In *Proceedings of the 8th International Conference on Theoretical and Methodological Issues in Machine Translation* (TMI 99) Chester, England, August 1999, pp 139-149.

Profiling Translation Projects:
An Essential Part of Routing Translations

Nancy L. Underwood, Bart Jongejan
Center for Sprogteknologi
Njalsgade 80
2300 København S, Denmark
nancy@cst.ku.dk, bart@cst.ku.dk

Abstract

Every day in translation departments and agencies managers must decide on the best way to carry out new translation projects. The TransRouter project is building a system to support this decision-making process and we present a brief outline of the system's architecture before describing the central role of profiling a translation project as a preliminary to calculating the most viable potential routes for carrying out a translation. Profiling a translation project involves collecting together all possibly relevant information about the project. This depends not only on information provided by the requester of the translation but also crucially on analysing the source texts themselves. To this end a number of tools for analysing properties of the source texts are also being developed and are described here.

1 Introduction

One of the most pressing tasks for the manager of a busy translation department or agency is to decide, for each incoming translation project, how best it can be carried out. Which translators should be employed on the project? Which translation aids (MT systems, translation memories, terminology management tools) are suitable? What linguistic resources (terminology, lexica, previous translations) are available and should be used? Until now such decisions on how to route a translation have often been based on the translation manager's own assessment of the texts to be translated and the available resources, both human and computational. The manager must also take other requirements such as the deadline for the completed translation, cost and the level of quality required into account. In a busy translation agency or department dealing with a large volume of translation, when time is of the essence and a number of different options are available this can become an onerous task. A translation manager may not always be confident that he or she has all the relevant information to hand. The TransRouter project (see Hammwöhner 1998 and TransRouter 1999) is building a decision support system which will not itself make the routing decision (nor indeed carry out the translation) but will speed up the decision process and increase a manager's confidence in the decisions taken.

In order to make the correct routing decisions it is essential to know as much as possible about the nature of the current translation project and potential resources as possible. Profiling is thus a central activity in the process of routing a translation. The TransRouter system is stocked with descriptions or profiles of the available translators, backend translation tools and their associated resources. In addition, for each new translation project the user will interactively build a profile of that project. It is the project profile in conjunction with the profiles of the available resources which are then used to calculate the different potential routes by which the translation can be carried out. Currently in many organisations, profiling a project tends to be done implicitly and informally by translation managers. However in order to choose the best possible route, a profile should not only register the requirements of the requester of the translation but also include as much detailed relevant information on the nature of the texts themselves as possible in order to discover their suitability for translation by a particular translator and/or translation tool. In TransRouter a number of tools are being developed which analyse various properties of the texts to be translated in order to ensure that such information is available in the decision making process.

A first prototype of the system has been built, and in this paper we first briefly describe the overall architecture of the TransRouter system and then go on to describe the process of profiling a project, and look at some of the analysis tools which feed into that process.

2 A Decision Support System for Routing Translations

The decision support system consists of a collection of different components. Figure 1 schematically indicates the interaction between the major components of TransRouter.

