Target-Text Mediated Interactive Machine Translation

GEORGE FOSTER, PIERRE ISABELLE and PIERRE PLAMONDON
Centre for Information Technology Innovation (CITI)
1575 Chomedey Blvd.
Laval, Quebec, Canada, H7V 2X2

Abstract. The use of machine translation as a tool for professional or other highly skilled translators is for the most part currently limited to postediting arrangements in which the translator invokes MT when desired and then manually cleans up the results. A theoretically promising but hitherto largely unsuccessful alternative to postedition for this application is *interactive machine translation* (IMT), in which the translator and MT system work in tandem. We argue that past failures to make IMT viable as a tool for skilled translators have been the result of an infelicitous mode of interaction rather than any inherent flaw in the idea. As a solution, we propose a new style of IMT in which the target text under construction serves as the medium of communication between an MT system and its user. We describe the design, implementation, and performance of an automatic word completion system for translators which is intended to demonstrate the feasibility of the proposed approach, albeit in a very rudimentary form.

Key words: Machine-Assisted Human Translation, Interactive Machine Translation, Target-Text Mediation, Word Completion, Statistical Translation Models, Statistical Language Models

1. Introduction

Machine translation systems have generally not, in the past, fared well as assistants to professional or other highly skilled human translators in situations where high-quality results are required. They have been used in essentially all possible temporal configurations: *postedition*, *preedition*, and *interactive MT*—that is, in which the machine's contribution occurs respectively before, after, and in tandem with the human's. Postedition is the simplest method, but it is viable only in the relatively rare cases where MT output is good enough to make the effort required for its revision substantially less than that of producing a new translation from scratch. Preedition is based on the premise that a certain amount of labour invested in preparing a source text for machine analysis will be more than repaid by the resulting improvement in MT performance. This is a promising idea, but in practice annotating text so as to make it easier for a machine to analyze has proven an onerous task for human translators. Interactive MT inherits the theoretical potential of preedition, with the added advantage that the information required from the translator can be made to depend on the machine's current state and thus in principle reduced to a minimum. It has foundered in the past for essentially the same reason as preedition: producing explicit linguistic analyses appears to be a task that is not

significantly easier—and certainly not more appealing—for a competent translator than translation itself.

Designers of current translator's support environments, eg (Eurolang, 1995; Frederking et al., 1993; IBM, 1995; Kugler et al., 1991; Nirenburg, 1992; Picchi et al., 1992; Trados, 1995), exhibit a healthy respect for the lessons of the past. MT components are optional on most of these systems, and the mode of interaction is invariably postedition. The translator is never encumbered with unwanted MT output because he or she has control over when such output will be displayed and the freedom to ignore it when it is not useful. This is a very sensible approach, but given the rather striking disparity in degree of sophistication between an MT system and the other specialized tools typically available to a translator—dictionaries, term banks, concordances, and the like—one cannot help but wonder whether there is really no better way of exploiting the capabilities of MT in this context. Specifically, might it be possible to revive the old and theoretically promising idea of interactive MT for translators? This is the question we propose to investigate in this paper.

2. Interactive Machine Translation

2.1. Current Approaches

The first IMT facility was implemented as part of Kay's MIND system (Kay, 1973), where the user's role was to help with source text disambiguation by answering questions about word sense, pronominal reference, prepositional phrase attachment, etc. Later systems, eg (Blanchon, 1994; Boitet, 1990; Brown, 1990; Maruyama, 1990; Melby, 1987; Tomita, 1985; Whitelock et al., 1986; Zajac, 1988), have essentially all been cast in the same mold. Research has concentrated primarily on making the disambiguation process more efficient and less burdensome for the user via techniques like ordering questions so as to minimize the expected number that will need to be asked; finding more natural formulations in terms of alternate paraphrases of the source text; presenting multiple choice responses with the most likely answer as a default choice; and tailoring the interaction to suit the user's degree of familiarity with the system. Despite progress in these endeavors however, the question-and-answer process remains a laborious one and current IMT is therefore most appropriate in applications where the cost of manually producing a translation is high enough to justify the extra effort involved, for example when the user's knowledge of the target language may be limited or non-existent, or when there are multiple target languages.

From the viewpoint of an accomplished translator, the problems with the conventional approach to IMT can be summarized as follows. First, the interaction between person and machine is for the most part concerned with something that does not normally need to be made explicit in human translation, namely the precise meaning of the source text—and it largely ignores that which invariably does,

namely the target text. Second, the language in which the interaction takes place is an awkward one for a human, because it is ultimately based upon the MT system's model of the source text. Some of the awkwardness can doubtless be avoided by reformulating the questions in natural language and having them incorporate relevant portions of the source text, but this appears to be a difficult enterprise and it is not clear how far it can be taken.

2.2. TARGET-TEXT MEDIATION

In our opinion, these problems would be greatly alleviated if the focus of interaction were shifted from the *meaning* of the source text to the *form* of the target text. This would relieve the translator of the burden of having to provide explicit source analyses, and give him or her direct control over the final translation without having to resort to postedition. It would also make possible a very simple and natural style of interaction consisting of manipulations of the actual words and characters in the target text. In such a system, a translation would emerge from a series of 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 might be called *target-text mediated* (TTM) IMT, encompasses a number of interesting possibilities. The basic unit of interaction is likely to be the character, which the translator has a very efficient device for producing, and whose manipulation would necessitate a minimum of special commands beyond those to be found in a word processor. This does not preclude the concurrent use of other methods for specifying text, such as pop-up menus containing lists of alternate words, commands to select particular morphological variants of a word, etc. Indeed, should it prove advantageous and feasible, there is no reason why the direct specification of text by the translator could not be augmented with specifications of certain of its local or global properties. For instance, it might be possible to indicate that a particular lexical item is to be avoided; or that a sentence is to be rendered in the passive voice; or even that the translation should tend toward conciseness or prolixity.

