ 0 \text{ else} \end{cases} \tag{9}$$

$$\mathcal{G}_l(w_i^j) = \begin{cases} j - i + 1 \text{ if } j > i \\ 0 \text{ else} \end{cases} \tag{10}$$

$$\mathcal{G}_\rho(w_i^j) = \begin{cases} \rho(j - i + 1) \text{ if } j > i \\ 0 \text{ else} \end{cases} \tag{11}$$

$$\mathcal{G}_n(w_i^j) = \begin{cases} 1 \text{ if } j > i \\ 0 \text{ else} \end{cases} \tag{12}$$

It should be emphasized that grouping units monolingually is not an optimal solution. As a matter of fact, there is no strong evidence that the grouping process is bijective (or injective): A source unit in a sentence is not necessarily associated with a target unit in its target counterpart, and vice versa. The mapping process may thus be directly affected by this operation. For example in Table IV, the source and target units found using the frequency-based criteria ($\mathcal{G}_f$), *mr. speaker* has been grouped into a single source unit, while its target counterpart *monsieur le président* as been separated into two units. Some mapping procedure may properly handle this configuration, e.g., IBM model 3 and later (Brown et al., 1993).

3.3.2. *Training Unit Models*

Once the corpus of tokens has been transformed into a sequence-based corpus, there are several ways of estimating the parameters of the unit model; we tried three of them. In the first one, $\mathcal{E}_1$, the translation parameters are estimated by applying the standard EM algorithm on all entities (tokens and units) present at least twice in the sequence-based corpus (the entities seen only once are mapped to a special "unknown" word). The two next methods filter the probabilities obtained with the $\mathcal{E}_1$ method. In $\mathcal{E}_2$, all probabilities $p(t|s)$ are set to 0 whenever s is a token (that is not a unit), thus forcing the model to contain only associations between source units and target entities (tokens or units). In $\mathcal{E}_3$ any parameter of the model that involves a token is removed (that is, $p(t|s) = 0$ if t or s is a token). The resulting model will thus contain only unit associations. In both cases, the probabilities are renormalized.

Of course, the method chosen has a direct impact on the number of parameters for our unit model. For instance, the model obtained by grouping our training

Table IV. Some differences in the grouping of words into units. Units are separated by · . Words within units are delimited by a space, punctuations are also considered as words.

Operator	Grouping output
left-to-right, larger-first	*from time to time , · mr. speaker , the · rcmp · launches · investigations · in canada .* de temps à autre, · monsieur le président , · la gendarmerie royale du canada · fait des · enquêtes · au canada .
$\mathcal{G}_f$ (9)	*from time to time , · mr. speaker · , the · rcmp · launches · investigations · in · canada .* de temps à autre , · monsieur le · président , · la · gendarmerie royale · du canada · fait des · enquêtes · au canada · .
$\mathcal{G}_l$ (10)	*from time to time , · mr. speaker , · the rcmp · launches · investigations · in canada* de temps à autre , · monsieur le président , · la gendarmerie royale du canada · fait · des enquêtes · au canada .
$\mathcal{G}_\rho$ (11)	*from time to time , · mr. speaker · , · the rcmp · launches · investigations · in canada · .* de temps à autre , · monsieur le président · , la · gendarmerie royale · du canada · fait · des enquêtes · au canada · .
$\mathcal{G}_n$ (12)	*from time to time · , mr. · speaker , · the rcmp · launches · investigations · in · canada .* de temps à autre · , monsieur · le président · , la · gendarmerie royale · du canada · fait · des enquêtes · au · canada .

corpus using the length criterion $\mathcal{G}_l$ (10) on a lexicon of units whose likelihood score is above 5.0 and whose entropy score is above 0.2 (that is, applying the filter $\mathcal{F}_1(2, 2, 5, 0.2)$) has 1,071,127 parameters under method $\mathcal{E}_1$, 567,630 under $\mathcal{E}_2$ and 349,258 under $\mathcal{E}_3$.

Table V shows some entries of a unit model (M_u) obtained after 15 iterations of the EM algorithm on a sequence corpus resulting from the application of the length-grouping criteria $\mathcal{G}_l$ (10) on a lexicon of units whose likelihood score is above 5.0. The probabilities have been obtained by application of the method $\mathcal{E}_2$. We found

Table V. Examples of bilingual associations obtained after 15 iterations. The second column indicates a source unit, the third its frequency in the training corpus. The fourth column reports its three best-ranked target associations (α being a token or a unit, p being the translation probability).

