Machine Translation In A Monolingual Environment

Xiuming Huang*
AI Systems Section
Telecom Research Laboratories
770 Blackburn Rd, Clayton 3168
Australia
xiuming@trlamct.trl.oz.au

Abstract

Machine translation systems can gain wider application if they can be operated in a monolingual environment. To design such systems, the major issue is how to assure the user that the translation is correct without understanding either the source or the target language. This paper first discusses the various approaches to assuring correct translation in a monolingual environment and then describes the selective confirmation approach adopted in our experimental machine translation system for the target language inexpert.

1 Introduction

Machine translation (MT) systems commercially available today are mostly designed for multi-lingual users, normally with post-editing facilities (Melby, 1987). To cater for the needs of users who are not skilled in either the source language or the target language, such as companies requiring foreign language communication in a language not spoken by staff, an MT system must provide facilities for the user to check into the quality of the translation, without knowing the foreign language. In this paper two presumptions are made: first, we discuss only translation from the user's native language into a foreign language (or "export translation"); second, we will restrict the quality checking to accuracy checking only, leaving style considerations aside.

Quality checking is important for at least two reasons. The first is, machine translation has not yet reached the stage when translation accuracy can be guaranteed without some form of human assistance. Secondly, even if an MT system is capable of resolving various kinds of ambiguities (and as a result producing accurate translations), there might be

* The work was carried out when the author was with the Department of Computer Science, The University of Melbourne, Parkville, Vic. 3052, Australia

situations when the ambiguity is not resolvable within a limited context. ("The conductor died of a heart attack yesterday. His family and friends were shocked... " [the story goes on] The writer knows who the conductor was (a bus conductor or an orchestra conductor), but the system would have difficulty deciding it).

The central issue of designing an MT system to be used in a monolingual environment, therefore, is how to assure the user that the translation is accurate.

In the following sections we will first discuss approaches to assuring correct translation, and then describe the selective confirmation approach adopted in our MT system.

2 Approaches to Assuring Correct Translation

2.1 Back Translation

The seemingly most natural way of finding out whether the translation into the target language (henceforth *forward translation)* is correct is to translate the forward translation back into the source language (back *translation).* This approach can indeed expose the wrong forward translations if the back translations differ significantly from the original. There have been many anecdotes, some trustworthy and others apocryphal, about how wrong a forward translation could be. A recent one involves Mrs. Thatcher's visit to a foreign company. When asked to give a sentence for the company's MT system to translate, she said: "It is an honour and a pleasure to be visiting this company today." The system translated it forward and then back, producing "this company, having been visited by me today, is honourable." (*Guardian* October 1, 1989) Mrs. Thatcher may have been pleased with the back translation, though the system developers may not share her feelings.

Despite the value of the back translation method, it has its complexities and is not reliable. An MT system may either employ grammars designed to be reversible, i.e., to be used for both generation and analysis, or employ separate grammars for generation and analysis. In the former case, theoretically whatever the forward translations are, if the grammars are truly reversible the back translation should always produce sentences in the source language that closely approximate the original[1]. As a result we have no way to tell whether the forward translation is correct or not.

For example the sentence

(la) John liked the girl in the park.

might be wrongly translated into Chinese as

(1b) Yuehan zai gongyuan li xihuan nage gunian.

(John, in the park, liked the girl)

If we use the same Chinese grammar responsible for generating the forward translation to analyze (1b) and then use the same English grammar to generate the back translation, the

[1] Whether this degree of reversibility is achievable is an open question. As one of the reviewers of the paper remarks, the back translation may deviate considerably from the original since the content of any representations used for transfer (in transfer-based systems) can be critical, and a grammar used for generation needs typically to be more constraining than does an analysis grammar (at least if the grammars are based on unification). The range of such deviations and means to control them can be interesting research topics in their own right.

result will be (not surprisingly) "John liked the girl in the park", which will give the user the impression that the forward translation is perfect.

If the back translation is done in a system where separate grammars are used for generation and analysis, then the back translation itself may be incorrect inasmuch as the forward translation might be, so that a correct forward translation may be *wrongly* translated back, or vice versa. In either case there is no guarantee of accuracy.

Back translation, therefore, cannot be used as a reliable means to check the translation accuracy in an MT system for the target language inexpert.

2.2 Paraphrasing

In this approach, the system generates paraphrase(s) for the original sentence, based upon the syntactic and semantic analysis result, before passing it for further processing (by the transfer and/or generation component). The user checks the paraphrase and gives the system directives by either confirming or rejecting it. For example, for the input sentence

(2a) The man saw the woman in the park with the telescope.

the system might produce the paraphrase

(2b) With the telescope, the man saw the woman who was in the park.