TTM can in principle accomodate a wide range of MT proficiencies. Simple systems would be of benefit mainly in speeding the transcription of the translator's work; more capable ones would add to this the ability to occasionally suggest solutions that may otherwise have eluded (at least temporarily) the human partner. It is also conceivable that other tools could be usefully integrated into a TTM framework. For example, a translation memory that can locate approximate matches to the current source text segment might be used instead of the MT system to produce a first draft when appropriate. An automatic dictation system such as TransTalk (Brousseau et al., 1995) could also fill this role, with the translator using TTM to simultaneously correct dictation errors and revise an initial rough translation.

A bilingual dictionary could furnish alternatives for a given target-text word by relying on context to determine the source-text word for which translations were sought.

2.3. IMPLEMENTING TTM

Much of the foregoing is of course highly speculative. The conventional approach to IMT has been adopted because it is technically feasible, and it remains to be seen whether this is the case for the approach we are advocating here. Furthermore, any hope of making TTM useful in the short term (not to mention the stated rationale for this paper) turns on being able to rely to at least some extent on existing MT techniques.

At first glance, implementing a TTM system seems a formidable challenge. In the worst-case scenario, the system will need to generate a new translation of the source text for each *character* a translator types, fast enough so as never to force even the swiftest typists to wait—this implies a rate of about two complete translations per second. The task here is easier than producing new translations from scratch, since the source text remains the same between user interventions, and since the information already given about the target text can in principle be used to facilitate the generation of an updated version. Nonetheless, even for source texts no longer than an average sentence it seems likely that this level of performance will be difficult to achieve, so it is useful at present to consider machine contributions of more limited scope. These could take several forms, including, for example, modifying word inflections in order to preserve the grammaticality of a sentence after a change introduced by the translator, or replacing only a few words around the point of an intervention. A very basic operation is the determination of a single word that has been partially specified in some way by the translator, for instance by giving a few of its characters. This shall be our focus for the remainder of the paper.

The task is to find an appropriate word for a particular position in the target text, given the corresponding source text and a set of constraints (which we leave unspecified for the moment) on the target text, some of which may apply to the word in question. Needless to say, a *sine qua non* for this operation is that it take substantially less time than producing a complete new target text. For classical rule-based MT, this does not appear to be a trivial problem. It would be difficult to get a rule-based system to efficiently generate a single word without also producing the whole target text in which it occurs. It would probably also be hard to build in a capacity to generate translations compatible with arbitrary sets of target-text constraints. For statistical MT, on the other hand, the solution seems easier. A single word can be chosen efficiently by searching the vocabulary for the entry with the highest probability conditional on the source text and what is known about the target text. Arbitrary target-text constraints can be handled by simply ignoring those

for which the model used to estimate probabilities makes no provision. A statistical approach therefore seems the natural choice for this form of TTM.

That a statistical model can be used to evaluate word hypotheses does not of course guarantee that evaluation will be fast or accurate enough for the purpose. In the next section we demonstrate that both these characteristics are attainable in a rudimentary TTM system based on simple statistical MT techniques. The system is intended for English to French translation, and its function is to automatically complete each word in a French target text from some (possibly nil) prefix typed by the translator. Although it seems reasonable to expect that some form of word completion is likely to be useful for translators, we should stress that we have not yet attempted to verify this conjecture, and it is not the aim of this paper to assert it. Rather we hope to open the door to future research by showing that at least one plausible form of TTM is well within the reach of current MT technology.

3. Word Completion

Our word completion system works as follows: a translator selects some portion of the source text, nominally a sentence, and begins typing its translation. After each character is entered, the system displays a proposed completion for the current word, which the translator may either accept using a special command or reject by continuing to type. We chose this interface for our initial prototype because it is simple and because it allows performance to be measured easily by counting the proportion of characters or keystrokes saved in a test corpus; these are statistics that seem likely to correlate well with actual savings in human effort.

The core of the system is a completion engine which comprises two main parts: an *evaluator* which assigns probabilistic scores to word hypotheses in context; and a *generator* which uses the evaluation function to select the best word. These are described in the following two sections, after which some test results are given.

3.1. Evaluating Hypotheses

The evaluator is a function $p(t|\mathbf{t}', \mathbf{s})$ which assigns to each target-text token t an estimate of its probability given a source text $\mathbf{s}$ and the tokens $\mathbf{t}'$ which precede t in the current translation of $\mathbf{s}$.[⋆] 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, ie:

$$p(t|\mathbf{t}', \mathbf{s}) \propto p(\mathbf{s}|\mathbf{t}', t)p(t|\mathbf{t}')$$

[⋆] We assume the existence of a deterministic procedure for tokenizing the target text. Tokenization errors can severely affect performance in this application, but they are infrequent because the translator's previous completion commands can be used to help establish correct word boundaries.

[⋆⋆] So called because the aim is seen as the reconstruction of a target text which has been garbled during passage through a noisy channel into an observed source text.

(where proportionality holds when $\mathbf{t}'$ and $\mathbf{s}$ are fixed), we use a simple linear combination of separate predictions from a language model $p(t|\mathbf{t}')$ and a translation model $p(t|\mathbf{s})$:

$$p(t|\mathbf{t}', \mathbf{s}) = p(t|\mathbf{t}')\ \alpha(\mathbf{t}', \mathbf{s}) + p(t|\mathbf{s})\ [1 - \alpha(\mathbf{t}', \mathbf{s})] \tag{1}$$

where $\alpha(\mathbf{t}', \mathbf{s}) \in [0, 1]$ are context-dependent interpolation coefficients. Although the noisy channel technique is powerful, it has the disadvantage that $p(\mathbf{s}|\mathbf{t}', t)$ is more expensive to compute than $p(t|\mathbf{s})$ when using IBM-style translation models: $O(T'S)$ operations versus $O(S)$ operations, where T' is the number of tokens in $\mathbf{t}'$ and S is the number of tokens in $\mathbf{s}$. Since speed is crucial for our application, we chose to forego the noisy-channel approach in the work described here.

