Towards a Two-stage Taxonomy for Machine Translation Evaluation

Andrei Popescu-Belis and **Sandra Manzi** and **Maghi King**
ISSCO/TIM, Université de Genève
40, bd. du Pont d'Arve
CH-1211 Genève 4
Switzerland

{andrei.popescu-belis,sandra.manzi,margaret.king}@issco.unige.ch

Abstract

Evaluation guidelines for a given domain or task must be rooted in a general model for software evaluation. In this paper, we consider as a starting point the ISO/EAGLES guidelines for natural language processing software evaluation, which we first summarize. From these considerations, we derive several principles for a taxonomy aimed at the evaluation of machine translation (MT) systems. Then, we compare two editions of such a taxonomy, arguing in particular for a dichotomy relating user needs to system characteristics. We also outline the software infrastructure underlying the electronic publication and updating of the taxonomy, and conclude with a brief overview of the workshops that were staged to test, modify and disseminate the taxonomy.

1 Introduction: The ISLE Project

ISLE stands for International Standards for Language Engineering. The project derives a lot of its work and vision from the Eagles I and Eagles II projects that ran from 1993 to 1998. The Eagles guidelines for evaluation are themselves inspired by the ISO 9126 standard for software evaluation, with links and references to the further ISO 14598 standards.

1.1 Background

Machine translation (henceforth, MT) evaluation has a long history. Even very early in that history, attempts were made to produce well designed and well founded evaluation schemes. For example, even though much that is contentious surrounds the work of the ALPAC committee in the mid-60's, there is no denying that the ALPAC Report (Pierce *et al.* 66) contains a laudable attempt at good experimental design. Later, the Van Slype report for the European Commission (Van Slype 79) provided a very thorough critical survey of evaluations done to that date. Much valuable material can also be found in (AMTA 92), in the OVUM report of 1995 (Mason & Rinsche 95) and in the accounts of the (D)ARPA Machine Translation program, which, as is often the case with DARPA programs, contained a substantial evaluation component (White *et al.* 94).

However, no consensus has ever been reached in defining one single evaluation procedure, validly applicable to any MT system in any circumstances.

The reason for this is perhaps latent in the final phrase of the last sentence, "in any circumstances". MT is a complex exercise in itself. Its applications are manifold, and the possible contexts in which its results might be deployed are at least equally various. One paper (Church & Hovy 93) even suggests that very low quality MT may be useful in appropriate circumstances.

One way, therefore, to get a handle on the question of how an evaluation for an MT system should be designed, might be to take seriously the notion that judgement of a system critically depends on the context in which use of that system is envisaged. This was the strategy adopted by the JEIDA report (Nomura & Isahara 92), which attempted to systematize the description of a user's present situation and of their real needs. This lead to a way of identifying which of the seven tested MT systems, analyzed first using the same descriptive mechanisms in order to facilitate comparison with user needs, would best allow those needs to be fulfilled.

1.2 The Eagles Project

The idea that user requirements can be specified and used as a tool in designing an evaluation also lies behind the Eagles work on evalua-

tion – the acronym stands for Expert Advisory Groups on Language Engineering Standards. In the early 1990s, the European Commission saw a need to establish standards in the language engineering field, in order to encourage development of resources that could be produced in common for the common good. The first EAGLES project was set up to meet this need. A number of areas were picked out as suitable for such a development, and standards for evaluation were one of them (Eagles Evaluation Group 1996).

Influenced by the earlier work on evaluation very briefly summarized above, a primary insight during the first EAGLES period was that it was impossible to come up with one general recipe that would be valid for all evaluations in all circumstances. It was however believed that it should be possible to create a framework, theoretically sound and well-motivated, within which particular evaluations could be constructed. Following a common framework would facilitate the design of individual evaluations, would help to ensure their validity and would help to make shared work on definition of useful and valid metrics possible, as well as facilitating comparison of the results of evaluation. It was quickly realised that adopting this point of view also fitted well with work done by ISO on creating standards for the evaluation of software. ISO (1991) had published a standard – ISO/IEC 9126 – which set out the idea of a quality model for software. It stipulated a set of characteristics that were pertinent to the quality of software. A non-normative section of the publication also suggested guidelines for the use of the standard in designing a particular evaluation. A leading idea was that behind every evaluation a user is present, either explicitly or tacitly, and that every evaluation should start by trying to determine the needs of that user with respect to the software. It should be noted that the "user" is not necessarily an end user: at various stages of the product cycle it may be a developer, an investor, a manager, etc.