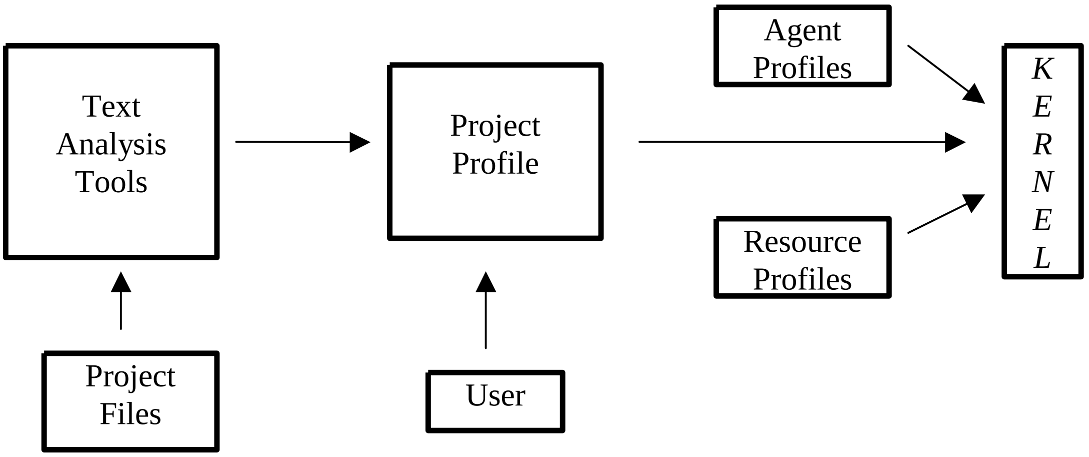

Figure 1: A decision support system for routing translations.

As mentioned above, the system is based on three different types of profile, which are fed into a *kernel* which outputs the different possible routes which can be taken to translate the project, along with the expected time it will take, the expected quality of the final product and how much it will cost. It is planned that a pre-defined set of route types will be developed which can be instantiated with the relevant agents and resources. *Agent profiles* are descriptions of the "translation agents" which are available or potentially available for the translation project. These consist of descriptions of e.g. human translators, MT systems, and translation memory (TM) systems, as well as other translators' aids such as terminology management systems, or alignment tools. In addition to agents which are currently available to the translation manager, it is also possible to have profiles of systems or tools which the translation manager may consider acquiring if the cost benefit of so doing makes financial sense. *Resource profiles* are descriptions of the linguistic resources that are available within the translation aids described in the agent profiles, such as an MT system's lexicon, the contents of a terminology database or the translation units stored in a TM system. Both agent and resource profiles are created and maintained off-line.

Project profiles, on the other hand, are created interactively by the user for each new project. To compile a project profile, two types of information are required: organisational information input by the user (such as the deadline for completion of the translation, the size of the project) and information about the nature of the texts to be translated themselves. In order to ascertain information about the nature of the texts, the system also includes a suite of text analysis tools which take the texts as input and outputs the results to the project profile. As will be seen below, these analysis tools either analyse properties of the text independently of the resources available (e.g. estimating syntactic complexity or the amount of repetitiveness in the text) or compare the texts with the resources available (e.g. measuring the proportion of the text which has already been translated and stored in a TM).

3 Profiling a Translation Project

All types of profiles take the form of tables of attribute value pairs which the kernel will access in calculating the potential routes for a given translation project. In developing project profiles we took as a starting point a pre-prototype tool (produced by LRC) designed exclusively to evaluate the suitability of TMs for localisation projects. A template project profile has now been developed to account for as many different scenarios as possible, ranging from the case where the user has access to translators working with all the different types of translators' aids on the market, to the case where only one type of translator's aid is available. So for an individual project or a particular user certain features will not be relevant. For example if the user only has access to human translators and MT systems, those features which are designed to enable the kernel to assess whether the use of a TM is feasible, will be redundant. Alternatively, if the translation project covers a completely new subject area or text type then features designed to help assess whether there is previously translated material suitable for re-use in the current project, may also be redundant. Therefore the process of profiling a translation project is structured in order to avoid either supplying redundant information or carrying out unnecessary analyses on the texts to be translated. Figure 2 (overleaf) schematically indicates this structuring of the workflow, which will be mirrored in the user interface to the system.

The first stage in profiling the project (1) concerns collecting and inputting information about the translation which will be relevant for all types of projects. One basic item of information is, of course, the language pairs that will be treated[*]. Clearly any translation tools to be used must be able to support translations between the relevant pairs of languages. This can affect the choice not only between different MT systems but also between different TM systems. Although a TM does not itself perform translations (being an archive of source texts or segments and their translations), this does not mean that they do not include information about the source and target languages they treat. In particular they must be able to support the character sets of the languages in question, not all TMs that support Western languages can also support Asian languages. Because of this problem, organisations that make heavy use of TM systems and translate between a number of different language pairs can often have more than one type of TM system in use.