The problem posed by an interpolated model such as (1) is to find features of $\mathbf{t}', \mathbf{s}$ that indicate which component—language or translation—will be a better predictor of t in a particular context. Because no simple features of the source text seem likely to be highly informative, we considered only features of the target-text context $\mathbf{t}'$. Using the models described in the next sections, we collected performance statistics for several of these, including the previous bigram, the training-set frequency of the previous bigram, the current part-of-speech, and the sentence position. Although some patterns emerged—for example, the translation model tends to perform relatively better at the beginnings of sentences—none was sufficiently striking to encourage further efforts.

The problem has to do with the relative strengths of the two components: the language model captures local grammatical constraints and collocations, but displays a high degree of lexical uncertainty in many contexts; the translation model has a good idea of which words should appear in the target text, but a poor notion of exactly where they should go. Thus lowering the weight on the language model in frequent contexts where it is uncertain, for example *de l'*, can weaken grammatical constraints to the point of allowing sequences like *de l' le* on the basis of a strong recommendation from the translation model. Conversely, raising the weight on the language model in frequent contexts where it is very confident, such as *projet de*, means that frequent collocations like *project de loi*[⋆] tend to be favoured even when there is overwhelming evidence against them from the source text.

This difficulty might be avoided by somehow making the interpolation coefficients depend on source as well as target context. However, a better solution would probably be to abandon the interpolated model altogether, and integrate language and translation components more closely in order to impose grammatical constraints on the translation model and lexical constraints on the language model. This could be achieved through a word-class mechanism like the following:

$$p(t|\mathbf{t}', \mathbf{s}) = \sum_c p(t|c, \mathbf{t}', \mathbf{s}) p(c|\mathbf{t}', \mathbf{s})$$

⋆ This particular trigram was one of the most frequent in the somewhat idiosyncratic Hansard corpus used to train the language model.

$$\approx \sum_c p(t|c, \mathbf{s})p(c|\mathbf{t}'),$$

where the idea is to let the language model predict which class c is most likely for the next word, and to have the translation model fill in the word subject to the class constraint. If the set of classes is chosen appropriately (perhaps even made to depend on $\mathbf{t}'$), this technique has the potential to combine the strengths of the two models in a complimentary way. Although we feel that this approach is likely to be fruitful, we have not yet investigated it in depth. The method we used to combine the models described below was interpolation with a single coefficient α, independent of context, whose value was established so as to maximize performance on a test corpus (see section 3.3.3).

3.1.1. *Language Model*

We experimented with four different French language models: a cache model, similar to the cache component in Kuhn's model (Kuhn, 1990), in which the probability of a word is estimated from its relative frequency among some fixed number of previous tokens; a unigram model in which it is estimated from relative frequency in a static training corpus; a triclass model (Derouault, 1986), in which the probability of a token depends on its morphosyntactic class, and the probability of a class on the classes of the previous two tokens; and an interpolated trigram (Jelinek, 1990), in which the probability of a token depends directly on the previous two. Various simple linear combinations of these models were tested, as described in section 3.3.

The triclass model was based on a French dictionary containing approximately 60,000 entries (380,000 word forms) and 96 morphosyntactic classes. Tokens not in the dictionary were assigned sets of classes according to their morphological features. Parameters were initially estimated from smoothed relative frequencies in a hand-tagged text containing 118,000 words, then reestimated on 47M words from the Canadian Hansard corpus, and finally smoothed using a non-linear backoff method similar to that of Katz (1987).

The trigram was trained on the same Hansard corpus as the triclass, with 75% of the corpus used for relative-frequency parameter estimates, and 25% used to reestimate interpolation coefficients.

3.1.2. *Translation Model*

Our model for $p(t|\mathbf{s})$ is based on the IBM translation models 1 and 2. For a given source text, model 1 defines a Hidden Markov Model (HMM) of the target language whose states correspond to source text tokens. Output distributions depend only on the corresponding source text *words*, and all state transition distributions are uniform. In other words, each target-text token is considered to arise from exactly one randomly-chosen source-text word, independent of both source- and

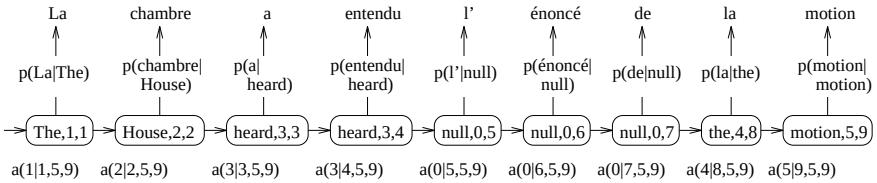

Figure 1. A plausible state sequence by which the translation model 2 for the English source sentence *The House heard the motion* might produce the French target sentence shown. States are triples (s, j, i), where s is an English word, j is the position of s in the English sentence, and i is a position in the French sentence. Vertical arrows indicate productions from states, and are labelled with the corresponding translation parameters $p(t|s)$; horizontal arrows indicate state transitions, and are labelled with the corresponding alignment parameters $a(j|i, S, T)$. The joint probability of the state sequence with the generated sentence is the product of the translation parameters times the product of the alignment parameters.

target-text context. To accomodate target language words that have no strong correlation to any in the source text, the model also contains a special *null* state.

Model 2 is similar to model 1, except that state transition probabilities depend on the positions of both source and target tokens, as well as on the lengths of the source and target texts: states are augmented with source and target position components, and there is a topographical constraint that a state's target position component must match the position of the token it generates. Thus each target-text token still arises from exactly one source-text token, but with a probability that also depends on the distance between the two (see figure 1).

Due to the simple form of their state transition distributions, models 1 and 2 have the propery that—unlike HMM's in general—they generate tokens independently. The total probability of a target-text token is just the average of the probabilities with which it is generated by each source text token; for model 2, this is a weighted average that takes the distance from the generating token into account:

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

where $p(t_i|s_j)$ is a word-for-word translation probability, S and T are the lengths of the source and target texts, and $a(j|i, S, T)$ 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. in letting s_0 designate the null state.)

