[TMI 99: *Proceedings of 8th International Conference on Theoretical and Methodological Issues in Machine Translation*, August 1999, Chester, UK]

A Confidence Index for Machine Translation

Arendse Bernth
IBM T.J. Watson Research Center
P.O. Box 704,
Yorktown Heights, NY 10598, USA
arendse@watson.ibm.com

Abstract

We argue that it is useful for a machine translation system to be able to provide the user with an estimate of the translation quality for each sentence. This makes it possible for bad translations to be filtered out before post-editing, to be highlighted by the user interface, or to cause an interactive system to ask for a rephrasing. A system providing such an estimate is described, and examples from its practical application to an MT system are given.

1 Introduction

High-quality machine translation (MT) is highly desirable in today's global community and is the goal of many computational systems. Unfortunately, natural languages are very complex, and this poses great challenges for any MT system. No MT system today is able to produce perfect translations of arbitrary text. For any given system, translations range from perfect to unintelligible.

In order to guarantee high quality, some systems require that the source text be constrained more or less severely. Not only does this place a considerable burden on the author, but it also means that documents that are not specially prepared cannot be handled. Some of these combinations of *Controlled Language Checker* and MT system require strict conformance to the Controlled Language, e.g. the KANT system (Mitamura & Nyberg 1995; Nyberg & Mitamura 1996; Hayes et al. 1996; Kamprath et al. 1998), and the CASL system (Means & Godden 1996). Other systems, e.g. EasyEnglish (Bernth 1997; Bernth 1998a; Bernth 1998b), help the writer prepare a document for machine translation by pointing out hard-to-translate constructions without enforcing strict control. But systems like EasyEnglish neither guarantee a perfect translation nor give an indication of how well the source text would translate.

Controlling the source language certainly helps the quality of translation; however, for many applications, e.g. on-the-fly translation of random Web pages (Bernth & McCord 1998), it is not possible to constrain the source text. This means that the user will be subjected to bad translations as well as to good translations, without any indication of how good the translation of any given segment may be.

Bad translations cause a high degree of frustration for the user, because the user has no way of avoiding the garbled-up nonsense that is produced occasionally by even the best MT systems. It appears to be an unfortunate fact of life that MT gets a bad reputation from translations that are bad rather than a good reputation from all the translations that actually are useful. If the user could know that the translation was likely to be bad, the user would have the choice not to look at it. In other words, if the MT system could provide the user with an indication of the probable accuracy of the translation, it would be up to the user to decide to look at it or not.

Previous systems such as the Logos Translatability Index (TI) assign a measure of the translatability of a *complete* document by the LOGOS system. The Logos Translatability Index was not expected to "provide sentence-specific information with any degree of reliability. The TI applies to the corpus or document as a whole but is not useful in pinpointing problem sentences." (Gdaniec 1994).

In this paper we describe the Translation Confidence Index (TCI), which is designed to provide the user with a measure of the MT system's own confidence in its translation, segment for segment. The TCI engine associates a number, the TCI, between 0 and 100 (inclusive) to each segment. A TCI of 100 expresses perfect confidence in the translation, whereas a TCI of 0 expresses zero confidence.

The TCI engine, which works with a transfer-based MT system, is fully implemented, and it has been integrated with the LMT machine translation system (McCord 1985, 1989a, 1989b, 1989c; Bernth & McCord 1991; McCord & Bernth 1998; Bernth & McCord 1998; Gdaniec 1998).

In Section 2 we look at how the TCI can be used. Section 3 describes the basic ideas of the TCI. In Section 4 we describe the role of choices in the translation process for calculating the TCI. Section 5 gives an overview of the types of problems that contribute to the TCI. Section 6 describes tuning the TCI for a specific language pair. In Section 7 we look at the language pair profile, and in Section 8 we show some practical results for LMT English-German.

2 Modes of Use for the TCI

In this section we describe different ways the TCI can be used in different translation contexts.