Potentially, the pre-eminence of a user with specific needs is in tension with the desire to build an evaluation framework of general validity. The EAGLES group resolved this tension by thinking in terms of classes of users, groups of individuals who could be considered to share certain needs. At the time, we called it the "consumer report paradigm" by analogy to the sorts of reports published in consumer magazines on products like washing machines or cars, where different aspects of the items being evaluated are linked to the different needs of certain consumers. This leads to the notion that in any given context, not all quality characteristics are of equal importance. In other words, it may well be that in one context reliability is critical and far more important than speed, while in another speed in obtaining even imperfect results outweighs both accuracy and reliability.

By pooling the requirements of each of the classes of users, though, it should be possible to come up with a general quality model. Such a model will be a structured collection of quality characteristics, usually broken down into subcharacteristics, bottoming out in a set of measurable attributes. Each measurable attribute is accompanied by one or more valid metrics, which yield a a score for that attribute when the evaluation is executed. The scores thus achieved can be combined in ways that reflect the relative importance of the attributes in a specific evaluation. Indeed, when designing a specific evaluation, it may be decided that some attributes are of no importance whatever and that they should be left out of the evaluation. In this way, a specific evaluation can be extracted from a general quality model.

In the first period of EAGLES work, it was thought important to validate the theoretical ideas built on ISO work by designing practical evaluations of relatively simple language engineering products. Thus, rather thorough evaluations of spelling checkers were designed and carried out, and preliminary work was done on designing evaluations for grammar checkers and for translation memory systems. It was both salutary and sobering to be forced to realize the sheer amount of meticulous attention to technical detail required in order to construct rigorous evaluations even of systems whose technology is relatively simple.

1.3 The Need for Evaluation Guidelines

Evaluation is needed to know if a given system fulfills its promises, however, it also serves other side purposes. Because the users have to state their needs and desiderata, they also reach a better understanding of what these are. Evalu-

ation brings out software's characteristics better than other methods (such as reports, software manuals, training courses). It focuses on software's pros and cons. It also allows users to compare different software products on a similar basis.

In our case, the final aim of translating a document can be different depending on the motivation. Some documents are translated only in order to determine possible relevance to somebody's query (for example, a researcher who needs information on a given subject, but who does not speak the language of a given document), whereas others have to be translated because of mandatory requirements (for example, international organizations who need documents to be made available in the languages of all member states). Still others have to be translated because they are intended for a given population with a given language (e.g., health information leaflets in a regional language). In certain cases, MT is envisaged as a way of lightening the burden on human translators, who would then revise (or more accurately, post-edit) the MT output. In yet others, the MT output is intended to be used in its raw form as, perhaps, an information dissemination tool. Thus, it becomes clear that the evaluation model used for MT software must be adaptable to the various tasks that the software may be assigned to do.

If the needs are so various, how can quality characteristics, metrics and measurement methods be defined? Benchmarking, in the sense of giving the same source text to various systems as well as to human translators and comparing the results, might be thought to be the simplest answer. But benchmarking MT systems is both time-consuming and expensive in man-power - and, in view of the above, may miss the point. The effort spent in carrying out numerous tests should be preceded by a clear analysis of which parameters are measured and how. However, some intuitive concepts such as fluency or clarity are vague, subjective, and sometimes overlapping. They need to be better defined in order to formulate a metric that would describe them.

Evaluation is in a way a "learn by experience" mixed with a "learn by error" pattern, where both errors and experiences might be those of other people. In ISLE, in the previous EAGLES projects, and in other related projects (e.g., TEMAA, TSNLP, DiET), we have been keenly aware of the shortfalls of "dive-in-head-first" in evaluation. Some general constraints on evaluation design emerged from these initiatives:

- evaluators must have clear in their minds what the features that they are seeking in a system are;
- evaluators must isolate the necessary characteristics required from the system;
- given the characteristics, evaluators must find a method to measure them **accurately**, which is often far from easy.

