Word Completion: A First Step Toward Target-Text Mediated IMT

George Foster, Pierre Isabelle, and Pierre Plamondon
Centre for Information Technology Innovation (CITI)
1575 Chomedey Blvd.
Laval, Quebec, Canada, H7V 2X2
{foster,isabelle,plamondon}@citi.doc.ca

Abstract

We argue that the conventional approach to Interactive Machine Translation is not the best way to provide assistance to skilled translators, and propose an alternative whose central feature is the use of the target text as a medium of interaction. We describe an automatic word-completion system intended to serve as a vehicle for exploring the feasibility of this new approach, and give results in terms of keystrokes saved in a test corpus.

1 Introduction

Machine translation is usually significantly inferior to human translation, and for most applications where high-quality results are needed it must be used in conjunction with a human translator. There are essentially three ways of organizing the process by which a person and a machine cooperate to produce a translation: *preedition*, in which the person's contribution takes the form of a source-text analysis and occurs before the MT system is brought to bear; *postedition*, in which the translator simply edits the system's output; and *interactive* MT (IMT), which involves a dialog between person and machine. Of the three, IMT is the most ambitious and theoretically the most powerful. It has a potential advantage over postedition in that information imparted to the system may help it to avoid cascading errors that would later require much greater effort to correct; and it has a potential advantage over preedition in that knowledge of the machine's current state may be useful in reducing the number of analyses the human is required to provide.

Existing approaches to IMT (Blanchon, 1994; Boitet, 1990; Brown and Nirenburg, 1990; Kay, 1973; Maruyama and Watanabe, 1990; Whitelock et al., 1986; Zajac, 1988) place the MT system in control of the translation process and for the most part limit the human's role to performing various source language disambiguations on demand. Although this arrangement is appropriate for some applications, notably those in which the user's knowledge of the target language may be limited, or where there are multiple target languages, it is not well suited to the needs of professional or other highly skilled translators. The lack of direct human control over the final target text (modulo postedition) is a serious drawback in this case, and it is not clear that, for a competent translator, disambiguating a source text is much easier than translating it. This conclusion is supported by the fact that true IMT is not, to our knowledge, used in most modern translator's support environments, eg (Eurolang, 1995; Frederking et al., 1993; IBM, 1995; Kugler et al., 1991; Nirenburg, 1992; Trados, 1995). Such environments, when they incorporate MT at all, tend to do so wholesale, giving the user control over whether and when an MT component is invoked, as well as extensive postediting facilities for modifying its output, but not the ability to intervene while it is operating.

In our view, this state of affairs should not be taken as evidence that IMT for skilled translators is an inherently bad idea. We feel that there is an alternate approach which has the potential to avoid most of the problems with conventional IMT in this context: use the target text as a medium of communication, and have the translator and MT system interact by making changes and extensions to it, with the translator's contributions serving as progressively informative constraints for the system. This arrangement has the advantage of leaving the translator in full control of the translation process, of diverting his or her attention very little from the object of its natural focus, and of necessitating a minimum of interface paraphernalia beyond those of a word processor. It can in principle accomodate a wide range of MT proficiencies, from very high, in which the system might be called upon to propose entire translations and modify them in response to changes made by the translator; to very low, in which its chief contribution will be the reduction of typing labour.

The aim of this paper is to explore the feasibility of this *target-text mediated* style of IMT in one particularly simple form: a word-completion system which attempts to fill in the suffixes of target-text words from manually typed prefixes.[1] We describe a prototype completion system for English to French translation which is based on simple statistical MT techniques, and give measurements of its performance in terms of characters saved in a test corpus. The system has not yet been integrated with a word processor, so we cannot quantify the amount of actual time and effort it would save a translator, but it seems reasonable to expect this to be fairly well correlated with total character savings.