If the user is satisfied with a suggested paraphrase s/he directs the system to complete the translation by using the analysis result (responsible for the paraphrase). An important assumption is made here that the system's generation phase is reliable, so that a correct analysis of the input sentence will guarantee a good translation.

The problem with the paraphrasing approach is that the recovery may come in late, after a large amount of time is spent doing all the syntactic and semantic analysis required for generating the paraphrase. If the input sentence is long and complex the cost can be high. Furthermore, it may not be unusual that only after several trials the user finally accepts an appropriate paraphrase, with all the previous efforts wasted.

2.3 Pre-editing

This is the approach adopted by many commercial MT systems (very often paired with post-editing). It requires the user to edit the input sentences before they are passed to the system for analysis and subsequent processing. The machine has a predefined translation capacity which must be known to the user so that anything in the text which may cause difficulties to the system will be removed or rewritten. For example, for the following sentence

(3a) The woman cannot bear children.

if the user knows the system would have difficulties resolving the ambiguous word "bear", s/he can rewrite the sentence as follows (if that is what s/he wants to say):

(3b) The woman cannot give birth to children.

To apply pre-editing, a set of rules must first be devised to set up lexical and structural constraints, then the user must keep the rules in mind and apply them consistently. This may involve expensive training of system users, although systems like Titus can sidestep the problem by having the system impose on the user obedience to the rules. In both cases strong restrictions are olaced on the system use in practice.

2.4 Interactive Disambiguation

Interactive disambiguation can take place at two levels. At the lexical level, whenever an ambiguous word is encountered, the user can be asked to help. (Carbonell and Tomita, 1987) gives an example of resolving word-sense ambiguity in this approach:

The word "pen" means:
1) a writing pen
2) a play pen
NUMBER?>

The problem with this approach is, given that any natural language is highly polysemous, the frequent occurrence of ambiguity at the lexical level will unnecessarily prolong the translation process and easily bore the user. Moreover, with lexical items with many different senses, it may become very difficult to pinpoint one in particular from a screenful of choices.

At the structural level, ambiguities can also be referred to the user for disambiguation, as is done in *Ntran,* an English-to-Japanese prototype MT system for monolingual users (Wood and Chandler, 1988). For the sentence

(4) The cursor corresponds to the puck position on the tablet.

Ntran asks the user to choose either 1 or 2 from the following:

1 on is location of position
2 on is location of correspond

(here "on" represents "on the tablet".) A severe problem with this listing of all choices is that the number of possible interpretations of an ambiguous structure can reach the hundreds (Church and Patil, 1982), making their handling very difficult.

3 Selective Confirmation

3.1 Basic Structure of the Parser

In this section we describe the approach taken by our system to ensuring correct forward translation. The basic idea is to let the machine do most of the work without human interference, and only at certain decision making points ask for human assistance. To understand how those decision making points are chosen, we first briefly describe the parsing mechanism of the system when running without any human assistance.

The basic structure of the English parser of our system is illustrated in (5) and (6).

```
(5) SENTENCE -> SUBJECT_NP, VERB,SUBJ_VERB_MATCH,OBJECT_NP,
                VERB_OBJ_MATCH, PP, PP_ATTACHMENT.
```

The schema in (5) states that in parsing a sentence the parser tries to find a subject noun phrase, followed by a finite verb, then executes a semantic match along the line of selectional restrictions between the noun phrase and the verb; upon the successful execution of the match, it proceeds to find the object noun phrase (suppose the verb is transitive), then carries out a semantic match between the verb and the object. Finally, it analyses any possible prepositional phrases and decides their attachment by applying semantic knowledge.

```
(6) NP -> DETERMINER, ADJECTIVES, NOUN, ADJ_NOUN_MATCH, POST_MODIFIERS.
```

In the noun phrase grammar, the semantic match is carried out between adjectives and the head nouns.

There are three subroutines for carrying out lexical disambiguation: *adj_noun_match* for sense matching between adjectives and nouns (when constructing noun phrases, NPs), *subj_ verb_match* and *verb_obj_match* for sense matching between nouns and verbs (when establishing subject-verb relations and verb-object relations, respectively), and *pp_attachment* for sense matching between prepositions and nouns (when constructing prepositional phrases, PPs), as well as for deciding the relationship between the PPs and the preceding nouns or verbs (PP attachment).