	Source unit (s)	$f(s)$	Target units ([α, p])
1	the government	3061	[le gouvernement, 0.56] [du gouvernement, 0.12] [, le gouvernement, 0.10]
2	we have	1748	[nous, 0.49] [avons, 0.41] [, nous avons, 0.07]
3	i think	1061	[je pense, 0.19] [je crois, 0.15] [je pense que, 0.12]
4	we must	720	[nous devons, 0.61] [il faut, 0.19] [nous, 0.14]
5	this bill	640	[ce projet de loi, 0.35] [projet de loi ., 0.21] [projet de loi, 0.18]
6	british columbia	512	[colombie-britannique, 0.31] [de la colombie-britannique, 0.31] [en colombie-britannique, 0.23]
7	people of	429	[les habitants, 0.19] [la population, 0.18] [de, 0.15]
8	the acting speaker (mr. paproski) :	293	[le président suppléant (m. paproski) :, 0.95] [le président, 0.02] [suppléant (m. paproski) : la chambre est elle, 0.02]
9	people of canada	282	[les canadiens, 0.26] [des canadiens, 0.21] [la population, 0.07]
10	we cannot	270	[nous ne pouvons, 0.40] [pas, 0.24] [nous ne, 0.11]
11	mr. speaker :	269	[m. le président :, 0.80] [a, 0.07] [à la, 0.06]
12	notwithstanding clause	202	[clause de dérogation, 0.29] [clause, 0.26] [la, 0.16]
13	what is happening	190	[ce qui se passe, 0.21] [ce qui se, 0.16] [et, 0.15]
14	of course ,	178	[évidemment, , 0.26] [naturellement, 0.08] [bien sûr, 0.08]
15	in my constituency	178	[dans ma circonscription, 0.35] [circonscription, 0.26] [ma, 0.11]
16	high interest rates	138	[taux d' intérêt élevés, 0.39] [des, 0.14] [à, 0.12]
17	over the years	136	[au fil des, 0.18] [au cours, 0.18] [des années, 0.17]
18	the first time	117	[la première fois, 0.51] [c' était, 0.11] [c' est, 0.07]
19	minister of indian affairs and northern development	43	[ministre des affaires indiennes et du nord canadien, 0.29] [donnée, 0.10] [classe, 0.10]
20	some hon. members :	37	[des voix :, 0.93] [!, 0.07]
21	is it the pleasure of the house to adopt the	14	[plaît-il à la chambre d' adopter, 0.49] [la motion ?, 0.42] [motion ?, 0.04]

many partially correct associations (*over the years*|*au fils des*, *we have*|*nous*, etc.)[3] that illustrate the weakness of decoupling the unit identification from the mapping problem. In most cases however, these associations have a lower probability than the good ones. We also found few erratic associations (*the first time*|*c'était*, *some hon. members*|*!*, etc.) due to distributional artifacts. It is also interesting to mention that the good associations we found are not necessarily compositional in nature (*we must*|*il faut* lit. 'it is necessary', *people of canada*|*les canadiens* lit. 'the Canadians', *of course*|*évidemment* lit. 'obviously', etc.).

3.4. FILTERING TECHNIQUES

One way to increase the precision of the mapping process is to impose some linguistic constraints on the sequences such as simple noun-phrase constraints (Gaussier, 1995; Kupiec, 1993; Chen and Chen, 1994; Fung, 1995; Evans and Zhai, 1996). It is also possible to focus on non-compositional compounds, a crucial key-point in bilingual applications (Su et al., 1994; Melamed, 1997; Lin, 1999). Another interesting approach is to restrict sequences to those that do not cross constituent boundary patterns (Wu, 1995; Furuse and Iida, 1996).

In this study, we investigated the impact of two filters: one that considers only units containing a noun phrase (NP), and another that focusses on source and target units that have a good chance of being the translation of one another.

3.4.1. *NP-Filtering*

We filtered out potential sequences using simple regular expressions describing sequences of part-of-speech tags that characterize NPs. An excerpt of the association probabilities of a unit model trained considering only the NP-sequences is given in Table VI. Inspection of this table reveals that most associations seem correct and that some badly ranked ones are erratic: for instance the association *the guidelines|le 1er avril*, line 18, comes from the fact that the source unit appears only six times in the training corpus but once in a sentence containing also the NP *first of April*. Post-filtering procedures could be designed to remove many of these erratic associations.

Applying this filter (henceforth $\mathcal{F}_{NP}$) to the 39,093 English sequences still alive after previous filters $\mathcal{F}_1$ and $\mathcal{F}_2$ removes 35,939 of them (92%). More than half of the 3,154 remaining NP sequences contain only two words. This is very small compared to the number of sentences of our training corpus (over 15,000), but this also reflects the fact that we did not apply any morphological transformation to our corpus, thus lowering the chance of seeing a sequence (*a fortiori* an NP one) twice.

3.4.2. *Bilingual Filtering*

We also investigated the use of the bilingual likelihood test in order to filter out badly ranked sequences. More precisely, we computed $\rho(x, y)$ for each source (x) and target (y) sequences that appeared at least once together in a pair of sentences of our training bitext; a, b, c and d of (6) respectively stand for the number of times x and y are seen together, x is in a pair without y, y is in a pair without x, neither x nor y are in a pair. n is the number of pairs of our training corpus.

For each source sequence, we kept the ten best target associations provided that their bilingual likelihood score was above a given threshold. We finally applied a last filter to remove from these candidate associations (x, y) those from which (y, x) is not ranked first by the likelihood test computed on target to source associations. This test has been proposed by Gaussier (1995).

Table VI. Examples of bilingual nominal group associations obtained after 15 iterations. The second column indicates a source NP, the third its frequency in the training corpus. The fourth column reports a maximum of its three best-ranked target associations.