The TCI can be used by the MT system's interface in various ways. In one scenario, the user will have decided on a threshold based on personal preferences, and the system simply will not show translations where the TCI falls below this threshold. This scenario is particularly useful in the context of professional translators using a translation workbench. A professional translator is apt to become both annoyed and insulted by bad translations. Furthermore it may also be harder to post-edit a bad translation than to start from scratch. In any case it makes sense to protect the professional translator from bad translations produced by the MT system.

In another scenario, the interface could pass on all of the translations, regardless of the TCI, but could indicate the TCI either by giving the specific number or by marking bad translations in red for example. This would be useful for the casual user, who does not know the source language. This type of user would then have the information to take the bad translations with a grain of salt.

In the context of interactive translation, the TCI can be used to provide immediate feedback so that users can rephrase the input if the TCI is below the given threshold. This can be used for both text-to-text translation, where the user types the text of the source document and has it translated on-the-fly, and for speech-to-speech translation, where the system may ask the user to repeat or rephrase the input.

3 The TCI: A Measure of Complexity

The overall idea behind the design and implementation of the TCI is to measure the complexity of the translation process. As the complexity increases, the confidence decreases.

The complexity depends on three major factors: The choices coded in the MT system that are encountered during the various steps of the translation process, the complexity of the source text, and how each of these two factors affects the translation for a given language pair. In the following sections we shall address these points.

Generally, the TCI for a segment will be computed by assigning penalties to different kinds of complexities that could create potential problems. Penalties are integers, with a larger number representing a worse problem. The penalties for a segment S are added together to obtain the total penalty P for S. P should thus be viewed as the *accumulation* of potential problems, small or large, rather than as an indication of one specific large problem. Most potential problems would not be big enough to lower the TCI substantially, if they occur in isolation, but taken together, they may be a good indication that this is a problematic segment. Ambiguity in part of speech between nouns and verbs, for example, may not be a problem in itself, but combined with a short segment length (as well as other factors like the presence or absence of determiners) it can signal a degree of uncertainty that should be penalized.

Given the total penalty P for the segment S, the TCI of S is then computed as

```
TCI(S) = max(100 - P, 0).
```

In other words, we subtract the total penalty from 100 to get the TCI, but we do not bother with scores that are less than zero, because they will certainly represent unusable translations.

Obviously, the impact on the translation quality of any given type of problem will be specific to the language pair you are translating between. For this reason, the TCI makes use of *language pair profiles* (described in Section 7) where penalties can be set for each language pair.

4 Choices in the Translation Process

In a computational system where heuristic choices are made, each such choice introduces a potential for a mistake. Such mistakes can be of one of the following types:

1. The choices include the correct choice, but an incorrect choice is made.
2. The choices do not include the correct choice, and an incorrect choice is inevitable. This typically reflects either “direct” lack of information (e.g. missing lexical information or missing grammar rules) or “indirect” lack of information caused by a wrong choice earlier in the process (e.g. some analysis was incorrectly pruned away).

The second type of mistake is particularly difficult to handle, except indirectly as it may be reflected in mistakes of the first type; and we shall restrict our treatment to mistakes of the first type.

No attempt is made to evaluate the choice made by the MT system. Rather, the number of potential choices at any given choice point contributes to the penalty (the more choices there are, the higher the probability of not making the correct choice). Also the *type* of choice affects the score. This is expressed in the penalties stated in the language pair profile.

5 Monitoring the Translation Process

A transfer system consists of three distinct phases that all may introduce mistakes: Source analysis, transfer, and target generation. Hooks into the LMT system at crucial points during these phases call routines that look for the various types of problems, assign penalties, and calculate intermediate values for the TCI. An option is to abandon the translation process for a given segment if the TCI falls below a specified threshold. This saves processing time in the scenario where the user is not interested at all in translations whose TCI falls below the threshold.

