RESOLVING AMBIGUITIES IN SYSTRAN

Katerina Antonopoulou

ABSTRACT

When translators ask a terminologist for the meaning of a term, they will usually provide the subject matter and context of the word requiring translation. A term might have different translations in different domains, or even several translations within the same domain, depending on context, so this information is essential to the terminologist in order to decide what the most appropriate translation would be.

However, when compiling a dictionary to be used by a machine translation system, we are faced with a problem. We know neither the exact domain of the text nor the particular context of the term in advance. Thus it is not always easy to decide which meaning to include in the system's dictionaries.

Solutions include the coding of expressions specifying the exact context of the term and the creation of specialised dictionaries which the user would be required to request when sending his text for translation. However, such remedies can cause additional problems when, for example, the text covers more than one subject matter. In this paper, we will examine how one particular system, the one used by the European Commission, deals with the problem of multiple translations.

0 INTRODUCTION

When translators ask a terminologist for the meaning of a term, they will usually provide the subject matter and context of the word requiring translation. A term might have different translations in different domains, or even several translations within the same domain, depending on context, so this information is essential to the terminologist in order to decide which would be the most appropriate translation. The fact that the translation of a word may change with context is one of the most frustrating problems for translators working on machine translation.

While a human has at his/her disposal the subject field and context of the term, when compiling a dictionary to be used by a machine translation system, the coders are faced with a problem. They know neither the exact domain of the text nor the particular context of the term in advance. Thus, since they can neither foresee in what context a word will appear nor to code all possibilities, it is not always easy to decide which meaning to include in the system dictionaries. Hence the problem is attacked on several fronts - one can use the most generally accepted or the most frequent meaning as the default, and try to find other solutions for the exceptions.

In this paper we are going to focus on the way one particular system solves the multiple translation problem. The system we will concentrate on is Systran, the MT system used by the European Commission. The Systran MT system was developed by Peter Toma in the late 1950s and the European Commission acquired certain rights for it in 1976. Since then the CEC has added many language pair combinations, the total now being 17 pairs, which provide translation from English into French, Italian, German, Dutch, Spanish, Portuguese and Greek; from French into English, German, Dutch, Italian and Spanish; from German into English and French; from Spanish into English and French; and from Greek into French. The development of English-Greek began in 1988 and that of Greek-French (which is still at a developmental stage) in 1993. Both pairs are co-funded by the Commission and the Greek government.

1 DESCRIPTION OF THE SYSTEM

Before going into detail about how the Commission's MT system resolves the problem of multiple translations, we would like to provide some information about the structure of its dictionaries and the translation process followed.

The system has two main types of dictionaries:

- Stem dictionary: This essentially contains entries coded independently of their linguistic context. Stem dictionaries were initially bilingual but when the number of language pairs increased, a mono-source/multi-target approach was preferred. The Stem dictionary of each source language contains basic syntactic information on a word, information required for homograph resolution, syntactic and semantic codes and, finally, its translation in all target languages of the system. Syntactic codes are used to specify the part of speech a word belongs to and help in the correct analysis of the source text. Semantic codes can be applied on any part of speech but are mostly used on nouns. They give a rough categorisation of the word or expression and are also mostly used during analysis. The ones most widely used are HUMANS and GROUPS. Other examples of semantic codes are CITIES, MONTHS, ANIMAL, PROPTY (property), COLOUR, etc.
- IDLS (idiom/limited semantics) dictionary: This contains expressions which determine the translation of a single word or an expression according to the context it appears in. There are five types of entries which can appear in the IDLS dictionary. The most

frequent ones are *idiom replaces*, *SLS (Straight Limited Semantics)* and *CLS (Conditional Limited Semantics). Idiom replaces* are usually prepositional, conjunctive or adverbial phrases such as *with respect to*, *in order that* and *in the long run*. Once an expression is coded as an idiom replace, it is treated as one word which has the advantage that the string is never misanalysed as having some other syntactic function and its meaning in the target language is easier to find. The SLS feature assists the coding of noun phrases (often technical terms) and facilitates the resolution of homographs by allowing developers to attach a different translation to some words when they appear in a certain string. For example, an SLS rule would allow us to translate the expression *power station* as *centrale*. CLS rules are another very powerful feature for covering exceptions to the default meaning provided in the Stem dictionary. Their flexibility allows the developer to specify in exactly what context the rule should apply, to examine the rest of the sentence for certain conditions and so on.

2 TRANSLATION PROCESS

In order to explain a bit more about how the dictionaries are used we will very briefly describe the translation process the system goes through, concentrating mainly on when the dictionaries are accessed and when rules are applied. It should be borne in mind that the source text is translated sentence-by-sentence and that the translation procedure has three main stages: analysis, transfer and synthesis.