	Source unit	freq	Target units ([α, p])
1	the world	201	[le monde, 0.46] [du monde, 0.33] [le monde entier, 0.19]
2	this bill	136	[ce projet de loi, 0.97] [du projet de loi, 0.03]
3	the people of canada	112	[les canadiens, 0.57] [aux canadiens, 0.24] [des canadiens, 0.13]
4	child care	86	[les garderies, 0.59] [la garde d' enfants, 0.23] [des services de garde d' enfants, 0.13]
5	interest rates	85	[les taux d' intérêt, 0.71] [des taux d' intérêt, 0.26] [le loyer de l' argent, 0.03]
6	this matter	76	[cette affaire, 0.39] [la question, 0.26] [cette question, 0.22]
7	the free trade agreement	75	[l' accord de libre-échange, 0.96] [la décision du gatt, 0.04]
8	post-secondary education	66	[l' enseignement postsecondaire, 0.75] [l' éducation postsecondaire, 0.15] [des fonds, 0.06]
9	the first time	62	[la première fois,1.00]
10	the throne speech	54	[le discours du trône, 0.80] [du discours du trône, 0.10] [ce discours du trône, 0.05]
11	the canadian aviation safety board	36	[le bureau canadien de la sécurité aérienne, 0.55] [du bureau canadien de la sécurité aérienne, 0.31] [l' un, 0.14]
12	the facts	36	[les faits, 0.79] [la réalité, 0.12] [la vérité, 0.09]
13	accident investigation	30	[les accidents, 0.31] [enquête sur les accidents, 0.31] [accidents de transport, 0.21]
14	minister for international trade	27	[ministre du commerce extérieur, 1.00]
15	the next five years	26	[au cours des cinq prochaines années, 0.53] [cinq prochaines années, 0.27] [25 milliards de d ollars, 0.10]
16	the canadian environmental protection act	18	[la loi canadienne sur la protection de l' environnement, 0.51] [de la loi canadienne sur la protection de l' environnement, 0.39] [des articles, 0.03]
17	the people of china	17	[le peuple chinois, 0.38] [la population chinoise, 0.25] [les chinois, 0.13]
18	the guidelines	6	[les lignes directrices, 0.95] [le 1er avril, 0.05]

Once again we applied a dynamic-programming scheme to find the grouping, optimizing over each pair of sentences the sum of the bilingual likelihood scores.

4. Evaluating Unit Models

Any application-oriented task should be evaluated in a real environment, and in our case, this would mean asking translators to evaluate the usefulness of TRANSTYPE. We did carry out an in-situ evaluation of TRANSTYPE, which is summarized in Section 5. In this section, however, we focus on the comparison of the many variations we attempted in training unit models. Here, we cope with the evaluation problem by an automatic process which measures the theoretical performance of TRANSTYPE over a pair of translated texts. More precisely we measure the number of keystrokes saved by a hypothetical user producing the target text as a translation of the source text. We assume a left-to-right mode in which the user is expected to type the translation sentence by sentence, going from left to right. A completion is proposed automatically by the system after each keystroke. Then the user has two choices: (a) accept the completion by typing an acceptance key, or (b) ignore the completion by typing the next character of the word under translation. In the first version of the evaluation protocol (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 a far too strong hypothesis and this scenario is only valid in the case of a translator typing very slowly. It is however directly reproducable.

4.1. A USER-ORIENTED EVALUATION SCENARIO

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. An example of such a situation is reported in Table I above, line 8, where the completion proposed is *au projet de loi sur* while the user wanted *au projet de loi que*.

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 largest 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 VII 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.

Table VII. 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 make s.

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	ière	-	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

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 (13),

$$spared = 100 \times \frac{\sum_R (|p| + cost) + acceptances + separators}{\sum_{i=0}^{n} (\sum_{j=1}^{n_t^i} (|w_j^i| + 1) - 1)} \tag{13}$$

where $|w_j^i|$ stands for the number of characters of the jth token of the ith target sentence to produce, $acceptances$ stands for the number of times a completion has been accepted (we assume that an acceptance keystroke also adds a separator) and $separators$ 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 eventual corrections made on partially good completions that have been accepted.

In the one-sentence session example given in Table I, $|p| = 14$, $cost = 1$ (one keystroke to remove the last unwanted word of the completion *au projet de loi sur*), $acceptance = 11$ and $separator = 0$ (here, TRANSTYPE always proposed a good completion before the user finished typing the token under translation). The number of characters of the target text to produce is 106 (86 plain characters plus 20 separators).

4.2. RESULTS

We collected completion results on a test corpus of 747 sentences (13,386 English tokens and 14,506 French ones) taken from the Hansard corpus. These sentences have been selected randomly from among sentences that have not been used for the training. Around 18% of the source and target words are not known by the translation model.

In this experiment, we did not use the predictions from the language model (i.e., $\lambda(\Theta(t', s)) = 0$ in Equation (1)). The results are reported in Table VIII. In

this table, the only grouping operator used was the length one $\mathcal{G}_l$ (10) described in Section 3.3.1. Other operators yielded worse performance.