In this section we give an overview of the types of problems that we look for during each phase of the translation process.

5.1 Source Analysis

The earlier in the translation process a problem occurs, the more impact it is likely to have, because the problem is likely to propagate and affect all subsequent steps. Thus problems in source analysis are given relatively higher penalties in the TCI. Due to the importance of the quality of the source analysis in producing a good translation, we are devoting a separate paper to this issue (Bernth & McCord 1999); here we give a brief overview of the major issues.

The most evident problem is a sentence that cannot be given a complete parse. The parser used with LMT, the ESG parser (McCord 1980, 1990, 1993), produces in these cases a pieced-together version of a failed parse. But other, more subtle, things may go wrong during source analysis, e.g. segmentation, lexical choices, syntactic analysis, and various ambiguous or otherwise hard-to-parse constructions.

One of the most serious segmentation problems is caused by footnotes, because people are not consistent in the way they use footnotes; the role of footnotes in the sentence is far from unambiguous. They may be separate segments or actual parts of the sentence.

Parts of speech can be very ambiguous in short segments of one to four words. In longer segments, the context often disambiguates the part of speech.

Certain constructions or words are known to increase the likelihood of a bad parse and/or translation. The most obvious case is of words that are not found in the lexicon. However, many other characteristics of the source text may cause problems. Consider e.g. the following non-exhaustive list: Occurrences of coordination may be hard to parse correctly. Missing subjects in the sentence may make it problematic to get correct subject-verb agreement in the target. All structural ambiguities, e.g. double passives, nonfinite verbs, and prepositional phrases entail a risk of incorrect attachment. Time references like *next year,* which may may be either adverbs or nouns, may affect the parse.

5.2 Transfer

Problems in transfer may stem from wrong source analysis as well as from inadequate lexical information (this includes missing semantic types and domain specifications) and wrong application (or non-application) of transformations. In addition, mixed domains in the document can be a problem.

Problems during transfer fall into two different categories:

1. Lexical Transfer.
2. Restructuring Transfer.

For lexical transfer, the most glaring problem is lack of transfer for a given source word. This is a subcategory of a more general problem, viz. mistakes in the transfer lexicon. The more complex the entry, the more likely it is that the desired transfer is actually in the lexicon, but also the greater the chance that a mistake was made in creating the entry. Informal empirical studies show the latter to be a factor that should not be discounted, so we count the number of transfer elements.

On the other hand, if the transfer was found in a specialized lexicon, we increase the confidence. This is done by giving this “problem” a negative penalty, which is equivalent to a reward.

During restructuring transfer, the problems may arise from application of certain transformations that are known “troublemakers”. The language pair profile allows the transformation writer to specify the names of these transformations and assign suitable penalties. Some transformations may also be known as real "life savers", and they can be given negative penalties in the profile. Another source of problems during restructuring transfer is transformations that apply partially; i.e. the transformations manage to make some changes in the tree structure, but they fail at a later point and do not succeed completely. This reflects mistakes in the transformations and is penalized accordingly.

5.3 Target Morphological Generation

Problems in this area are very insignificant, since target morphology in itself is a rather well-defined and limited area. Any problems axe likely to have been propagated from previous steps, particularly steps that assign features to words.

However, in highly inflected parts of speech, a wrong feature stemming from an earlier step is likely to cause a certain amount of bad inflection, so this needs to be taken into consideration. The morphology writer has the possibility of specifying a small number of highly inflected parts of speech in the language pair profile, and whenever one of these parts of speech is encountered, the specified penalty will apply.

6 Production Mode and Tuning Mode

The TCI engine is part of the translation shell. In addition to using the engine, it is necessary to specify the penalties in the language pair profile.

The purpose of *production mode* is simply to use the TCI to control the output of the MT system. However, the purpose of *tuning mode* is to tune the system to give the most accurate indication of the translation quality by setting the optimal penalties in the profile. In tuning mode, the system creates an output file that shows the TCI for each segment as well as all the individual penalties that the TCI is made up of. This *analysis file* is interfaced to a text editor, and the language specialist can experiment with the various penalties.