1.4 The Need for Coherent Metrics

The ISO 9126 standard (ISO 91) defines three stages in the evaluation process: definition of the required qualities (for a given piece of software), preparation of the evaluation, and the execution itself. For each characteristic that is evaluated, a *metric* must first be defined, then its values converted to *rating levels*, several rating levels being then integrated in a global score thanks to the *assessment criteria*. The rating and integration phases obviously produce different results depending on the goal of the evaluation. The evaluation guidelines do not indicate, in general, the precise coefficients to be used, but state the elementary characteristics and metrics that must be taken into account.

We consider metrics to be functions from a "quality space" onto the [0,1] or [0%, 100%] scale, often materialized for a particular system by the quality of its output or responses on some test data. Evaluators often need to compare various metrics, and work on this topic in MT evaluation is still ongoing. As a first step, several criteria were given in (Popescu-Belis 99) to estimate the coherence of a given metric. They are summarized here without giving their formal definitions:

- a metric must reach 100% for a perfect response and only in that case;
- a metric must reach 0% for "the worst possible" response and only in that case. It is in fact not easy to define the set of worst possible responses, therefore the two following criteria are sometimes substituted:

  - "bad" responses must receive low scores (examples or rather counter-examples are useful to examine this criterion);
  - the lowest possible scores of a metric must be close or equal to 0%;
- a metric must be "monotonous": if response a is obviously better than response b, then a's score must be higher than b's.

To compare two metrics, we can say that m_1 is more severe than m_2 if it yields lower scores for each possible response. The application of such criteria among the collections of measures in the taxonomy described below would highly increase its utility for evaluators.

2 Principles of a Taxonomy for MT Evaluation

2.1 Origins and First Version

While the EAGLES projects witnessed the development of a formalization of NLP software evaluation, it was only in the ISLE follow-up project that a systematic application to machine translation (MT) evaluation was started. The fundamental idea of the application was to list, in a hierarchical order, all the features pertaining to the quality of MT software and the main metrics associated to them.

The first draft of this *taxonomy* was developed at the Information Science Institute by Eduard Hovy and Elena Filatova, based mainly on previous work by Eduard Hovy presented at an EAGLES workshop (Hovy 99). This proposal contained two parallel levels of classification, one containing measures related to the purpose of an MT system, and the other containing measures related to the translation process itself. As the author acknowledged, these were only example taxonomies, and did not represent a final or complete classification. A significant choice was further made to design a hypertext document and make it available over the Internet. This first version is available (as of June 2001) at: `www.isi.edu/natural-language/mteval`.

The introduction to this first draft particularized the classification principle, stating that each characteristic is sub-divided into more detailed features by advocating a strong initial dichotomy: *"at the top level [of the taxonomy], there is only a single item to consider, namely the grand unified evaluation score; one level lower, there are only two items to consider, namely the scores for how well the system achieves the user's purposes and how well the system performs its internal operations."*

It is not completely clear, however, whether the *satisfaction of user needs* and the *system's own characteristics* are supposed to be two parallel aspects that have to be evaluated separately (thus yielding two independent scores), or whether they are somehow inter-related and this relation must be considered for the final score. If at first sight it seems that the first viewpoint was endorsed, the contents of the taxonomy suggest the second interpretation.

It must be noted that Hovy and Filatova leave open the following possibility at any given level of the taxonomy: *"in some cases, sibling points under one parent are alternatives; generally, the user will choose only one of them. [...] In other cases, the sibling points at a level are complementary, and the user can choose more than one."*. The first level seems nevertheless to have a special status.

The first draft departs however from both alternatives, in that it provides *"three different, parallel, [sub]taxonomies: User Purpose, Application Process, and General Software Characteristics."* In contradistinction, the second draft described here grants a special role to the first level articulation between user needs and system characteristics.