The baseline models (line 1 and 2) are obtained without any unit model (i.e., $\beta = 1$ in Equation (2)). The first is obtained with an IBM-like model 1 ($a(i|j, |s|)$ = constant in Equation (3)) while the second is an IBM-like model 2. We observe that for the pair of languages we considered, IBM model 2 improves the number of saved keystrokes by almost 3% compared to IBM model 1. Therefore we made use of alignment probabilities for the other models.

The next three blocks in Table VIII show how the training methodology (see Section 3.3.2) influences the performance. Training models under the $\mathcal{E}_1$ method give the worst results. This lies in the fact that the word-to-word probabilities trained on the sequence-based corpus (predicted by M_u in Equation (2)) are less accurate than the ones learned from the token-based corpus. The reason is simply that there are less occurrences of each token, especially if many units are identified by the grouping operator. This explains for instance that model 3 (no sequence filtering) is worse than model 6 (bilingual filtering) because more groupings are performed with the former. The distribution of words and units of the sequence-based training corpus after applying several filtering strategies is reported in Table IX.

In methods $\mathcal{E}_2$ and $\mathcal{E}_3$, the unit model of equation (2) only makes predictions $p_u(t|s)$ when s is a source unit, thus lowering the noise compared to method 1.

We also observe in these three blocks the influence of the filtering done on sequences: the more we filter, the better the results, independent of the training method used. In the fifth block of Table VIII we observe the positive influence of the NP-filtering, especially when using the third estimation method.

The best combination we found is reported in line 20. It outperforms the baseline by around 1.5%. This model has been obtained by retaining (monolingually) all sequences seen at least two times in the training corpus for which the likelihood test value was above 5 and the entropy score above 0.2 ($\mathcal{F}_1(2, 2, 5, 0.2)$).

In terms of the coverage of this unit model, it is interesting to note that among the 747 sentences of the test session, there were 228 for which the model did not propose any unit at all. For 425 of the remaining sentences, the model proposed at least one helpful (good or partially good) unit. Actually the active vocabulary for these sentences contained an average of around 2.5 good units (1031/425) per sentence, among which only half (495) have been proposed during the session.

5. An In-Situ Evaluation of TRANSTYPE

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

Table VIII. Completion results of several translation models. "Spared" is the theoretical proportion of characters saved; $|p|$, the number of prefixes typed; "Cost" is the amount of keystrokes needed to correct partially good units; "OK" is the number of target units accepted by the user; "Good" is the number of target units that matched the expected whether they were proposed or not; nu is the number of sentences for which no target unit was found by the translation model; and u is the number of sentences for which at least one helpful unit has been found by the model, but not necessarily proposed. $\mathcal{F}_1(x)$ is a shorthand for $\mathcal{F}_1(2, 2, x, 0.2)$.

	Model	Spared	$\|p\|$	Cost	OK	Good	nu	u
1	Baseline – model 1	48.98	23522	2120	0	0	747	0
2	Baseline – model 2	51.83	21455	1988	0	0	747	0
3	$\mathcal{E}_1 + \mathcal{F}_1(0)$	50.98	21769	3342	527	1702	5	626
4	$\mathcal{E}_1 + \mathcal{F}_1(5)$	51.61	21386	3314	596	2149	5	658
5	$\mathcal{E}_1 + \mathcal{F}_1(5) + \mathcal{F}_2$	51.72	21387	3188	633	2265	5	657
6	$\mathcal{E}_1 + \mathcal{F}_1(5) + \mathcal{F}_2 + \mathcal{F}_3(10)$	51.86	21554	2148	134	522	5	353
7	$\mathcal{E}_2 + \mathcal{F}_1(0)$	51.39	21648	3126	514	1551	43	578
8	$\mathcal{E}_2 + \mathcal{F}_1(5)$	51.99	21373	2863	470	1889	46	614
9	$\mathcal{E}_2 + \mathcal{F}_1(5) + \mathcal{F}_2$	52.12	21384	2742	493	1951	46	606
10	$\mathcal{E}_2 + \mathcal{F}_1(5) + \mathcal{F}_2 + \mathcal{F}_3(10)$	52.23	21342	2137	155	525	168	355
11	$\mathcal{E}_3 + \mathcal{F}_1(0)$	51.07	21638	3463	577	1699	43	588
12	$\mathcal{E}_3 + \mathcal{F}_1(5)$	51.47	21321	3527	629	2124	46	618
13	$\mathcal{E}_3 + \mathcal{F}_1(5) + \mathcal{F}_2$	51.68	21308	3405	665	2209	46	615
14	$\mathcal{E}_3 + \mathcal{F}_1(5) + \mathcal{F}_2 + \mathcal{F}_3(10)$	52.74	21098	2396	374	527	176	355
15	$\mathcal{E}_1 + \mathcal{F}_1(0) + \mathcal{F}_{NP}$	52.83	21094	2178	416	1302	4	564
16	$\mathcal{E}_1 + \mathcal{F}_1(5) + \mathcal{F}_2 + \mathcal{F}_{NP}$	52.83	21094	2178	416	1302	4	564
17	$\mathcal{E}_1 + \mathcal{F}_1(5) + \mathcal{F}_{NP}$	52.84	21021	2240	411	1309	4	563
18	$\mathcal{E}_3 + \mathcal{F}_1(5) + \mathcal{F}_{NP}$	53.12	20949	2163	439	1031	228	425
19	$\mathcal{E}_3 + \mathcal{F}_1(5) + \mathcal{F}_2 + \mathcal{F}_{NP}$	53.16	20931	2172	458	1052	199	439
20	$\mathcal{E}_3 + \beta = 0.4 + \mathcal{F}_1(5) + \mathcal{F}_{NP}$	53.22	20871	2241	495	1031	228	425