[*] It might be possible to use an automatic language guesser to identify the source language in a translation project. However since the requester of a translation will tell the translation manager which source and target languages are required, using a language guesser would be redundant.

Information on scheduling and updates has a two-fold purpose. The start and finish dates for the translation project itself provide the means to calculate how much real time is available for the translation. In addition, it is useful to know whether the current translation project consists of documents (or software) which are expected to be updated in the future (for example, user manuals for products which will be released in new versions in the future, or legislation which is constantly updated), and if so when and how often. If it is expected that the translation agency will be asked to translate updated versions of the current documents in the future, then it may be worthwhile creating a translation memory during the initial translation, which can be re-used when translating the updated versions later.

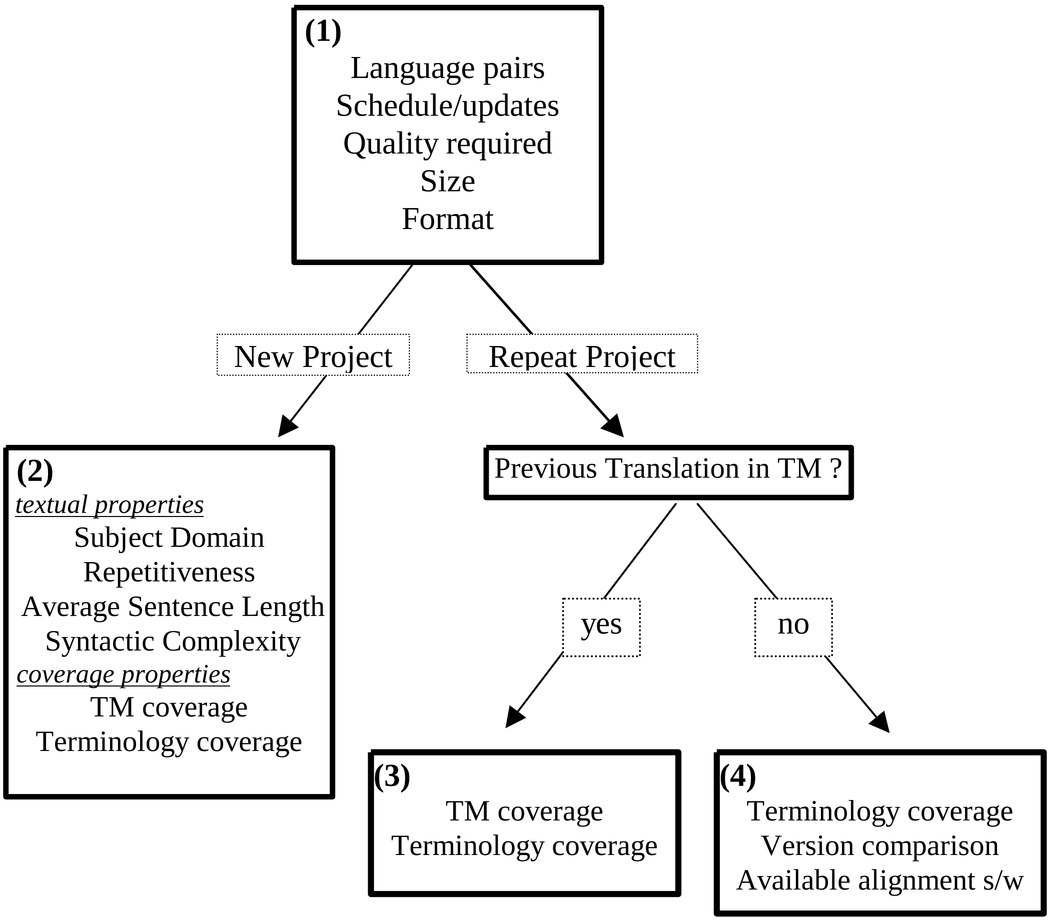

Figure 2. Workflow in Profiling a Project

The requester of a translation will have indicated the level of quality required for the final translation and this also has an impact on the way the translation should be carried out. For example, if the requester requires only a rough translation in order to get a general idea of the contents of a document, then it may be sufficient to translate it using an MT system with minimal post-editing afterwards. At the other end of the scale the translated text may be intended for publication with a very high level of quality not only in terms of accuracy but also in a fluent idiomatic style. In such a case one would expect that the translation process would involve a great deal more effort from a human translator and an associated increase in the time needed to complete the work. Although, if the current project is to translate an updated version of a previously translated text with the same high quality level which is already stored in a TM, then the amount of time and effort needed will be considerably reduced. The issue of formalising translation quality has long been a vexed question in translation technology community (see for example Sparck Jones & Galliers (1996) and references therein) but quality requirements cannot be ignored when routing a translation project. Indeed translation managers already take this into consideration every day in routing translation projects. In the current project profile we allow for three values for the attribute of required quality: *publication quality*, *information dissemination quality* or *browsing quality*, whilst different translation agent profiles will also contain information on the sort of quality which they can produce. Such values are necessarily rather arbitrary and may not be fine-grained enough for a particular user, so the translation manager has the opportunity to customise the types of values for quality to meet his normal assessment criteria. It should be noted that the final quality achieved is dependent upon the combination of different agents and resources which are employed and it is one of the tasks of the kernel to calculate the optimum combinations for a particular translation project.