2 Word Completion

Our scenario for word completion supposes that a translator works on some designated segment of the source text (of approximately sentence size), and elaborates its translation from left to right. As each target-text character is typed, a proposed completion for the current word is displayed; if this is correct, the translator may accept it and begin typing the next word. Although more elaborate completion schemes are imaginable, including ones that involve the use of alternate hypotheses or provisions for morphological repair, we have opted against these for the time being because they necessitate special commands whose benefit in terms of characters saved would be difficult to estimate.

The heart of our system is a completion engine for English to French translation which finds the best completion for a French word prefix given the current English source text segment under translation, and the words which precede the prefix in the corresponding French target text segment. It comprises two main components: an *evaluator* which assigns scores to completion hypotheses, and a *generator* which produces a list of hypotheses that match the current prefix and picks the one with the highest score.

[1] This idea is similar to existing work on typing accelerators for the disabled (Demasco and McCoy, 1992), but our methods differ significantly in many aspects, chief among which is the use of bilingual context.

3 Hypothesis Evaluation

Each score produced by the evaluator is an estimate of $p(t\,|\,\tilde{\mathbf{t}}, \mathbf{s})$, the probability of a target-language word t given a preceding target text $\tilde{\mathbf{t}}$, and a source text $\mathbf{s}$. For efficiency, this distribution is modeled as a simple linear combination of separate predictions from the target text and the source text:

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

The value of λ was chosen so as to maximize completion performance over a test text (see section 5).

3.1 Target-Text Based Prediction

The target-text based prediction $p(t\,|\,\tilde{\mathbf{t}})$ comes from an interpolated trigram language model for French, of the type commonly used in speech recognition (Jelinek, 1990). It was trained on 47M words from the Canadian Hansard Corpus, with 75% used to make relative-frequency parameter estimates and 25% used to reestimate interpolation coefficients.

3.2 Source-Text Based Prediction

The source text prediction $p(t\,|\,\mathbf{s})$ comes from a statistical model of English-to-French translation which is based on the IBM translation models 1 and 2 (Brown et al., 1993). Model 1 is a Hidden Markov Model (HMM) of the target language whose states correspond to source text tokens (see figure 1), with the addition of one special *null* state to account for target text words that have no strong direct correlation to any word in the source text. The output distribution of any state (ie the set of probabilities with which it generates target words) depends only on the corresponding source text *word*, and all next-state transition distributions are uniform. Model 2 is similar to model 1 except that states are augmented with a target-token position component, and transition probabilities depend on both source and target token positions,[2] with the topographical constraint that a state's target-token position component must always match the current actual position. Because of the restricted form of the state transition

[2] Along with source and target text lengths in Brown et al's formulation, but these are constant for any particular HMM. The results presented in this paper are optimistic in that the target text length was assumed to be known in advance, which of course is unrealistic. However, (Dagan et al., 1993) have shown that knowledge of target-text length is not crucial to the model's performance.

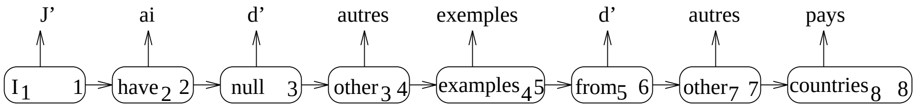

Figure 1: A plausible state sequence by which the HMM corresponding to the English sentence *I have other examples from many other countries* might generate the French sentence shown. The state-transition probabilities (horizontal arrows) are all 1/9 for model 1, and depend on the next state for model 2, eg $p(\langle from_5, 6\rangle \mid \cdot) = a(5|6)$. The output probabilities (vertical arrows) depend on the words involved, eg $p(\text{d'} \mid \langle from_5, 6\rangle) = p(\text{d'} \mid \text{from})$.

matrices for these models, they have the property that—unlike HMM's in general—they generate target-language words independently. The probability of generating hypothesis t at position i is just:

$$p(t \mid \mathbf{s}, i) = \sum_{j=0}^{|\mathbf{s}|} p(t \mid s_j)\, a(j \mid i)$$

where s_j is the jth source text token (s_0 is a null token), $p(t \mid s_j)$ is a word-for-word translation probability, and $a(j \mid i)$ is a position alignment probability (equal to $1/(|\mathbf{s}| + 1)$ for model 1).