Table IX. Distribution of words and units in the sequence-based training corpus according to several filtering strategies. w (u) is the number of tokens (units) in the corpus, w_f (u_f) is the number of different words (units) found, and w_{f2} (u_{f2}) is the number of different words (units) which occur at leats two times.

	Model	w	w_f	w_{f2}	u	u_f	u_{f2}
1	$\mathcal{F}_1(0)$	35745	10600	4853	84335	36508	18952
2	$\mathcal{F}_1(5)$	64379	10753	5197	73410	23000	14056
3	$\mathcal{F}_1(5) + \mathcal{F}_2$	71133	10914	5361	71464	19295	13778
4	$\mathcal{F}_1(10) + \mathcal{F}_2$	95671	11186	5971	65201	10727	8980
5	$\mathcal{F}_1(100) + \mathcal{F}_2$	181844	11807	7031	35424	522	515
6	$\mathcal{F}_1(5) + \mathcal{F}_2 + \mathcal{F}_{NP}$	198064	11568	6608	27094	4138	4038
7	$\mathcal{F}_1(5) + \mathcal{F}_2 + \mathcal{F}_3(10)$	203266	11118	6402	6997	2406	1265

translation process? Is speed and/or keystroke savings 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, by asking translators to use it. In this section, we summarize this work which was carried out at the same time as we investigated the training procedure previously described. Hence, we were only able to evaluate TRANSTYPE in its word-completion mode (that is, β was set to 1 in Equation (2)).

5.1. THE PROTOCOL

We designed an evaluation protocol that encompasses three major steps. In the first one, TRANSTYPE works in a silent mode (i.e., it does not propose anything) and the user only uses the editing functions implemented in the prototype.[4] In the second step, TRANSTYPE is switched to its normal mode, that is, proposing after each keystroke the completion of the current word. The third and last step of the protocol intended to measure how a user may perceive TRANSTYPE if it were able to predict the next several words instead of the only current one. In this stage, we manually introduced sequences of words (called "briskels") that a translator would be likely to want to use in a translation. These briskels are provided in a special area of the interface, once the user selects the source sentence to translate. A briskel may be inserted into the translation simply by clicking on it. Roughly, a briskel corresponds in quality to the automatic associations we found with our unit models. The evaluation protocol ends with a 10-minute feed-back survey to collect the subject's feelings and suggestions.

5.2. THE USERS

Our volunteer users were identified via word of mouth. All of the users we found had a special interest in testing MT or IMT prototypes. Four were either professional translators or professors from the University of Montreal actively involved in teaching translation. The other six were graduate students engaged in a translation program. All of them were very familiar with computers. The testing was carried out over a period of three weeks at RALI.

5.3. THE MATERIAL

We put together a corpus of about 100 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 or contained too many complicated proper names or numbers, etc. Finally, we inspected the sentences selected in order to remove those that we found ambiguous or difficult to translate without reference to larger context (e.g., sentences with ellipses, etc.). Our users found the corpus relatively easy to translate and representative of a realistic translation task.

5.4. THE QUALITATIVE SURVEY

A set of open questions were asked in order to get qualitative feedback from the users. This survey and its analysis are detailed in Langlais et al. (2000). There were, however, a few interesting results which we would like to present here. First of all, except for one user who said clearly that she hated TRANSTYPE and would never use such a tool in her work, the nine others expressed in various ways that they liked it and would enjoy using it in their daily work. They did however mention some points that would improve 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. Other suggestions were of more interest to us. For example, all the users (even the one who disliked TRANSTYPE) agreed that they did like the stage where they were provided with 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.

We also asked the users if they found that the proposals made by TRANSTYPE after each keystroke were disturbing. 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 have to reformulate part of a sentence, in which case they would prefer a dumb prototype.[5] These six users also mentioned that it is difficult simply to 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

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

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

a term). Furthermore, they also mentioned that being disturbed by TRANSTYPE is not necessarily a drawback: According to some users, it often happens 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.

The last point we would like to mention is that most of the users felt TRANSTYPE was helping them type faster. Interestingly, except for one user, none of the users actually managed to type faster using the system's completions.

5.5. A QUANTITATIVE ANALYSIS DATA

All interactions between the user and TRANSTYPE were recorded in a log file during the evaluation test. In order to assess precisely how TRANSTYPE influences the work of the subjects, we computed three measurements: productivity, effort and efficiency. "Productivity" is computed as the typing speed of a subject, that is, the ratio of the characters produced in the translation over the time spent to type those characters. "Effort" is the ratio of any action (keystrokes or mouse click) produced over the time spent to translate. Finally, the "efficiency" of a user is the ratio of productivity over effort.