The penalties assigned to the various types of potential problems reflect the current state of the MT system; hence, the TCI must be tuned every now and then as the MT system is improved, even though of course the impact of certain problems like incomplete parses is likely to remain constant over time. The process of tuning in itself provides valuable feedback to the developers about the weaknesses of the MT system.

7 The Language Pair Profile

The TCI language pair profile allows the user to set the penalties for each problem type. The problem types are identified by a code, e.g. *nonfinite* for ambiguous nonfinite constructions. In addition, it is possible to specify names of transformations and highly inflected parts of speech.

The profile is a simple ASCII file with lines of the following form:

code1 = valuel
code2 = value2

where each problem type code*i* is assigned a penalty value*j*.

This profile is read in by the system and used in the calculation of the TCI for translations for a specific language pair. Every time a specific type of problem or choice is encountered, the relevant penalty is applied either as-is, or with a weight, depending on the specific problem and its context.

8 Practical Results

We have successfully integrated the TCI with LMT and tuned it for English-to-German translation. Of course the threshold below which a translation is considered unuseful is a matter of context and personal taste. But we have found with our current profile that there is a distinct separation into good and bad translations around a TCI of 65-70. Assuming a threshold of 70, the TCI divides the output of LMT English-German into reasonable translations and bad translations with a precision that turned out to be 72%. We expect to be able to improve this precision some by further tuning.

Here are some examples of output from LMT English-German, where the TCI is stated in the beginning of the translation. Let us first look at some examples of bad translations, indicated by a low TCI, as in (1).

(1) *Instead of selling platforms, IBM can now focus on selling "best-fit" server solutions into its target, corporate-wide solution markets.* ⇒
34-85: Anstatt Plattformen zu verkaufen, kann die IBM darauf jetzt zielen, "am besten passende" Serverlösungen in sein Ziel zu verkaufen, unternehmensweite Lösung vermarktet.

The play being over, we went home. ⇒
16.80: Der spielen Sie Wesen, wir gingen nach Hause.

In the first example, the parse of *corporate-wide solution markets* is wrong; *corporate-wide solution* is taken as a noun and hence the subject, while *markets* is taken as a finite verb. This makes total nonsense of the translation.

In the second example, the parse is incomplete, with *being* as a noun and *play* as a verb. This is reflected in the translation.

Some examples of better translations, indicated by a higher TCI, are given in (2).

(2) *These include seven high-growth areas, which it clusters into the following three broad categories:* ⇒
90.36: Diese umfassen sieben Hochwachstumsbereiche, die es in die folgenden drei breiten Kategorien bündelt:

Write the information in the space provided on page &spotref. in the front of this book. ⇒
88.81: Schreiben Sie die Informationen in die auf Seite &spotref. am Anfang dieses Buchs vorgesehene Stelle.

9 Conclusion

We have argued that – given the state of the art of MT – it is very useful for any MT system to supply the user with an indication of the quality of translation output on a segment basis. Such a measure can be based on the general idea of monitoring choice points and noticing problematic constructions in the source text, and relating the impact of these to the language pair in a given translation process. We have described a specific implementation of this general idea for the LMT system and given examples that illustrate some practical results.

References

Bernth, A.: 1997, 'EasyEnglish: A Tool for Improving Document Quality', in *Proceedings of the Fifth Conference on Applied Natural Language Processing,* Association for Computational Linguistics, pp. 159-165.

Bernth, A.: 1998a, 'EasyEnglish: Preprocessing for MT', in *Proceedings of the Second International Workshop On Controlled Language Applications,* Carnegie-Mellon University, Pittsburgh, pp. 30-41.