We introduced a simple enhancement to the IBM models designed to extend their coverage and make them more compact. It is based on the observation that there are (at least) three classes of English forms which most often translate into French either verbatim or via a predictable transformation: proper nouns, numbers, and special alphanumeric codes such as *C-45*. We found that we could reliably detect such "invariant" forms in an English source text using a statistical tagger to identify proper nouns, and regular expressions to match numbers and codes, along with a filter for frequent names 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.

When the translation models were trained, invariant tokens in each source text segment were replaced by special tags specific to each class (different invariants occuring in the same segment were assigned serial numbers to distinguish them); any instances of these tokens found in the corresponding target text segment were also replaced by the appropriate tag. This strategy reduced the number of parameters in the models by about 15%. When evaluating hypotheses, a similar replacement operation is carried out and the translation probabilities of paired invariants are obtained from those of the tags to which they map.

Parameters for the translation models were reestimated from the Hansard corpus, automatically aligned to the sentence level using the method described in (Simard et al., 1992), with non one-to-one alignments and sentences longer than 50 words filtered out; the retained material consisted of 36M English words and 37M French words.

4 Hypothesis Generation

The main challenge in generating hypotheses is to balance the opposing requirements of completion accuracy and speed—the former tends to increase, and the latter to decrease with the number of hypotheses considered. We took a number of steps in an effort to achieve a good compromise.

4.1 Active and Passive Vocabularies

A well-established corollary to Zipf's law 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 used for generation, and a much larger *passive* part which comes into play only when the active vocabulary contains no extensions to the current prefix.

Space requirements for the passive vocabulary were minimized by storing it as a special trie in which common suffix patterns are represented only once, and variable-length coding techniques are used for structural information. This allows us to maintain a large dictionary containing over 380,000 forms entirely in memory, using about 475k bytes.

The active vocabulary is also represented as a trie. For efficiency, explicit lists of hypotheses are not generated; instead, evaluation is performed during a recursive search over the portion of the trie below the current completion prefix. Repeat searches when the prefix is extended by one 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 2).

4.2 Dynamic Vocabulary

To set the contents of the active vocabulary, we borrowed the idea of a *dynamic vocabulary* from (Brousseau et al., 1995). This involves using

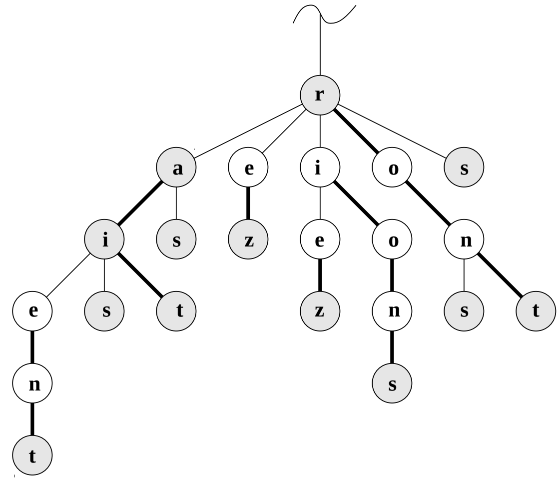

Figure 2: Memoized portion of the active vocabulary trie for the French prefix *parler*—heavy lines show best-child links and shaded nodes represent valid word ends. The current best candidate is *parleront*; if an *a* is appended by the translator, the new best candidate *parlerait* can be retrieved from the best-child links without having to re-evaluate all 6 possible hypotheses.

translation model 1 to compute a list of the n most probable target text words (including translation invariants), given the current source text segment. As figure 3 illustrates, compared to an alternate method of statically choosing the n most frequent forms in the training corpus, use of a dynamic vocabulary dramatically reduces the average active vocabulary size required to achieve a given level of target text coverage. Motivated by the fact that recent words tend to recur in text, we also added all previously encountered target-text tokens to the dynamic vocabulary.