Table X reports the average values of these three ratios (all subjects taken together) measured for each stage. The results of the complete analysis we have carried out (Langlais et al., 2000) may be summarized in few points.

First of all, and contrary to the impression they had, the users were less productive using TRANSTYPE. Looking at the productivity rates of each subject in detail, it turns out that except for one subject who managed to outperform stage 1 using the completions, all the other subjects were less productive. The decline in productivity is either moderate or drastic, the latter being the case of the subject who disliked TRANSTYPE; her typing speed was reduced by half. However, a careful analysis of the logfiles leads us believe that this disappointing result is partly a matter of undertraining. Finding a good strategy for using TRANSTYPE is not as easy as we first thought.

Second, it is interesting to note that the typing speed measured in the third stage (that is, where we simulated a unit completion scenario) is encouraging: three subjects outperformed stage 1 in terms of productivity, and only a few were significantly slower. This confirms the need for a better translation model.

Lastly, Table X shows 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 us say, 100 characters, a user requires on average 143 actions (keystrokes or mouse clicks) in stage 1. In stage 2, only about 77 actions are required to produce the same translation. Finally, in stage 3, the user requires only 53 actions. This at least indicates that users do use some of the completions that TRANSTYPE offers! However, the acceptance rate of a completion is far from the one we measure in the theoretical evaluation and ranges from 15% to 43%. In general, a user tends to accept a completion if it is proposed before they type the first or the second letter of the intended word, and only for longer completions (at least 4 or 5 characters).

6. Related Work and Discussion

Considerable work has been devoted to the automatic identification of bilingual collocations in corpora. The widely adopted approach we followed is first to calculate monolingual n-gram statistics and then apply a mechanism to map the collocations of the two languages. Many metrics and filters have been investigated and proposed for this purpose, depending on the nature of the bilingual corpus considered – parallel aligned, parallel nonaligned (Kaji and Aizono, 1996; Ohomori and Higashida, 1999), non-parallel (Rapp, 1999; Tanaka and Matsuo, 1999) – and on the resources available (tokenizers, taggers, chunkers, etc.). The evaluation of such algorithms is difficult and usually done by a manual inspection of a (small) sample of the bilingual associations found and the calculation of the classical precision and recall rates. Such an evaluation procedure is not only laborious but also difficult and subjective. In addition, it does not allow investigation of the impact of different metrics or filters on the global performance rate.

These studies have not necessarily been carried out with the goal of improving MT, but very often to get a new resource that translators may use. Our motivation is to enhance TRANSTYPE, so we devised a translation model which accounts for both words and units. This model, although simple, captures some linguistic constraints that improve the completion task. However, as we mentioned before, we consider this model as a baseline for further investigations on translation modeling. One obvious direction for future research is to revise our current strategy of decoupling the selection of units from their bilingual context.

Recent proposals have been made for overcoming to some extent the independence assumptions of statistical models such as IBM models 1 and 2. These studies report improvements on some specific tasks (task-oriented limited vocabulary) which by nature are very different from the task TRANSTYPE is devoted to. First

of all, Brown et al. (1993) proposed IBM models 3, 4 and 5 which account for $1 - n$ bilingual notions automatically discovered during the training procedure; IBM model 2 is in fact just an intermediate model that serves to initialize the parameters of the more elaborate models.

Berger et al. (1996) investigated a maximum entropy approach to derive as many models $P_e(y|x)$ as source words e, where $P_e(y|x)$ represents the probability that y is a good translation of e, given its surrounding source context x. They illustrated the use of context-sensitive models on some specific tasks and showed how attractive the approach is to reconcile both a classical linguistic introspection within a principled framework.

Many works have also been carried out to improve translation models within the Verbmobil framework. Wang and Waibel (1998b) propose a new alignment model based on shallow phrase structures automatically acquired from a parallel corpus using a clustering and mutual-information based phrasing procedure. The idea is to align first the structures with a rough alignment, then to align words within the aligned structures, thus introducing some constraints relevant from a linguistic point of view.

Och and colleagues (Och and Weber, 1998; Och et al., 1999) also presented an algorithm in the same spirit, with a different way of integrating the phrasing, also deriving benefits from a bilingual clustering procedure.

Wu and Wong (1998) introduced a stochastic grammatical channel model for MT that has some appealing characteristics, among which is the fact that the translation produced conforms to a target grammar (here a bigram language model).

Most of these studies are still evolving and usually require complex decoding strategies that we cannot afford because of real-time considerations. It is however worth noticing that recent work on dynamic-programming decoders (Tillman et al., 1997; Nießen et al., 1998) and stack-based (Wang and Waibel, 1997, 1998a) make us foresee some possible integrations within TRANSTYPE.

7. Conclusions

We have presented a prototype which implements an innovative and appealing IMT scenario where the interaction is mediated via the target text under production. Among other advantages, this approach relieves the translator of the burden of source analyses, and gives them direct control over the final translation without having to resort to postediting. The theoretical model behind TRANSTYPE has been described and alternative approaches that we are currently investigating have been discussed. We proposed a mechanism to enhance the power of TRANSTYPE, which is now able to predict sequences of words. We observed an improvement in overall prediction performance that will serve as a baseline for our future investigations in translation modeling. Finally, we have presented a first in-situ evaluation of

TRANSTYPE, the results of which are encouraging for the pursuit of a targeted approach to IMT.