Disambiguation is implemented both locally and globally: *adj_noun_match* is implemented when NPs are constructed, no matter whether they are subjects or objects at the sentence level, or objects of prepositions. The output of *adj_noun_match* would be a representation of an NP where both modifying adjectives (if any) and the head noun are substituted for with specific senses. This unambiguous representation is utilized in choosing the proper verb sense (*subj_verb_match*). If none can be found, the parser backtracks, trying to find another reading for the subject NP. The output of *subj_verb_match* is a noun sense - verb sense pair. The verb sense is then used in the *verb_obj_match*, once an object NP is tentatively constructed. The object NP is either accepted or rejected by the *verb_obj_match.* In the latter case, an alternative reading for the object NP is constructed, and tested by *verb_obj_match* again. If no object noun sense matches the particular verb sense, the system backtracks to redo *subj_verb_match* in an attempt to find another suitable verb sense which will take one of the readings for the object NP. The backtracking can go further to *subject_np* in an attempt to find alternative readings for the subject NP (by redoing *adj_noun_match*) if no verb sense selected for the previous reading of the subject NP matches any reading for the object NP.

3.2 Soliciting Confirmation at Chosen Points

From the above description it can be seen that the natural places to incorporate human assistance are the semantic routines since those are where decisions are made as to how to interpret the constituents as the building blocks of a sentence. At this (constituent) level the system both avoids frequent and unintelligent questioning of the user (as is the case with the interactive disambiguation approach at the lexical level) and does not suffer from late recovery (as is the case with the paraphrase strategy).

As an example, let us consider the following sentence.

(7) The talented conductor dated a young star.

The system does not ask for the user's help the first time it sees the word "conductor", but waits till the analysis of the NP with "conductor" as its head is complete just before the tree representation is built. At this point it asks:

Does *"conductor" here mean "an official on a bus or train or tram who collects fares"? (y/n)*

Here the system selected one of the "human being" senses of the word "conductor" as a result of carrying out semantic matches of the modifier "talented" and the head noun (a tram conductor can certainly be talented). The order of selecting the "bus conductor" sense vs. the "orchestra conductor" one is arbitrary, though with more domain information and statistical consideration preference can be given to one over the other. If the user answers "no" the system backtracks to find another "human being" interpretation of the word:

Does "conductor" here mean "a person who conducts an orchestra" ? (y/n)

The answer from the user at this point is likely to be "yes". If, however, the user insists on "no" the system will relax the semantic constraints (Huang, 1988) and accepts the "a substance that ..." sense of the word "conductor", treating "talented" as used metaphorically.

Suppose the user has chosen the "orchestra conductor" sense. The system continues the parsing to process the finite verb "dated" and upon the completion of subject-verb match for deciding a proper sense for the verb which suits the selected sense for "conductor", it asks the user to confirm its choice:

Does "date" here mean "to go out on dates with"? (y/n)

If the user answers "yes" the systems carries on to find an interpretation for the object NP, using the chosen sense of "date" to help disambiguate "star":

Does "star" here mean "a celebrity"? (y/n)

Similar to the confirmations described above, when prepositional phrases are processed confirmation is carried out after the semantic matches have resolved the attachment ambiguity of the PPs. For example, for the sentence

(8) John met the girl he worked with at a dance.

the system asks the user:

Does "at a dance" modify "met"? (y/n)

after it executes the PP-attachment resolution procedures (for more details see (Wilks et al., 1985)). The confirmation is likely to be positive, based on the strong preference of "meet" for a location case. But if the user intends to say something different ("John met the girl, with whom he had worked at a dance"), s/he might answer "no", and then the system will present (after backtracking) to the user another interpretation:

Does "at a dance" modify "worked"? (y/n)

Confirmations would also be needed at points where semantic matches are carried out to resolve the coordinate conjunction ambiguity, such as contained in the phrase "the man and the woman with an umbrella" ("[the man] and [the woman with an umbrella]") or ("[the man and the woman] with an umbrella") [2] (Huang, 1983).

An important factor in designing the interactive system is the number of questions asked of the user. The less questions asked, the more productive the system will be, and the less bored the user will become.

Ideally the system should be intelligent enough not to ask for confirmation about the word "dated" when processing

(9) ANU dated the world's oldest rock.

or

(10) Ann dated the department's oldest professor.

Whereas if the input sentence is

[2] At this stage only confirmations involving NPs, NP-Verb pairs and Verb-Np pairs are implemented, although all semantic matches are actually carried out (Huang, 1987).

(11) Ann dated the town's oldest coach.

the user may not feel it unreasonable if the system asks him/her to confirm the disambiguation of "dated", "oldest", and "coach". What makes difference here is the so called "genuine" ambiguity (sentences for which more than one interpretation is meaningful): sentences (9) and (10) should have only one valid interpretation each according to our common knowledge even though when standing in isolation the word "dated" is ambiguous, whereas sentence (11) might have two meaningful readings.