In a nutshell, when a text is received by the system, it is first passed through a set of pre-processing routines which serve to identify sentence boundaries and separate the text from any formatting information, thus turning it into a suitable format. The system then proceeds to access the Stem dictionary, compare every word in the text with the entries existing in the Stem dictionary and retrieve as much information as possible about the source-language terms. This process is called *main dictionary look-up*. The information acquired will be subsequently used in order to establish relations between words at the analysis stage.

Next, the analysis stage starts. During analysis the system is above all concerned with obtaining enough data about the source language and does not yet actually translate into the target language. The first step of the analysis phase is an attempt to resolve any homograph problems. Several routines are available for this purpose.

During the next stage of analysis, the system tries to establish the main syntactic relationships between words. To do this the system uses any information acquired by the main dictionary look-up and passes each sentence through many programs and subroutines, each dealing with a particular relationship (subject-predicate, co-ordinated lexical items, deep structure relationships and so on). Several passes are required for this phase. The expression dictionary is accessed several times throughout the analysis, the system looking for matches between terms in the source text and its own entries.

In contrast with the analysis stage, the transfer stage is bilingual and therefore different for every language pair. On a lexical level, the uniterms and expressions identified previously are now translated, but not in their final form. Source-language prepositions are also translated according to the context they appear in. Finally, structural transfer is achieved by means of small programs called *lexical routines.* These refer either to particular source-language words or to particular structures.

The final stage of the translation process is synthesis. By now all source language words have received their equivalent in the target language. *Synthesis* deals with morphological inflection, the insertion of articles, prepositions, particles and auxiliary verbs and the rearrangement of words in an order permissible by the syntax of the target language. These goals are achieved by synthesis programs.

After the text has been translated, it is returned to the original page format by post-processing routines.

3 WAYS OF CODING MULTIPLE TRANSLATIONS IN SYSTRAN

When coding dictionary entries in the Commission's MT system, the basic rule is that the most general meaning should be coded as the default, while the exceptions should be handled by some special mechanism. By "most general" we mean the one that occurs most frequently in the texts which are usually translated, and which is applicable in as many contexts as possible. The most general meaning is found by checking frequency listings of words in texts which have already been translated by human translators. The method most frequently used is called KWIC (KeyWord In Context) and it allows the coder to search a huge database containing texts which have been translated in the past for the word/expression in question. A list of the phrases in which the word/expression has appeared is returned and then the coder can decide which meaning to code as the the default and which meanings to code as exceptions. The list of phrases also *gives the coder an idea of the kind of contexts* in which exceptions usually occur.

3.1 TOPICAL GLOSSARIES

One way of handling exceptions to the default meaning is through the use of Topical Glossaries (TGs). TGs are glossaries dealing with specific fields or subject matters (e.g. energy, finance, agriculture). They can be considered as sub-sections of the Stem and IDLS dictionaries and allow the developers to code, not only the most general meaning of a word or expression, but also a more specialised translation (which is stored in the relevant TG). It is then up to the user to judge which, if any, of the specialised glossaries suits his document best and to request that it be used when sending his/her text for translation. Users in the Commission are currently allowed to request up to three TGs

An example of the usage of TGs can be seen if we consider the French word *coeur*. This would be coded as *heart* in the English default dictionary. However, it is reasonable to assume that if a document on nuclear energy is being translated, *coeur* should be translated as *core* instead of *heart*. What the coder can do is to put this more specialised translation in the "energy" glossary. This way a user sending a document on nuclear energy for translation would improve the translation and minimise the time spent on post-editing or correction by asking for the "energy" glossary to be used.

The main advantage of using TGs in order to distinguish between different translations of a word is that it is very effective when we are dealing with very specific subject fields. They are also remarkably easy to code. All coders need to do is find the different translations a word has in different domains and code each meaning in the relevant TG.

However, this method also has quite a few disadvantages. One lies in the fact that the users have to decide whether or not to request a TG when sending off their translation, and if so which one(s). This is not always straightforward since a text might contain several subject fields. Users are advised to experiment with various combinations of TGs until they get a satisfactory result.

A more serious disadvantage of TGs is the fact that once you specify a Topical Glossary to be used and a word is found in your text which has a meaning in that Topical Glossary this meaning will *always* be used in the text being translated. For instance, if the default translation of the French *poste* in the STEM dictionary is *post* and in the Informatics TG it is *station* (e.g. *poste de travail --> work station*) then, once the Informatics TG is chosen, *poste* will be translated as *station* throughout the whole text. This may or may not be desirable. Coding something in a TG means that the flexibility of having it translated differently *within* a text is lost. By means of a special code, the Commission's MT system provides users with the opportunity of changing the Topical Glossaries used within a text whenever the subject field changes. However, these changes obviously have to be specified in the text itself and cannot be indicated through the user interface, thus making the request procedure more time-consuming.