2.2 Key Principle of the Second Draft

The first draft was discussed during a workshop at the AMTA Conference Evaluation Workshop (Reeder & Hovy 2000). In an unpublished document, Maghi King synthesized the contributions of the participants to the discussions, and proposed a significant restructuring of the taxonomy while preserving most of its contents, i.e. the lower level categories and the individual characteristics.

The central point of this proposal was the articulation, for evaluation purposes, of the aspects already emphasized above, namely the user's needs and the system's characteristics. More precisely, taking into account the central role that the user of a system plays with respect to evaluation, the classification was divided into two complementary sections:

1. The first part relates user needs (in a very broad sense) to characteristics of the system. In other words, it is a repertoire of possible tasks for an MT system, user profiles, document types and qualities, organised hierarchically. Each item is refined using several alternative or complementary lower-level items. The entry for each item, taken from the first version, describes the system characteristics that are relevant for the item, and should hence be evaluated.
2. The second part is complementary to the first, since it contains, for each system characteristic, one or more metrics that have been proposed to quantify its quality level. These metrics were extracted, already in the first draft, from the MT litterature. Along with the metrics, each entry provides a definition, comments and references for the corresponding characteristic.

The evaluators of an MT system should first use the first part to identify, for the desired task and user profile, the relevant characteristics, then attempt to evaluate each of these using the metrics described in the second part. At the time of writing, the evaluators have to chose among the given metrics for a characteristic, but guidance will be provided in the next editions. Ideally, the consultation of the taxonomy should proceed using hyperlinks between the related items in the first and second parts. This is one of the reasons a hypertext taxonomy has been developed, and enhanced in the second draft, available at: `http://www.issco.unige.ch/projects/isle/taxonomy2`.

3 Implementation of a Structured Hypertext Taxonomy

While the interactivity of the first draft is a key feature that will be present in all the following versions, it appeared that several requirements should be addressed in later drafts: an easy mechanism for updating both form and content, a capacity to receive comments, the generation of a printable version, etc. The use of formatting tools related to the XML universe proved of great help on all these points.

3.1 Standardizing the Entries

One of the main advantages of XML is the separation between form and content. Therefore, the contents of the taxonomy can be stored in a conceptual format, and work on how they are dispayed can proceed separately. Regarding content, we propose a more formal definition: a *taxum* (pl. *taxa*) is the building block of the taxonomy, i.e. a feature or characteristic that is relevant to MT evaluation. The taxa can either be terminal (no further subdividsions) or non-terminal. As before, taxa at a given level (children of a same taxum) can be mutually exclusive or not.

The formal structure of a taxum is described using a DTD. The entries of the first draft have been converted to individual files, each containing a `<taxum>...</taxum>` element. The contents (markups) of the `taxum` element are given below:

```
<!ELEMENT taxum (index-number,
                 child-index-number*,
                 parent-index-number,
                 name,
                 definition,
                 how-to-measure,
                 references?,
                 comments?)>
```

For the time being, the same taxum structure is used in both parts of the taxonomy. Most of the fields are of course common, but there is a significant difference in the `how-to-measure` elements: while taxa from the first part contain here the system characteristics that are relevant for a user need, taxa from the second part contain here one or several metrics for a characteristic. The `how-to-measure` element is an historical remnant from the first version, and will be replaced with a more evocative field name in each part of the taxonomy.

3.2 The Life-cycle of the Taxonomy

All the relevant information is contained in the individual `<taxum>` files, which receive a unique, arbitrary index number. In order to modify or update the taxonomy, one simply edits the `<taxum>` files, changing either the contents (name, definition, how-to-measure) or the parent/children indexes. The index numbers are never modified, new numbers being used for new taxa. Modifications in the layout styles are not done at this level, but at the formatting level.

The generation of a readable version of the taxonomy relies essentially on the XSL mechanism. Stylesheets allow generation of HTML files for each taxum, or alternatively of a single HTML file corresponding to a printable version of the taxonomy. A module (under construction) infers the classification itself from the `<child-index-number>` and `<parent-index-number>` elements, and generates the corresponding HTML file. The other files that constitute the taxonomy website (frame definition, introduction, references, glossary) are not affected by the updating of the taxonomy.