4.3 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 number of different case-variant forms and there are no simple and absolute rules that specify which is appropriate in a particular context. 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 words are those which are normally written in lowercase; class 2 words are those such as proper nouns which normally take a special case pattern containing at least one uppercase character. 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; and lowercase hypotheses otherwise. Class 2 words generate uppercase hypotheses under the same conditions as class 1 words, otherwise verbatim hypotheses.

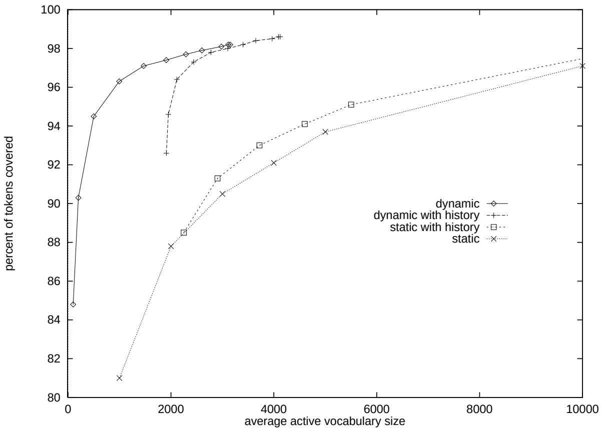

Figure 3: Target text coverage versus active vocabulary size, for static and dynamic methods. The *with history* curves reflect the addition of previously encountered target text tokens to the active vocabulary.

5 Results

We tested the completion engine on two different Hansard texts not in our training corpus. Text A, containing 786 (automatically) aligned pairs, 19,457 English and 21,130 French tokens, was used to determine optimum parameter settings; text B, containing 1140 (automatically) aligned pairs, 29,886 English and 32,138 French tokens, was used to corroborate the results. Tests were conducted with a 3000-word dynamic active vocabulary augmented with all encountered target-text forms.

Four measures of completion performance were used. All assume that the translator will accept a correct completion proposal as soon as it is made (ie, without typing further). The most direct index is the proportion of characters in correctly-completed suffixes. Related to this is the proportion of correctly-anticipated characters: those in correct suffixes plus any that match the next character the translator will have to type. The final two measures are intended to approximate the number of keystrokes saved within words. The first assumes that the translator uses a special command, costing one keystroke, to accept a proposal. The second assumes that acceptance consists merely in typing the character which follows the word—either a space or a punctuation mark.[3] Completions are free in this accounting,

[3] Some French prefixes such as *jusqu'* which elide

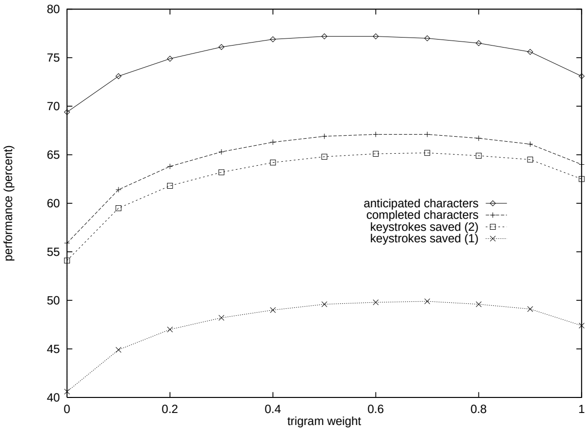

Figure 4: Combined trigram/translation model performance versus trigram weight λ.