The models were reestimated from a large corpus of Canadian Hansard text automatically aligned to the sentence level using the method described in (Simard et al., 1992). To improve the training material, all non one-to-one alignments were filtered out, as well as all French sentences longer than 50 words; the resulting corpus consisted of 36M tokens of English and 37M tokens of French text.

We made several modifications to adapt model 2 for our purposes, as described below.

Target-text Length Independence

First, because the translator is expected to type the target text from left to right, its length will not be known when the system is attempting to complete a particular prefix. This means that the condition on target-text length must be dropped from the alignment parameters: $a(j|i,S,T) \approx a(j|i,S)$. Although this approximation weakens the model somewhat, particularly in its ability to predict strong associations near the ends of both texts, we have not found the difference to be crucial.

Translation Invariants

A second modification was based on the observation that certain types of English forms almost invariably translate into French either verbatim or after having undergone a predictable transformation. This implies that it is not necessary to store and compute translation probabilities for such forms; the model can be made more compact and its ability to cope with unknown tokens improved by treating them specially.

We identified three classes of "translation invariant" forms: proper nouns, numbers, and special alphanumeric codes like *B-52*. To detect proper nouns, we used a statistical English tagger modified to explicitly capture the important cue provided by letter case—the tagger's lexical parameters are of the form $p(w \mid t, \textit{lead}) = p(\overline{w} \mid t)\, p(\textit{cp} \mid t, \textit{lead})$, where $\overline{w}$ is a case-normalized version of the word w, t is a part-of-speech tag, *cp* is w's case pattern (capitalized, uppercase, lowercase, or other), and *lead* is a boolean variable that is true iff w begins a sentence. Numbers and codes were detected using finite state matching. A stoplist was used to filter out certain frequent "false invariants", including proper nouns like *United States* that do not translate verbatim into French, and numbers like *10* that tend to get translated into a fairly wide variety of forms.

During training, invariant tokens in each source segment were replaced by tags indicating their class (with serial numbers when necessary to distinguish different invariants in the same segment). Any matching tokens—or, for numbers, legitimate variants thereof (see table I)—in the corresponding target segment were also replaced by the appropriate tag. This strategy reduced the number of parameters in the model by about 15%. During evaluation, a similar replacement operation was carried out and the translation probabilities of paired invariants were obtained from those of the tags to which they mapped; table I gives a sample of these.

Table I. French variants of an English number, and translation probabilities for the first invariant pair of each class found in a given pair of translated segments.

English	French
1,234.56	1,234.56
	1 234,56
	1234,56

class	probability
proper noun	.838
number	.842
code	.582

Il est est important de informer informer et et et et leur leur leur
It is important to properly inform the people and increase their awareness

Figure 2. The most probable French target text of length 14 generated under model 2 by the English source sentence shown.

Local Consistency

Our final modification to model 2 was an attempt to compensate for one of its main weaknesses: when predicting a target-text token, it does not take into account which source tokens have already been used to generate the target text up to the current position. This implies that there is no explicit way of preventing a single source token from generating all of the target text, and conversely that there is no requirement for a given source token to generate anything at all. In practice, the linear constraints captured by the alignment parameters alleviate the problem by keeping source tokens from generating target tokens whose relative positions are too distant, and ensuring that each source token gets favoured at some point during generation of the target text. However, the alignment parameters are not sufficiently precise to prevent certain source tokens from dominating particular regions of the target text, as illustrated in figure 2. During word completion, this means that the translation model has a tendency to repeatedly propose words that have already been typed.

To remedy this, we added an *anti-cache* to model 2—a small window of recent tokens to which it is forced to assign a very low fixed probability. The optimum size of the window was established experimentally, as described in section 3.3.

3.2. GENERATING HYPOTHESES

The task of the generator is to identify words which match the current prefix to be completed, and pick a single best candidate using the evaluation function. In this section we describe several design features which are essential to performing this operation in real time.

3.2.1. *Active and Passive Vocabularies*

A well-established corollary to Zipf's law (Zipf, 1949) holds that a minority of words account for a majority of tokens in text. To capitalize on this, our system's French vocabulary is divided 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.

To facilitate finding the set of words that match a given prefix, the active vocabulary is represented as a trie. For efficiency, explicit lists of matching hypotheses

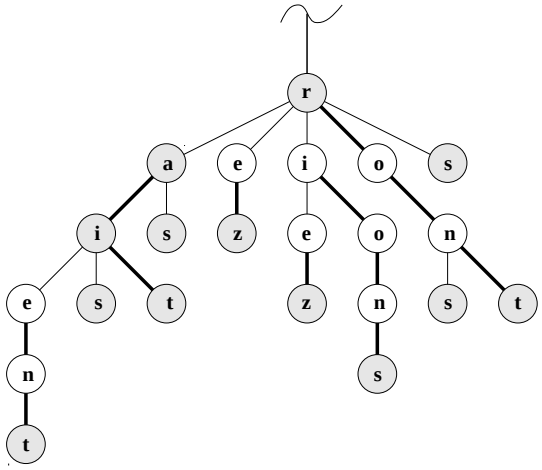

Figure 3. Memoized portion of the active vocabulary trie for the French prefix *parler*—heavy lines show best-child links and shaded nodes represent ends of words. The best extension for *parler* in the current context is *parleront*; if an *a* is appended by the translator, the new best extension *parlerait* can be retrieved from the best-child links without having to re-evaluate all 6 possible hypotheses. Alternately, if an *o* is appended, *parleront* will not be proposed again even though it remains the highest scoring extension of *parlero*; instead, *parlerons* will be retrieved by a new search.

are not generated; instead, words are scored during a recursive search over the portion of the trie below the current prefix. Repeat searches when the user appends a character are obviated in most situations by memoizing the results of the original search with a *best-child* pointer in each trie node (see figure 3).