In addition to the above information, more practical information about the files to be translated is also included in the profile, such as the size and format of the project files. The size of the project is defined both in terms of the number of separate files and the number of words contained in those files. A simple word counting procedure has been developed to calculate the length of the texts to be translated. Clearly the project files must also be in a format which is compatible with any translation tools which may be used, or convertible into a compatible format. Currently this information in input by the user, but it could also be discovered automatically.

Once the properties of the project which are applicable to all kinds of translation project have been recorded in the project profile, the user then decides whether the current translation project is a new project or whether a similar text or a previous version of the current text has been translated and is available for potential re-use in the current project. If the project is new (2) then a number of properties of the source text must be recorded in order to enable TransRouter to find suitable routes via which the translation could be carried out. First of all, the subject domain treated by the texts is recorded to help establish whether existing lexical or terminological resources should be investigated and which ones. Another important property is the amount of repetitiveness in the text. This may be a determining factor in deciding whether to use a TM application, although this is not the only factor to be taken into account since if it is expected that subsequent versions of the document will translated in the future this will also make a document a good candidate for translation with the support of a TM even though it may not be particularly repetitive in itself. A tool for estimating the amount of repetitiveness in a text is discussed in detail in section 3.1.1.

The average sentence length and the syntactic complexity (or simplicity) of the text will often have an effect on the quality of the translation which can be achieved by an MT system. A tool for analysing syntactic complexity will be briefly outlined in section 3.1.2. Even though a project is “new” it may be that some existing resources such as terminology, machine translation lexica or translation memories can be useful in the translation and therefore texts may be compared with the backend resources to determine their coverage (see section 3.2 below).

If, on the other hand, a translation of a previous version of the document exists (i.e. the current project is a "repeat project") then its potential for re-use must be investigated. If the translation is already stored in a TM (3) then once again the document should be compared with the TM and any available terminological resources to determine coverage. If, however, the previous translation is not in a TM (4) then we want to assess whether it is worth creating a TM from the previous translation to support the translation of the current project, and the effort this would require. In that situation not only terminology coverage should be analysed, but also the differences between the previous and current versions of the source text should be assessed. Finally, the availability of alignment software which enables a user to (semi-)automatically generate a TM from parallel texts will have an impact on the time and effort needed to create a new TM from the previous translation.