Acknowledgements

TRANSTYPE is a project funded by the Natural Sciences and Engineering Research Council of Canada. We are greatly indebted to Elliott Macklovitch and Pierre Isabelle for the fruitful orientation they gave to this work. We also would like to thank the anonymous reviewers of this contribution for their useful guidance.

Notes

[1] We assume the existence of a deterministic procedure for tokenizing the target text.

[2] See Section 4 for details of the evaluation protocol.

[3] In both cases the final word is missing from the French: *au fils des années*, *nous avons*.

[4] We implemented an editor which offers all the standard operations (cut and paste, delete, etc.).

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

References

Berger, Adam L., Stephen A. Della Pietra, and Vincent J. Della Pietra: 1996,'A Maximum Entropy Approach to Natural Language Processing', *Computational Linguistics* **22**, 39–71.

Blanchon, Hervé: 1991, 'Problèmes de désambiguïsation interactive et TAO personnelle' [Problems with Interactive Disambiguation and Personal MT], *L'environnement traductionnel, journées scientifiques du réseau thématique de recherche "Lexicologie, terminologie, traduction"*, Mons, pp. 31–48.

Brown, Peter F., Stephen A. Della Pietra, Vincent J. Della Pietra and Robert L. Mercer: 1993, 'The Mathematics of Machine Translation: Parameter Estimation', *Computational Linguistics* **19**, 263–312.

Brown, Ralf D. and Sergei Nirenburg: 1990, 'Human-Computer Interaction for Semantic Disambiguation', *COLING 90: Papers presented to the 13th International Conference on Computational Linguistics*, Helsinki, Vol. 3, pp. 42–47.

Chen, Kuang Hua and Hsin-Hsi Chen: 1994, 'Extracting Noun Phrases from Large-Scale Texts: A Hybrid Approach and its Automatic Evaluation', *32nd Annual Meeting of the Association for Computational Linguistics*, Las Cruces, New Mexico, pp. 234–241.

Derouault, A.-M. and B. Merialdo: 1986, 'Natural Language Modeling for Phoneme-to-Text Transcription', *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* **8**, 742–749.

Dunning, Ted: 1993, 'Accurate Methods for the Statistics of Surprise and Coincidence', *Computational Linguistics* **19**, 61–74.

Evans, David A. and Chengxiang Zhai: 1996, 'Noun-Phrase Analysis in Unrestricted Text for Information Retrieval', *34th Annual Meeting of the Association for Computational Linguistics*, Santa Cruz, CA, pp. 17–24.

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

Fung, Pascale: 1995, 'A Pattern Matching Method for Finding Noun and Proper Noun Translations from Noisy Parallel Corpora', *33rd Annual Meeting of the Association for Computational Linguistics*, Cambridge, MA, pp. 236–243.

Furuse, Osamu and Hitoshi Iida: 1996, 'Incremental Translation Utilizing Constituent Boundary Patterns', *COLING-96: The 16th International Conference on Computational Linguistics*, Copenhagen, pp. 412–417.

Gaussier, Éric: 1995, *Modèles statistiques et patrons morphosyntaxiques pour l'extraction de lexiques bilingues* [Statistical Models and Morphosyntactic Templates for the Extraction of Bilingual Lexicons], Thèse de doctorat, Université de Paris 7.

Haruno, Masahiko, Satoru Ikehara and Takefumi Yamazaki: 1996, 'Learning Bilingual Collocations by Word-Level Sorting', *COLING-96: The 16th International Conference on Computational Linguistics*, Copenhagen, pp. 525–530.

Ikehara, Satorn, Satoshi Shirai and Hajime Uchino: 1996, 'A Statistical Method for Extracting Uninterrupted and Interrupted Collocations from Very Large Corpora', *COLING-96: The 16th International Conference on Computational Linguistics*, Copenhagen, pp. 574–579.

Jelinek, Frederick: 1990, 'Self-Organized Language Modeling for Speech Recognition', in A. Waibel and K. Lee (eds), *Readings in Speech Recognition*, Morgan Kaufmann, San Mateo, CA, pp. 450–506.

Kaji, Hiroyuki and Toshiko Aizono: 1996, 'Extracting Word Correspondances from Bilingual Corpora Based on Word Co-Occurence Information', *COLING-96: The 16th International Conference on Computational Linguistics*, Copenhagen, pp. 23–28.

Kay, Martin: 1973, The MIND System', in R. Rustin (ed.) *Natural Language Processing*, Algorithmics Press, New York, pp. 155–188.

Kuhn, Roland and Renato De Mon: 1990, 'A Cache-Based Natural Language Model for Speech Recognition', *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* **12**, 570–583.

Kupiec, Julian: 1993, 'An Algorithm for Finding Noun Phrase Correspondences in Bilingual Corpora', *31st Annual Meeting of the Association for Computational Linguistics*, Columbus, OH, pp. 17–22.