An important characteristic of the generator is that it never proposes the same completion twice for a single word, even though one best completion may be compatible with several successive (increasingly longer) prefixes entered by the translator. This often permits an error at the end of a word—typically due to incorrect morphology—to be corrected without forcing the user to type or move past all of the intervening characters. Rejecting the current best hypothesis on the grounds that it has been proposed before necessitates a new search over the relevant portion of the trie. To avoid expensive repeat calls to the evaluator in this situation, the search makes use of previously-computed scores stored against each trie node.

If the current prefix has no extensions in the active vocabulary, or none that have not already been proposed, the active vocabulary is temporarily augmented with any matches found in the passive vocabulary. This is represented as a special trie in which common suffix patterns are stored only once (ie, the underlying structure is a directed acyclic graph rather than a tree), and variable-length encoding is used for all structural information. These techniques allow a large dictionary containing over 380,000 word forms to be maintained entirely in memory, using only about 475k bytes.

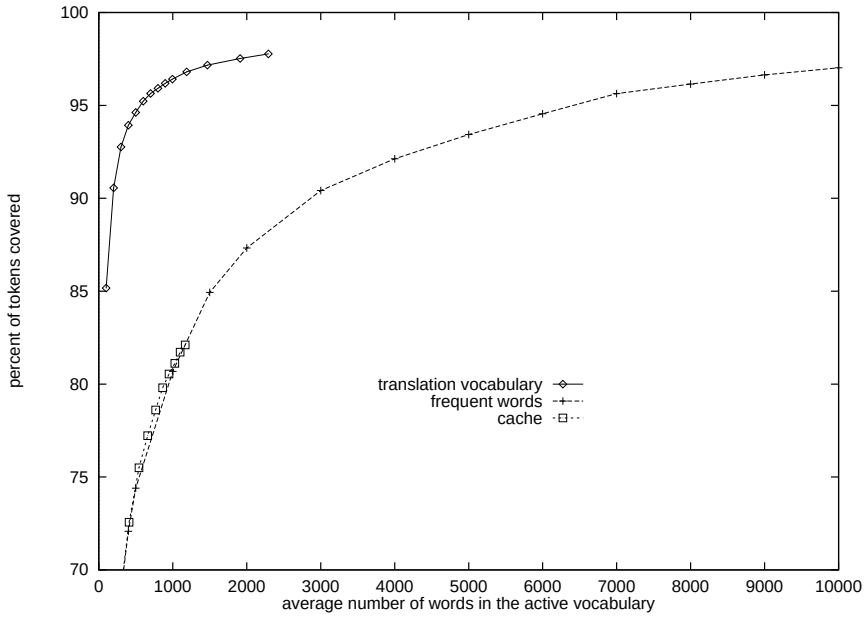

Figure 4. Coverage versus average vocabulary size in a 30,000 word test corpus for different methods of establishing the active vocabulary. Note that this shows *active* vocabulary coverage only; the total vocabulary includes contributions from the passive component, and its coverage is about 99.5% on this test text when active-vocabulary coverage is above 90%.

3.2.2. *Choosing the Active Vocabulary*

We considered three different sources for the active vocabulary: a static list of frequent words compiled from the training corpus; a fixed-length cache containing the most recently-encountered tokens in the test corpus; and a *translation vocabulary* containing the set of words to which the translation model 1 assigns the highest probabilities, given the current source text segment. To adjust for different target text lengths, the size of the translation vocabulary was set to a fixed multiple of the number of words (not tokens) in the source text. As figure 4 shows, the translation vocabulary performs dramatically better than the other two methods, achieving similar coverage with approximately an order of magnitude fewer words. Augmenting it with small cache and frequent-word vocabularies yielded no significant improvement. Although it takes time to compute a translation vocabulary (a second or so, depending on the size of the source text), this does not seriously affect the application because it can be performed as soon as a new source text segment is selected, when the translator can be expected to be busy reading.

3.2.3. *Case Handling*

The treatment of letter case is a tricky problem for hypothesis generation and one that cannot be ignored in an interactive application. Most words can appear in a

Table II. Test corpus

test text	aligned segments	English tokens	French tokens
A (Hansard–1986)	786	19457	21130
B (Hansard–1992)	1140	29886	32138
C (non-Hansard)	594	18881	21303

number of different case-variant forms and there are no simple and absolute rules that specify which form is appropriate in a particular context. The theoretically optimal approach would perhaps be to generate all combinatorially possible case variants in every context and let the evaluator pick the correct one, but of course this would be far too costly.

To cope with this situation, we adopted a heuristic strategy based on an idealized model of French case conventions in which words are divided into two classes. Class 1 contains "normal" words which are habitually written in lowercase but which may be capitalized (ie, with the first letter in uppercase and all others in lowercase) in some contexts, most often at the beginning of a sentence; and may also appear entirely in uppercase. Class 2 comprises proper nouns, acronyms and other forms which are written with a special and usually fixed case pattern containing at least one uppercase character, but which may also occasionally appear entirely in uppercase. Class 1 words generate capitalized hypotheses at the beginning of a sentence or when the completion prefix is capitalized; uppercase hypotheses when the completion prefix is uppercase and at least two characters long, or when the previous token was uppercase; and lowercase hypotheses otherwise. Class 2 words generate uppercase hypotheses under the same conditions as class 1 words, and verbatim hypotheses otherwise.

3.3. RESULTS

We conducted a series of tests to evaluate the performance of the system using different language models, translation models, and combinations of the two. The test corpus consisted of three automatically-aligned texts not used for training, two drawn from different parts of the Hansard corpus, and one from an unrelated corpus,[⋆] as shown in table II; text A was used to identify the best models, and the other two to corroborate the results. Based on the findings from section 3.2.2, all tests were restricted to active translation vocabularies of between 100 and 1,000 words inclusive. The main index of performance was the ratio of the number of characters in maximal[⋆⋆] automatically-completed suffixes to the total number of characters within tokens.

⋆ A paper on the competitiveness of the Canadian milk and dairy products industry.
⋆⋆ Under the assumption that the correct suffix is always accepted as soon as it is proposed.