Since the process of profiling the translation project is interactive, the user can choose to ignore certain attributes and avoid analysing texts for particular properties. For example if the user does not have an MT system available that could be used in the translation (either because he does not have one at all or because e.g. it does not support the required language pairs) then analysing the syntactic complexity of the text may be a waste of time. Similarly issues of repetitiveness become much less interesting if a suitable TM is not available. Clearly an experienced manager will have an idea beforehand about which translators and/or tools are most likely to be best suited to a particular project. Therefore the user can also specify which agents and resources should be taken into consideration when calculating possible routes.

In the next sections we will describe in more detail the various tools which are being developed to analyse source texts. Some of the analysis tools are or will be based on commercially available state-of-the-art products or parts of such (currently: the translation memory coverage tool).

3.1 Textual Analysis Tools

There are several textual factors that work in favour of some translation agents or rule out others. The factors which can be measured by automatic means and that are taken into account in the TransRouter project are described below.

3.1.1 Repetitiveness

If all available candidate translation memories do not cover a new project too well, one may still consider using a TM system instead of using other translation agents. Factors that TM systems are good at, compared with other translation agents, are consistency of translations and reduction of translation effort. Both factors have to do with the same piece of text occurring more than once. If consistency is important, then even very few repetitions of a handful phrases may be reason enough to choose a TM system. If many pieces of text are repeated many times, a human translator would become more productive by using a TM system, provided that the repeated parts of texts are spotted, stored and - last but not least - retrieved during the translation process. In the following, we will concentrate on the reduction of translation effort, but we will return to consistency later.

"Useful repetitiveness" (the factor by which translation effort can be reduced by translating repeated text only once) is to some degree dependent on both the TM system and on the user's mode of working with that system. There is little doubt that repetitiveness at the sentence level leads to a quantifiable cost reduction, because sentences are the translation units that are stored in a translation memory and are retrieved most easily.

On the other hand, unless special care is taken, repetitions of sequences of only a few words are probably not taken advantage of by a TM system. Such systems are trained to retrieve translation units that match the source text to some degree of fuzziness that can be chosen by the user, but a high degree of fuzziness has some disadvantages. If the user allows a high degree of fuzzy matching and if an exact match for a sentence is missing, then the user may be confronted with an overwhelming amount of fuzzy match candidates and there is no guarantee that sentences containing short word sequences are among these proposals. If the user only accepts little fuzziness then repetitions of short phrases are going to be unnoticed for sure.

There is, however, a way to take advantage of the repetitions of short sequences of words: most TM systems (and some MT systems as well) have an associated term bank that can store phrases. If a user has the means of finding all repeating phrases, these (at least the meaningful ones) can be translated beforehand and stored in the term bank. Regardless of the fuzzy match threshold, the term bank will spot and present to the user those phrases that are contained in the current sentence.

So, in some way or another, repetitiveness at the sentence and at the less than sentence ("phrase") level can be taken advantage of. Long repeated sequences, such as lengthy standard expressions that are typical for the text or project at hand, can be stored in a translation memory and found by the fuzzy match mechanism, while short phrases can be stored in the term bank.

TransRouter’s repetitiveness tool will produce a numerical value for repetitiveness at the sentence level and another number for repetitiveness at the “phrase” level. The kernel can freely decide to only take sentence level repetitiveness into account, if the translation agent’s profile indicates that phrase level repetitiveness is not taken advantage of.

There are a few problems to be solved, however. To begin with, tools that detect repetitiveness at the phrase level do not exist or are, at least, not heard of in the translating industry[*]. And then: which algorithm finds all repetitions of word sequences? Is the concept of repetitiveness clear enough to design such an algorithm? For example, do repetitions of word sequences that occur within larger repeated word sequences add to the overall repetitiveness, or should we disregard such embedded repetitions? If we do disregard them, what about repeating word sequences that merely overlap with other repeating word sequences? Clearly, we do have to make sensible choices in order to arrive at a repetitiveness measure.

There is another factor that has to be taken into account: not all repeated sequences of words are equally good phrases or standard expressions. From the set of all repeated sequences all those sequences that a human would not regard as one unit (semantic or otherwise) have to be pruned.