Similar to TGs are User Codes. A user can provide his own preferred translations to terms in a personal dictionary which, when requested, will take precedence over TGs.

3.2 IDLS dictionary

Another method of entering alternative translations which avoids this problem, but is also not very flexible, is to code expression strings (SLSs) with their translations in the IDLS dictionary. To use one of the above examples, the programmer could code *poste de travail* as one string. It would then always be translated as *work station* in English without affecting the translation of *poste* on its own. In this case the context decides on the translation and thus a TG does not need to be specified.

One potential disadvantage of this solution is that something which forms part of an expression might at some later point in the text be referred to on its own. Another problem is the fact that the words in SLS rules have to follow in their specified order if the expression is to be picked up. So for instance, let us examine the French expression *Honorable Parlementaire*. With an SLS rule this could be picked out and translated in English as *Honorable Member*. But if in our text we have *Honorable et Noble Parlementaire* the SLS rule will not match this phrase since the two words are not adjacent.

3.3 CLS rules

As we have already seen, through the use of CLS rules, the system provides one more way for defining translations of words or expressions which are different from the default meaning in the Stem dictionary. CLS rules allow the coder to describe the exact context a word/expression should appear in. This context can be either syntactic or semantic and the rules may be as simple or as complex as required. Thus, if the default meaning or *work* is *fonctionner*, a CLS rule could be used to obtain the translation *travailler* when the subject of *work* is human. Or, to use the example given above, a rule stating that "If *Honorable* qualifies *Parlementaire* (even if it does not immediately precede it) then translate *Parlementaire* as *Member*. Some of the more ambiguous terms may be covered by dozens or even hundreds of CLS entries.

We can see from the above examples that one or more conditions can be used and that the conditions required are specified in terms of syntactic (e.g. "if *x* qualifies *y*") and semantic (e.g. "if the subject is HUMAN") relationships between words in the sentence. However, you can also scan the sentence for certain words or look for certain types of words (e.g. "if *x* precedes *y*").

Furthermore, CLS rules provide the coder with an additional facility which makes them even more flexible. One of the disadvantages of coding an exception to the default translation as an SLS rule was the fact that a word which forms part of an expression might be encountered on its own later in the text. The same thing can happen when the translation of a word is specified as a CLS rule. In both cases a *special meaning code* called KEEPMN (keep meaning) can help. The function of special meaning codes is to provide additional information required for the relationships between words in the target language. In particular, KEEPMN ensures that the meaning selected will continue to be used as the translation of a given term throughout the text or until such time as a contradictory rule is provided. So, to return to the previous example, if to the rule *honorable parlementaire --> honorable member* a KPEEMN special code is added, and later we get *parlementaire* on its own, the translation will still be *member* as opposed to *MP*.

Finally, in a few cases, disambiguation is done through the use of lexical routines. As we have already seen, lexical routines are executed at transfer stage and are written for words or groups of words requiring special treatment which cannot be easily catered for by coding. One example is the French word *dont*. A lexical routine chooses between the translations *which*, *whose*, and *among which* and in some cases rearranges the words to suit the English word order. Similarly, a lexical routine for *ensemble* uses the semantic context to decide whether the translation should be *a number of* (in the case of *un ensemble de* + plural noun), *throughout* (in the case of *dans l'ensemble de GEOLOC*, where *GEOLOC* is a semantic code indicating a geographical location) and so on.

Finally, something we have not mentioned yet is the fact that a combination of the above methods is of course allowed and indeed in some cases necessary. For instance, an SLS entry or a CLS rule might have as one of its conditions that a particular TG is required.

ACKNOWLEDGMENTS

I am grateful to the Translation Service of the European Commission and in particular AGL/4 ("Development of Multilingual Tools" unit) for the opportunity to participate in its training scheme (Oct. '96 - Feb. '97). During this time, I familiarised myself with the workings of the Commission's MT system. I also want to thank Mr Cameron Ross, Mrs Rosemarie Sauer-Stipperger and Mrs Francine Braun-Chen for the information they provided and their invaluable comments.

REFERENCES

(1) Petrits Angeliki, *"The Commission's Machine Translation System"*, DGXIII - Translation Service, 1997 (available to the public on request).

(2) Petrits Angeliki, Lembessi Penelope, Roussou Sophia, *"The Commission's Systran English - Greek Machine Translation System"*, Luxembourg (Draft paper).

(3) Pigott Ian, *"Systran development at the EC Commission"*, Commission of the European Communities, Luxembourg (available to the public on request).

(4) *"The Systran Machine Translation System: Practical Aspects"*, Commission of the European Communities, date last revised 1994 (available to the public on request).

3I∴MΨΓΥΙ •Υ∴αΥΩΞΩεΣΩ]

+MPΘIΨEΣO 19

•ΠΦΥΙ 152-37