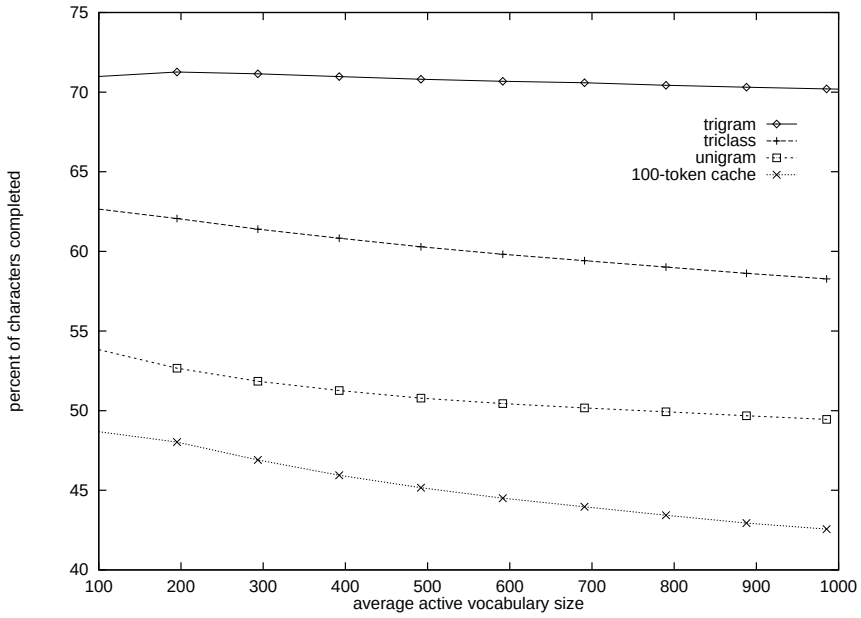

Figure 5. Language model performance versus average vocabulary size.

3.3.1. *Language Model*

The performances of various language models on text A are shown in figure 5. The interpolated trigram is obviously significantly superior to the other models; so much so, in fact, that no linear combination with any of the other models was able to improve on its performance. This is not a completely surprising result, although the triclass might have been expected to perform relatively better for a morphologically rich language like French. That is does not is likely due to the similarity between the test text and the training corpus, which is reflected in the fact that only about 3% of tokens in the former were new. On a more heterogeneous corpus, we would expect to be able to derive some benefit from the use of the cache and triclass models, but for the remainder of our testing we used only the pure trigram.

A noteworthy feature of figure 5 is that the performances of all models, with the exception of the trigram below 200 words, increase monotonically as the active vocabulary size *decreases*. This is exactly opposite to the behaviour one would expect from a fixed vocabulary, and it attests to the efficacy of the translation model. Since smaller vocabularies mean faster operation, this property implies that, remarkably, it is possible to simultaneously improve performance and speed.*

* A similar phonomenon is reported in (Brousseau et al., 1995). Due to the use of a passive vocabulary, this is not quite the whole story for our system. Smaller active vocabularies cause more prefixes to be sought in the passive vocabulary, and hence tend to be slower because passive lookups

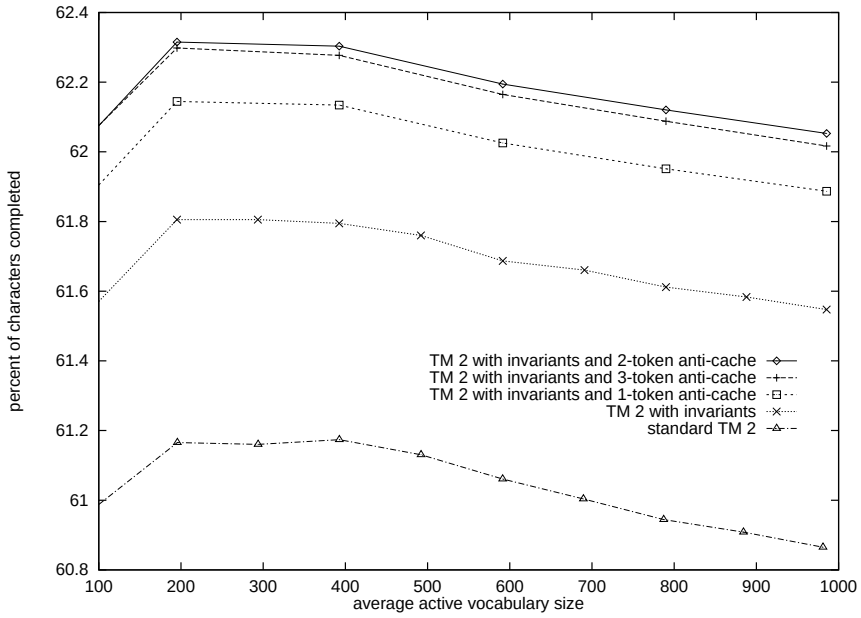

Figure 6. Translation model performance versus average vocabulary size.

It is also interesting to measure the trigram's performance when it operates without the benefit of *any* information from the source text, as would a unilingual text predictor such as described in (Darragh, 1992). Using a 3,000 word active vocabulary consisting of the most frequent words in the training corpus, the completion rate on text A was about 61%.

3.3.2. *Translation Model*

Figure 6 shows the performance on text A of the standard translation model 2 compared to modified versions which make use of invariants and an anti-cache. The best results are achieved by the invariant model with a 2-token anti-cache; this is about 1% better than the standard model, independent of active vocabulary size. It is interesting to note that the inverse relation between active vocabulary size and performance noted in the previous section holds to some extent here as well.

3.3.3. *Combined Model*

Figure 7 shows the performance on text A of various linear combinations of the best language and translation models. The peak of 72.5% of characters in correctly-completed suffixes occurs at a trigram weight of approximately .6, with a 500-word active vocabulary. The fact that this is less than 2% above the perfomance

typically retrieve very large sets of matching candidates. The active vocabulary sizes that give the fastest times are in the 300–400 word range.