In the TransRouter project, the authors are developing a tool that already in its present state does a good job at detecting useful word sequences and at computing a text's repetitiveness. It does so in a few steps: First, (nearly) all sequences of words that repeat are

[*] Nagao & Mori (1994) describe a method for the *exhaustive* enumeration of repeated n-grams in huge text corpora and point out some interesting applications. In contrast, our aim is to swiftly find *only the “best”* phrases in relatively modest-sized texts. Also, for calculating repetitiveness, overlaps between candidate-phrases must be resolved by filtering out the least promising of the conflicting phrases, thus further reducing the number of phrases.

detected, without regard to issues of overlap and embedding. Secondly, all found sequences are weighted and sorted according to their length, their frequency and to a factor that is high for sequences that contain relatively infrequent words and low for sequences that only contain frequent words. The resulting list has relatively "good" phrases at the start and "bad" phrases at the end. The third step is the pruning step, in which overlap and embedding are handled. All sequences in the sorted list, the best ones first, are counted once more in the text to be analysed, but this time words are not allowed to be part of more than one repeating word sequence. Sequences that after the second count occur zero or just one time are removed from the list: typically they were not useful sequences. The overall repetitiveness of the text is then defined as the ratio of the text length (number of words) and the reduced text length. The reduced text length is computed as the number of words that are not matched by any of the sequences in the pruned list plus the number of words in all sequences in the pruned list, each word sequence only counting once. A text without any repetitions whatsoever has a repetitiveness value of one, while any text with repetitions has a repetitiveness value that is greater than one.

As a by-product of computing the repetitiveness of a text, one obtains a list of repeated word sequences. In order to utilise a text's repetitiveness to reduce the translation effort, as many of these word sequences as possible must be pre-translated and stored in a term bank or a translation memory. One may choose to only pre-translate those word sequences that require exactly the same translation for each occurrence in the text. In that way the repetitiveness detector's main objective is to ensure consistency, while the reduction of translation time becomes of secondary importance.

Although repetitiveness is most advantageous for TM systems, the creation of a list of short repeated phrases may be a good supplement to the output of the terminology coverage tool (see below), which is important for all kinds of translation agents.

3.1.2 Sentence Simplicity

In order to analyse the syntactic complexity of a text, the University of Regensburg has developed a first version of a sentence simplicity checker. Rather than developing a tool that rivals the size and complexity of MT systems, a simpler approach has been chosen that defines three evaluation categories. Each characteristic of a text that influences sentence simplicity is assigned to one of these categories, depending on the “sophistication” of the simplicity contorting characteristic. In the first category are e.g. average sentence length and the average number of commas per sentence. In the second category there are slightly more sophisticated characteristics such as the number of auxiliary verbs. In the last, most sophisticated category belong e.g. ambiguous words. It is the kernel’s task to weight each of these simplicity-influencing characteristics, taking the agent profile of a candidate MT system into account.

3.2 Coverage Analysis Tools

The tools described in this section analyse texts with respect to coverage in the available resources.

3.2.1 Terminology Coverage

The presence of unknown terms in a text can severely degrade the output of an MT agent. Also TM systems can become much less productive if their associated term banks lack essential terms. The impact of low terminology coverage on MT systems is most severe, because it can be much more time consuming to make an MT lexicon entry than to make an entry in a term bank. TransRouter's terminology coverage tool (which is being developed by L&H Language Technology) works by comparing the words in the input material with a lexical resource that the user can specify, for example the contents of a term bank that is shared by several translation agents.

3.2.2 Translation Memory Coverage

The usefulness of a TM system for a given translation project depends in large part on the availability of a translation memory which covers the source text to a greater or lesser degree. TM-coverage will be great if a previous version of a text is stored in the Translation memory, but also a new text may be covered quite well by a big enough translation memory. (A strategy that is sometimes used by translation providers is to pool a single translation memory between many projects).