If user confirmation is required only for sentences containing genuine ambiguity, the system would raise much less questions without endangering the quality of the translation. But to decide when a word or a structure is "genuinely" ambiguous may involve more computing resources than is worthwhile employing. First of all it has to exhaust the whole search space, both syntactic and semantic, to find out whether more than one meaningful interpretation exists, despite the fact that often the first such interpretation might well be the only one. Secondly it can be very tricky to draw a clear line between "meaningful" and "meaningless" interpretations.

For example, assume that the verb "bear" has two distinct senses in the system's vocabulary, "give birth to" and "tolerate". Then the sentence

(12) John cannot bear children.

might be judged as not "genuinely" ambiguous because common sense dictates that since John is a male (based on world knowledge of names), it is impossible for him to give birth to children, and therefore "bear" can have only the other interpretation in the context. But then if you tell your granddaughter "John cannot bear children because he is a male", you are using "bear" in the "give birth to" sense, although "tolerate" is still an acceptable interpretation.

For this reason at the current stage we have not attempted at singling out genuinely ambiguous sentences but instead employ certain domain information and statistical consideration in arranging the system's lexicon, so that for sentence (7) ("The talented conductor dated a young star.") it may well be the case that the user need only answer "yes" once to each of the questions the system asks. The processing will be much prolonged if the intended meaning of the sentence deviates significantly from the "normal" interpretations (those likely to be reached first by the system), as is the case when sentence (7) is meant to say "the talented tram conductor ascertained the age of a new celestial object". Then of course there might be sentences containing metaphors, or simply meaningless structures. In such cases the user will have to answer the system's questions several times over the same word.

4 Future Work

4.1 Learning and Remembering

One of the answers to reducing the number of questions raised by the system would be to equip the system with facilities for learning, using the user's confirmation information as an active source of knowledge and updating the system's memory accordingly. A personalized data base may be established for each individual user so that most frequently used word senses or structures get considered first.

4.2 Singling Out Genuinely Ambiguous Sentences

Another answer may lie in differentiating between sentences that are "genuinely" ambiguous and those that are not, as is discussed in the last section. With richer knowledge bases and more powerful inference engines it may become practical to first recognize genuinely ambiguous sentences and then to assign scores to different interpretations of such sentences, and finally present them to the user for confirmation in the order of their scores. Sentences not containing genuine ambiguity will be processed without user assistance.

4.3 Unknown Words

Since the user of our MT system does not know the target language, how can s/he specify the target language equivalent of an unknown word, even though s/he might be able to provide all other information in the source language about the word (its part of speech, conjugation/inflexion, meaning, etc.)? There seem to be three possible solutions:

- Using synonyms which are known to the system. This is similar to the pre-editing approach, where the user must be familiar with the system's translation capacity.
- Entering all unknown words into a list, and periodically updating the dictionary by bilingual experts, using the list. This is what Melby calls "para-processing" (Melby, 1987) (p.146).
- Copying the unknown words into the translation, hoping it would not hinder the readability of the translation. This might be the best and simplest solution when the unknown words are proper nouns.

Acknowledgement

The permission of the Executive General Manager, Telecom Australia Research Laboratories, to publish this paper is acknowledged. I would also like to thank Li Chen for discussions, and the reviewers for comments. Any errors remain mine.

References

Carbonell, J. and Tomita, M. (1987). Knowledge-based machine translation, the CMU approach. *Machine Translation: Theoretical and Methodological Issues,* pages 68-89.

Church, K. and Patil, R. (1982). Coping with syntactic ambiguity or how to put the block in the box on the table. *Technical Report MIT/LCS/TM-216.*

Huang, X.-M. (1983). Dealing with conjunctions in a machine translation environment. *Proceedings of the 1st Meeting of the European Chapter of the Association for Computational Linguistics,* pages 81-85.

Huang, X.-M. (1987). XTRA: The design and implementation of a fully automatic machine translation system. *Unpublished Ph.D. thesis.* Also as Memorandum in Computer and Cognitive Science, MCCS-88-121, Computing Research Lab, New Mexico State University, Las Cruces.

Huang, X.-M. (1988). Semantic analysis in XTRA, an English - Chinese machine translation system. *Computers and Translation,* 3(2):101-120.

Melby, A. (1987). On human-machine interaction in translation. In Nirenburg, S., editor, *Machine Translation: Theoretical and Methodological Issues,* pages 145-154. Cambridge: Cambridge University Press.

Wilks, Y., Huang, X.-M., and Fass, D. (1985). Syntax, preference and right attachment. *Proceedings of IJCAI85 (1985 International Joint Conf. on Artificial Intelligence),* pages 779-784.

Wood, M. M. and Chandler, B. (1988). Machine translation for monolinguals. *Proceedings of COLING88,* pages 760-763.