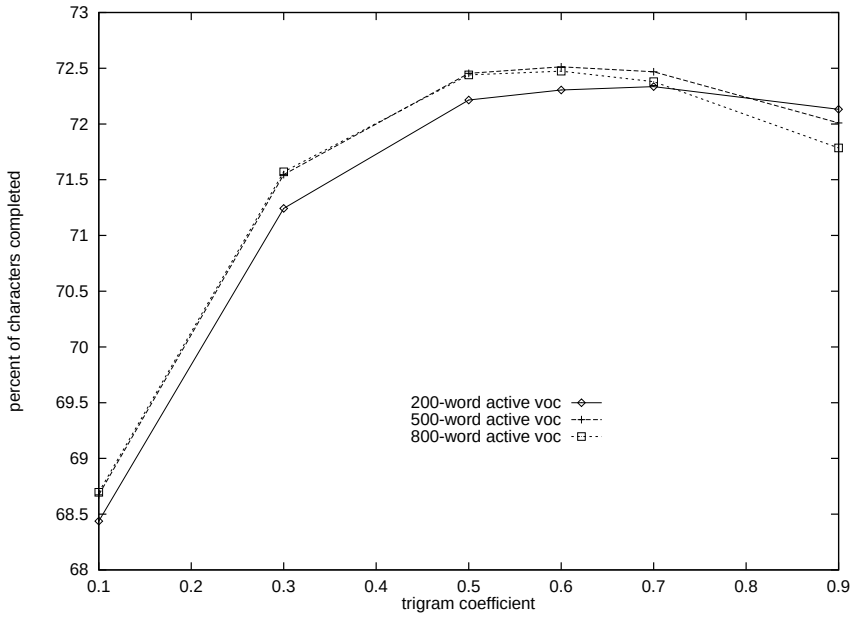

Figure 7. Combined trigram/translation model performance versus interpolation coefficient.

achieved by the trigram on its own is a result both of the translation model's prior contribution in non-linear fashion via the active vocabulary, and of the weakness of this method of combining language and translation components, as discussed in section 3.1.

3.3.4. *Summary of Performance*

Table III summarizes the performance of the optimum combined model on all three texts in the test corpus. Results for the non-Hansard text C were substantially worse than for either of the two Hansard texts; this is a reasonable estimate for a lower bound on the system's performance, since the training corpus is completely unrelated to text C, and since the model which was used lacks any capacity for dynamic adaptation apart from its use of translation invariants. The two rightmost columns in table III show estimates of the number of keystrokes (as opposed to characters) saved within tokens, according to two different scenarios. In the first, the translator uses a special command, costing one keystroke, to accept a proposal. In the second, acceptance consists merely in typing the character which follows the word—either a space or a punctuation mark;* completions are free in this accounting, but all

* Some French prefixes such as *jusqu'* which elide letters are not normally followed by either spaces or punctuation. We assume the system can detect these and automatically suppress the character used to effect the completion.

Table III. Character and estimated keystroke reduction on the test corpus.

test text	% characters	% keystrokes (I)	% keystrokes (II)
A (Hansard)	72.5	55.0	70.5
B (Hansard)	71.8	54.7	69.8
C (non-Hansard)	64.9	49.3	62.5

punctuation must be manually typed, and any spaces or punctuation characters in hand-typed prefixes are assessed a one-keystroke escape penalty.

The statistics given in table III will be proportional to effort saved if all characters require the same effort to type, but this is not likely to be the case. Intuitively, characters within long tokens should be harder to type than those in short tokens that are generally encountered more frequently. Fortunately, as shown in figure 8, completion performance actually increases with token length; furthermore, completions within medium and long tokens account for a substantial proportion of the total.

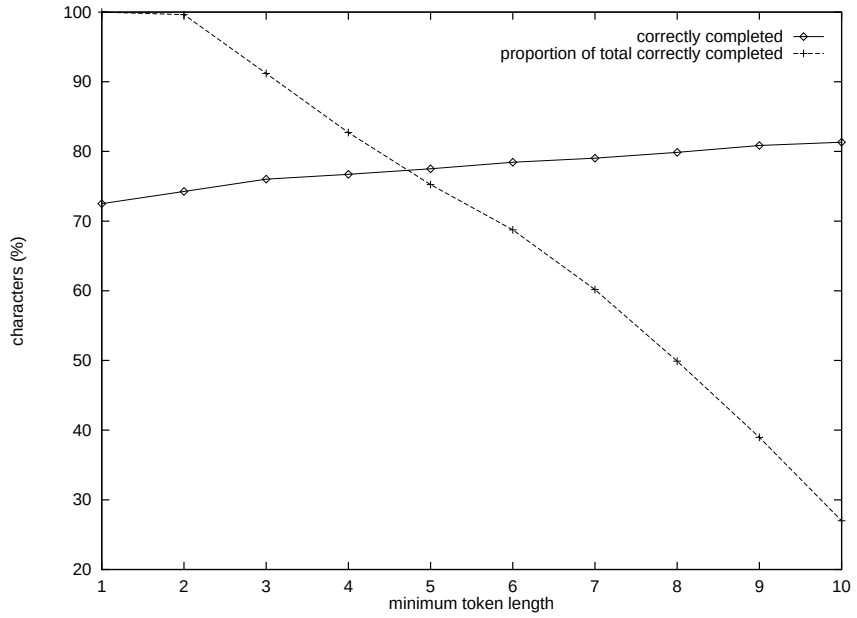

Figure 8. Completion performance versus token length in text A. For example, among tokens of six characters or more, 78.4% of characters are in correctly completed suffixes; this represents 68.7% of the total number of characters in correctly completed suffixes.

Figure 9 contains a detailed record of a completion session for a sentence pair from text A on which the system attains approximately its average performance.