Translation memory coverage depends on the way a human translator uses a TM system. Many TM systems allow the user to disregard candidate stored translation units if the source text does not match the current segment (sentence or heading) well enough. Of course, if the user only allows little fuzziness then he will not receive nearly as many proposals as when he chooses a high level. For that reason, the output of the coverage tool consists of a number of coverage measures, each measure corresponding to a different fuzzy match threshold.

The coverage tool accesses the matching procedure available in the TM system in question. In the TransRouter project we have chosen not to try to find or develop a TM Coverage tool that is independent of any TM systems, because different systems may apply different heuristics to find fuzzy matches. Another complicating factor is that at the present stage, translation memories of different systems have different proprietary formats. That circumstance makes it less worthwhile to predict the coverage of a translation memory that is in a different format from the system which could be used. The situation may change when the exchange format for translation memories is introduced (OSCAR*), although the differences in fuzzy match heuristics probably will still be with us and make the development of an independent coverage measuring tool difficult, meaningless or both.

3.2.3 Version Comparison

If no usable translation memory exists, one may still reuse previous translations and create a translation memory, using, for example, an alignment tool. A version comparison tool can give an indication of the usefulness of this approach by assessing how similar the previous source text is to the current one. Many of the issues that are discussed under the repetitiveness tool are equally relevant for version comparison: consistency versus reduction

* The Translation Memory eXchange (TMX) standard by the Open Standards for Container/Content Allowing Re-use (OSCAR) group. See http://www.lisa.org/tmx/

of translation effort, sentence level similarity versus part-of-sentence level similarity, the influence that agent and user profiles have on the practically attainable gain in productivity due to text similarity.

Due to the similarities between repetitiveness detection and version comparison we plan to base the version comparison tool on much of the same technology as the repetitiveness detector.

4. Conclusions

In this paper we have described the role and process of profiling a translation project as a preliminary to calculating possible routings. By partially automating the process of assembling a project profile the process will be at once more reliable and consistent across different projects and faster than the informal methods used by translation managers up till now. Whilst this paper describes the profiling and analysis tools within the first TransRouter prototype, it is clear that in certain organisations where the choices between routes are not so varied, the entire TransRouter system may be more complex than is required. However depending on the set-up in the organisation a number of the component tools such as the repetitiveness detector, sentence simplicity checker or the unknown term detector, could be used as stand-alone tools which can support routing decisions

Acknowledgements

The TransRouter project (LE4-8345) is supported by the European Commission under its Fourth Framework Telematic Applications Programme. The following organisations make up the project consortium: The Localisation Resources Centre, Ireland; Berlitz International, Dublin; CST, Copenhagen; LTG, University of Edinburgh; L&H Language Technology GmbH & Co. KG; University of Geneva (TIM/ETI); University of Regensburg.The authors would like to thank the Commission and their partners in the TransRouter project.

References

Hammwöhner, Rainer: 1998, ‘Entscheidungsunterstützung bei der Planung von Übersetzungsprojekten’, in *Knowledge Management und Informationssysteme – Workflow Management, Multimedia, Knowledge Transfer, Proc. Des 6. Internationalen Symposiums für Informationswissenschaft: ISI ’98* Prague, pp 47-57

M. Nagao and S. Mori: 1994, ‘New Method of N-gram Statistics for Large Number of n and Automatic Extraction of Words and Phrases from Large Text Data of Japanese’, in *Proc. of 15th COLING*, pp.611-615

Sparck Jones, Karen & Julia Galliers: 1996, *Evaluating Natural Language Processing Systems. An Analysis and Review.* Lecture Notes in Artificial Intelligence 1083, Springer

TransRouter Consortium: 1999, ‘TransRouter: a decision support tool for translation managers’, to appear in *Proceedings of Machine Translation Summit VII (MT Summit ’99) "MT in the Great Translation Era"* Sept 13-17, 1999 Kent Ridge Digital Labs, Singapore.