Langlais, Philippe and George Foster: 2000, 'Using Context-Dependent Interpolation to Combine Statistical Language and Translation Models for Interactive Machine Translation', *Proceedings of the 6th International Conference on Computer-Assisted Information Retrieval (RIAO)*, Paris, pp. 507–519.

Langlais, Philippe, Sébastien Sauvé, George Foster, Elliott Macklovitch, and Guy Lapalme: 2000, 'Evaluation of TransType, a Computer-aided Translation Typing System: A Comparison of a Theoretical- and a User-Oriented Evaluation Procedures', *LREC 2000 Second International Conference on Language Resources and Evaluation*, Athens, pp. 641–648.

Lin, Dekang: 1999, 'Automatic Identification of Non-Compositional Phrases', *37th Annual Meeting of the Association for Computational Linguistics*, College Park, MD, pp. 317–324.

Maruyama, Hiroshi, Hideo Watanabe and Shiho Ogino: 1990, 'An Interactive Japanese Parser for Machine Translation', *COLING-90: Papers presented to the 13th International Conference on Computational Linguistics*, Helsinki, Vol. 2, pp. 257–262.

Melamed, I. Dan: 1997, 'Automatic Discovery of Non-Compositional Compounds in Parallel Data', *Second Conference on Empirical Methods in Natural Language Processing*, Providence, RI, pp. 97–108.

Nagao, Makoto and Shinsuke Mon: 1994, 'A New Method of n-Gram Statistics for Large Number of n and Automatic Extraction of Words and Phrases from Large Text Data of Japanese', *COLING 94: The 15th International Conference on Computational Linguistics*, Kyoto, pp. 611–615.

Nießen, S., S. Vogel, H. Ney and C. Tillman: 1998, 'A DP Based Search Algorithm for Statistical Machine Translation', *COLING-ACL '98: 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, Montreal, pp. 960–966.

Och, Franz Josef, Christoph Tillman and Herman Ney: 1999, 'Improved Alignment Models for Statistical Machine Translation', *Proceedings of the 1999 Joint SIGDAT Conference on Empir-*

ical Methods in Natural Language Processing and Very Large Corpora, College Park, MD, pp. 20–28.

Och, Franz Josef and Hans Weber: 1998, 'Improving Statistical Natural Language Translation with Categories and Rules', *COLING-ACL '98: 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, Montreal, pp. 985–989.

Ohomori, Kumiko and Masanobu Higashida: 1999, 'Extracting Bilingual Collocations from Non-Aligned Parallel Corpora', *Proceedings of the 8th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI 99)*, Chester, pp. 88–97.

Rapp, Reinhard: 1999, 'Automatic Identification of Word Translations from Unrelated English and German Corpora', *37th Annual Meeting of the Association for Computational Linguistics*, College Park, MD, pp. 519–526.

Russell, Graham: 1998, *Identification of Salient Token Sequences*, Internal report, RALI, University of Montreal.

Shimohata, Sayori, Toshiyuki Sugio and Junji Nagata: 1997, 'Retrieving Collocations by Co-occurrences and Word Order Constraints', *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, Madrid, pp. 476–481.

Su, Keb-Yih, Ming-Wen Wu and Jing-Shin Chang: 1994, 'A Corpus-Based Approach to Automatic Compound Extraction', *32nd Annual Meeting of the Association for Computational Linguistics*, Las Cruces, New Mexico, pp. 242–247.

Tanaka, Kumiko and Hideya Iwasaki: 1996, 'Extraction of Lexical Translations from Non-Aligned Corpora', *COLING-96: The 16th International Conference on Computational Linguistics*, Copenhagen, pp. 580–585.

Tanaka, Takaaki and Yoshihiro Matsuo: 1999, 'Extraction of Translation Equivalents from Non-Parallel Corpora', *Proceedings of the 8th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI 99)*, Chester, pp. 109–119.

Tillman, C., S. Vogel, H. Ney and A. Zubiaga: 1997, 'A DP Based Search Using Monotone Alignments in Statistical Translation', *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, Madrid, pp. 289–296.

Wang, Ye-Yi and Alex Waibel: 1997, 'Decoding Algorithm in Statistical Machine Translation', *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, Madrid, pp. 366–372.

Wang, Ye-Yi and Alex Waibel: 1998a, 'Fast Decoding for Statistical Machine Translation', *5th International Conference on Spoken Language Processing, ICSLP'98 Proceedings*, Sydney, pp. 2775–2779.

Wang, Ye-Yi and Alex Waibel: 1998, 'Modeling with Structures in Statistical Machine Translation', *COLING-ACL '98: 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, Montreal, pp. 1357–1363.

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', *11th International Conference on Computational Linguistics: Proceedings of Coling '86*, Bonn, pp. 329–334.

Wu, Dekai: 1995, 'Stochastic Inversion Transduction Grammars, with Application to Segmentation, Bracketing, and Alignment of Parallel Corpora', *Fourteenth International Joint Conference on Artificial Intelligence*, Montreal, pp. 1328–1335.

Wu, Dekai and Hongsing Wong: 1998, 'Machine Translation with a Stochastic Grammatical Channel', *COLING-ACL '98: 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, Montreal, pp. 1408–1414.