A final very important aspect of performance is the speed with which the next completion hypothesis can be retrieved. The average processing rate on a Sparc-

```
Cette           C             /Ce C/ette
conférence                    /conférence
,               ,             /et
portant         port          /qui p/our po/urra por/tent port/ant
sur                           /sur
le              le            /les l/a
logement        l             /marché l/ogement
pour            p             /social p/our
les                           /les
personnes       p             /canadiens p/ersonnes
démunies        dém           /qui d/e dé/munis dém/unies
,               ,             /de
aura            aur           /les a/ux au/tomne aur/a
lieu            li            /des l/e li/eu
à                             /à
l'              l             /des l/'
automne         a             /1987 a/utomne
1987            1             /dernier 1/987
.               .             /de
```

Figure 9. A sample completion session for the English source sentence *This conference, which will deal with housing for the needy, will be held in the fall of 1987*. The first column contains the French target sentence; the second the prefix typed by the translator; and the third the record of successive proposals for each token, with a slash separating prefix from proposed completion. (Note that the version of the system used here suppresses punctuation hypotheses.)

station 20 with 128M of memory was about 180 tokens per minute when using the combined model for evaluation, and aabout 300 tokens per minute when using the trigram model alone. Both of these averages are comfortably faster than human typing rates, although we suspect that the combined model would occasionally cause slight delays when the translation of an especially long source text segment was being typed.

4. Conclusion and Future Work

This paper sets out a new approach to IMT which we feel is better suited to the needs of skilled translators than existing ones. Its main features are:

1. The computer assists the human, rather than vice versa: the purpose of interaction is to establish the target text directly, not just to enhance the process of machine translation.
2. The target text serves as the medium of interaction: the human translator issues directives in the form of characters, words, or, possibly, more abstract properties, and the computer reacts to each with a revised proposal for all or part of the target text.

To show that our proposal is feasible, at least in a rudimentary form, we have written a program which uses statistical language and translation models to pre-

dict completions for words in a French target text, given its English translation. Although we have not yet attempted to evaluate the program's usefulness as a tool for translators, we estimate from tests on the Canadian Hansard corpus that it could reduce the number of keystrokes needed to type target text words by approximately 70%.

There are numerous possibilities for extending the work presented here. We plan to experiment with better ways of combining source- and target-text based predictions, as discussed in section 3.1, and also with models capable of adapting to context. Another possibility is to investigate more sophisticated interfaces, such as those that provide lists of alternate hypotheses or explicit functions for morphological repair. The main challenge is obviously to predict more of the target text than just the next word, and thus begin to realize more of the potential of our approach.

Acknowledgements

We would like to thank Elliott Macklovitch for helpful discussions and comments on drafts of this paper.

References

Hervé Blanchon. Perspectives of DBMT for monolingual authors on the basis of LIDIA-1, an implemented mock-up. In *COLING-94*, pages 115–119, Kyoto, August 1994.

Christian Boitet. Towards personal MT. In *COLING-90*, pages 30–35, Helsinki, August 1990.

J. Brousseau, C. Drouin, G.Foster, P. Isabelle, R. Kuhn, Y. Normandin, and P. Plamondon. French speech recognition in an automatic dictation system for translators: the TransTalk project. In *Eurospeech 95*, pages 193–196, Madrid, Spain, September 1995.

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

Ralf D. Brown and Sergei Nirenburg. Human-computer interaction for semantic disambiguation. In *COLING-90*, pages 42–47, Helsinki, August 1990.

John J. Darragh and Ian H. Witten. *The Reactive Keyboard*. Cambridge University Press, 1992.

A.-M. Derouault and B. Merialdo. Natural language modeling for phoneme-to-text transcription. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):742–743, November 1986.

Eurolang. Eurolang Optimizer, product description, 1995.

Robert Frederking, Dean Grannes, Peter Cousseau, and Sergei Nirenburg. An MAT tool and its effectiveness. In *Proceedings of the DARPA HLT Workshop*, Princeton, NJ, 1993.

IBM. IBM Translation Manager, product description, 1995.

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

Slava M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-35(3):400–401, March 1987.

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

M. Kugler, G. Heyer, R. Kese, B. von Kleist-Retzow, and G. Winkelmann. The Translator's Workbench: An environment for multi-lingual text processing and translation. In *Proceedings of MT Summit III*, pages 81–83, Washington, July 1991.

Roland Kuhn and Renato De Mori. A cache-based natural language model for speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):570–583, June 1990.

Hiroshi Maruyama and Hideo Watanabe. An interactive Japanese parser for machine translation. In *COLING-90*, pages 257–262, Helsinki, August 1990.

Alan Melby. On human-machine interaction in translation. In Sergei Nirenburg, editor, *Machine Translation, Theoretical and methodological issues*, pages 145–154. Cambridge University Press, 1987.

Sergei Nirenburg. Tools for machine-aided translation: The CMU TWS. *META*, 37(4):709–720, 1992.

Eugenio Picchi, Carol Peters, and Elisabetta Marinai. A Translator's Workstation. In *COLING-92*, pages 972–976, Nantes, August 1992.

Michel Simard, George F. Foster, and Pierre Isabelle. Using cognates to align sentences in bilingual corpora. In *TMI-4*, Montreal, Canada, 1992.

Masaru Tomita. Feasibility study of personal/interactive machine translation systems. In Sergei Nirenburg, editor, *TMI-1*, pages 289–297, Colgate University, Hamilton, New York, August 1985. Department of Linguistics, Colgate University.

Trados. Trados Translators Workbench, product description, 1995.

P. J. Whitelock, M. McGee Wood, B. J. Chandler, N. Holden, and H. J. Horsfall. Strategies for interactive machine translation: the experience and implications of the UMIST Japanese project. In *COLING-86*, pages 329–334, Bonn, 1986.

Rémi Zajac. Interactive translation: A new approach. In *COLING-88*, pages 785–790, Budapest, 1988.

G. K. Zipf. *Human Behavior and the Principle of Least Effort*. Addison-Wesley, Reading, MA, 1949.

Address for correspondence: {foster,isabelle,plamondon}@citi.doc.ca