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

Figure 4 shows the performance of the system for various values of the trigram coefficient λ. A noteworthy feature of this graph is that interpolation improves performance over the pure trigram by only about 3%. This is due in large part to the fact that the translation model has already made a contribution in non-linear fashion through the dynamic vocabulary, which excludes many hypotheses that might otherwise have misled the language model.

Another interesting characteristic of the data is the discrepancy between the number of correctly anticipated characters and those in completed suffixes. Investigation revealed the bulk of this to be attributable to morphological error. In order to give the system a better chance of getting inflections right, we modified the behaviour of the hypothesis generator so that it would never produce the same best candidate more than once for a single token; in other words, when the translator duplicates the first character of a proposal, the system infers that the proposal is wrong and changes it. As shown in table 1, completion performance improves substantially as a result. Figure 5 contains a detailed record of a completion session that points up one further deficiency in the system: it proposes punctuation hypotheses too often. We found that simply suppressing punctuation in the generator led to another small increment in keystroke savings, as indicated in table 1.

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.

measure (% chars)	method			
	text A			text B
	std	PBHR	P+NP	P+NP
anticipated	77.2	80.0	79.2	78.9
completed	67.1	73.6	72.6	72.2
keystrokes1	65.1	71.8	72.3	71.9
keystrokes2	49.8	54.6	55.1	55.1

Table 1: Final performance figures. PBHR stands for *previous-best-hypothesis rejection*, and P+NP for PHBR without punctuation hypotheses.

6 Conclusion

The work described in this paper constitutes a rudimentary but concrete first step toward a new approach to IMT in which the medium of interaction is simply the target text itself. In contrast with previous interactive approaches, the translator is never expected to perform tasks that are outside the realm of translation proper (such as advising a machine about common sense issues). In line with the spirit of truly interactive approaches, the translator is called upon early enough to guide the system away from a "raw machine translation" he or she would rather not have to revise. And in fact the machine is now the one required to revise its own copy, making use of every keystroke entered by the translator to steer itself in a useful direction.

This strikes us as the "proper place" of men and machines in IMT, and we intend to continue exploring this promising avenue in our future research.

References

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

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

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

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

Peter F. Brown, Stephen A. Della Pietra, Vincent Della J. Pietra, and Robert L. Mercer. 1993.

Nous	+	/Nous
réalisons	réal+	/avons r/endre ré/aliser réa/lise réal/isons
tous	t+	/des t/ous
que	q+	/les q/ue
le	+	/le
Canada	C+	/gouvernement C/anada
comme	c+	/, c/omme
bien	bi+	/un b/eaucoup bi/en
d'	+	/d'
autres	+	/autres
pays	p+	/, p/ays
,	+	/,
riches	r+	/, r/iches
ou	+	/ou
pauvres	+	/pauvres
,	+	/,
a	a+	/les
beaucoup	b+	/été b/eaucoup
trop	t+	/de t/rop
de	+	/de
ses	se+	/temps s/ervices se/s
citoyens	c+	/trop c/itoyens
qui	q+	/. q/ui
...		

Figure 5: A sample completion run for the English source sentence *We all realize that like many other countries, rich or poor, Canada has too many citizens who cannot afford decent housing.* The first column contains the French target sentence; the second the prefix typed by the translator, followed by a plus sign; and the third the record of successive proposals for each token, with a slash separating prefix from proposed completion.

The mathematics of machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–312, June.

Ido Dagan, Kenneth W. Church, and William A. Gale. 1993. Robust bilingual word alignment for machine aided translation. In *Proceedings of the Workshop on Very Large Corpora (ACL 93)*, Columbus, Ohio.

Patrick W. Demasco and Kathleen F. McCoy. 1992. Generating text from compressed input: An intelligent interface for people with severe motor impairments. *CACM*, 35(5), May.

Eurolang. 1995. Eurolang Optimizer, product description.

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

IBM. 1995. IBM Translation Manager, product description.

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

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

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

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

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

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

Trados. 1995. Trados Translators Workbench, product description.

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

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