Bernth, A.: 1998b, 'EasyEnglish: Addressing Structural Ambiguity', in *Proceedings of AMTA-98,* Association for Machine Translation in the Americas, pp. 164-173.

Bernth, A. & M. C. McCord: 1991, 'LMT for Danish-English Machine Translation', in Brown, C. G. & G. Koch, eds: 1991, *Natural Language Understanding and Logic Programming III.* North-Holland, pp. 179-194

Bernth, A. & M. C. McCord: 1998, 'LMT at Tivoli Gardens', in *Proceedings of the 11th Nordic Conference on Computational Linguistics,* Copenhagen, pp. 4-12.

Bernth, A. & M. C. McCord: 1999, 'The Translation Confidence Index and Source Analysis', in preparation.

Gdaniec, C.: 1994, 'The Logos Translatability Index', in *Proceedings of AMTA-94,* Association for Machine Translation in the Americas, pp. 97-105.

Gdaniec, C.: 1998, 'Lexical Choice and Syntactic Generation in a Transfer System: Transformations in the New LMT English-German System', in *Proceedings of AMTA-98,* Association for Machine Translation in the Americas, pp. 408-420.

Hayes, P., S. Maxwell & L. Schmandt: 1996, 'Controlled English Advantages for Translated and Original English Documents', in *Proceedings of The First International Workshop On Controlled Language Applications,* Katholieke Universiteit Leuven, Belgium, pp. 84-92.

Kamprath, C., E. Adolphson, T. Mitamura & E. H. Nyberg: 1998, 'Controlled Language for Multilingual Document Production: Experience with Caterpillar Technical English', in *Proceedings of the Second International Workshop On Controlled Language Applications,* Carnegie-Mellon University, Pittsburgh, pp. 51-61.

McCord, M. C.: 1980, 'Slot Grammars', in *Computational Linguistics 6,* pp. 31-43.

McCord, M. C.: 1985, 'LMT: A Prolog-Based Machine Translation System', in *Proceedings of the 1st Conference on Theoretical and Methodological Issues in Machine Translation,* Colgate University.

McCord, M.C.: 1989a, 'Design of LMT: A Prolog-based Machine Translation System', in *Computational Linguistics 15,* pp. 33-52.

McCord, M.C.:. 1989b, 'LMT', in *Proceedings of MT Summit II,* Deutsche Gesellschaft für Dokumentation, Frankfurt, pp. 94-99.

McCord, M.C.: 1989c, 'A New Version of the Machine Translation System LMT', in *Literary and Linguistic Computing 4,* pp.218-229.

McCord, M.C.: 1990, 'Slot Grammar: A System for Simpler Construction of Practical Natural Language Grammars', in R. Studer, editor, *Natural Language and Logic: International Scientific Symposium,* Lecture Notes in Computer Science, Springer Verlag, Berlin, pp. 118-145.

McCord, M.C.: 1993, 'Heuristics for Broad-Coverage Natural Language Parsing', in *Proceedings of the ARPA Human Language Technology Workshop,* Morgan-Kaufmann.

McCord, M.C & A. Bernth: 1998, 'The LMT Transformational System', in *Proceedings of AMTA-98,* Association for Machine Translation in the Americas, pp. 344-355.

Means, L. & K. Godden: 1996, 'The Controlled Automotive Service Language (CASL) Project', in *Proceedings of The First International Workshop On Controlled Language Applications,* Katholieke Universiteit Leuven, Belgium, pp. 106-114.

Mitamura, T. & E. H. Nyberg: 1995, 'Controlled English for Knowledge-Based MT: Experience with the KANT System', in *Proceedings of the 6th International Conference on Theoretical and Methodological Issues in Machine Translation.*

Nyberg, E. H. & T. Mitamura: 1996, 'Controlled Language and Knowledge-Based Machine Translation: Principles and Practice', in *Proceedings of The First International Workshop On Controlled Language Applications,* Katholieke Universiteit Leuven, Belgium, pp. 74-83.