The evolution of the taxonomy is summarized in Figure 1 at end of the article. Starting with the XML files for the taxa, the XSL stylesheets and other scripts generate the website allowing users to consult and/or print the taxonomy. A comment function is automatically embedded in the HTML taxum files, allowing users to comment upon individual taxa or upon the whole classification. These comments and those received directly by the developers, e.g. in the series of workshops organized through the ISLE project, are gradually fed back to the taxum files. Once these suggestions have been validated, a new version of the HTML files is generated.

4 Perspectives

Dissemination of the taxonomy and feedback was sought through hands-on evaluation workshops, in which the participants use the taxonomy for sample evaluations defined on the spot. Four workshops have been scheduled: October 2000, April, June and September 2001 (see for instance the Geneva workshop website at `http://www.issco.unige.ch/projects/isle/mteval-april01`). It is too early to provide general conclusions, but the first three workshops pointed at the following issues that have to be answered soon, apart from the encouraging remarks that we globally received.

At the content level, more bibliographical work should be done to include also more recent contributions to the field. A more thorough description of the metrics should be extracted, and some comparison between metrics proposed for a given feature should appear in the taxonomy, along with some guidance for choosing one. The observations of the website use show that comments are still insufficiently used – feedback should increase, at least in these initial phases.

It has also been noted that participants often focus on the "functionality" characteristic – closest to the intuitive notion of software quality – and tend to neglect other features. Therefore, functionality should probably be more developed in future drafts (with a special discussion of the difficulty of finding good metrics) whereas the importance of the other features should also be better explained.

The present taxonomy sets up a schema for easing the process through which evaluators select features in a fine-grained manner, and to evaluate the quality of an MT system according to those features. Moreover, it appears that if a given feature is very important, fine-graining it might be useful for a better representation. The taxonomy will be of course kept alive through its web interface even after the end of the ISLE project. We intend to keep integrating as much as possible comments and feedback from evaluators, who are always welcome to contact us.

5 References

Pierce, J.R., Carroll, J.B., Hamp, E.P., Hays, D.G., Hockett, C.F., Oettinger, A.G. & Perlis, A. (1966) – *Computers in Translation and Linguistics (ALPAC)*. National Academy of Sciences, National Research Council Publication 1416, Washington, D.C.

AMTA (1992) – *MT Evaluation: Basis for future directions*. San Diego. Available from AMTA, Washington, D.C.

Church, K. & Hovy, E., (1993) – *Good Applications for Crummy MT*. Machine Translation 8, p.239-258.

EaglesEvaluationGroup (1996) – *EAGLES Evaluation Group, Final Report EAG-EWG-PR.2*. Center for Sprogteknologi, Copenhague, October 1996.

Hovy, E. (1999) – *Toward Finely Differentiated Evaluation Metrics for Machine Translation*. Proceedings of the EAGLES Workshop on Standards and Evaluation. Pisa, Italy.

ISO (1991) – *International Standard ISO/IEC 9126: information technology / software product evaluation / quality characteristics and guidelines for their use*. International

Organization for Standardization & International Electrotechnical Commission, Geneva.

Nomura, H. & Isahara, J. (1992) – "The JEIDA Report on machine Translation". In AMTA 1992, *Proceedings of the AMTA Workshop on MT Evaluation.*

Mason, J. & Rinsche, A. (1995) – *Translation Technology Products.* OVUM, London.

Popescu-Belis, A. (1999) – "L'évaluation en génie linguistique: un modèle pour vérifier la cohérence des mesures". *Langues: cahiers d'études et de recherches francophones*, vol. 2, n. 2, p.151-162.

Reeder, F. & Hovy, E., eds. (2000) – *Workshop on Machine Translation Evaluation at AMTA-2000.* Mexico, 10 October 2000.

Van Slype, G. (1979) – *Critical Study of Methods for Evaluating the Quality of Machine Translation.* Prepared for the European Commission, DG XIII. Report BR 19142.

White, J., et al. (1994). – *ARPA Workshops on Machine Translation.* Workshops on comparative evaluation. PRC Inc., McLean, VA.

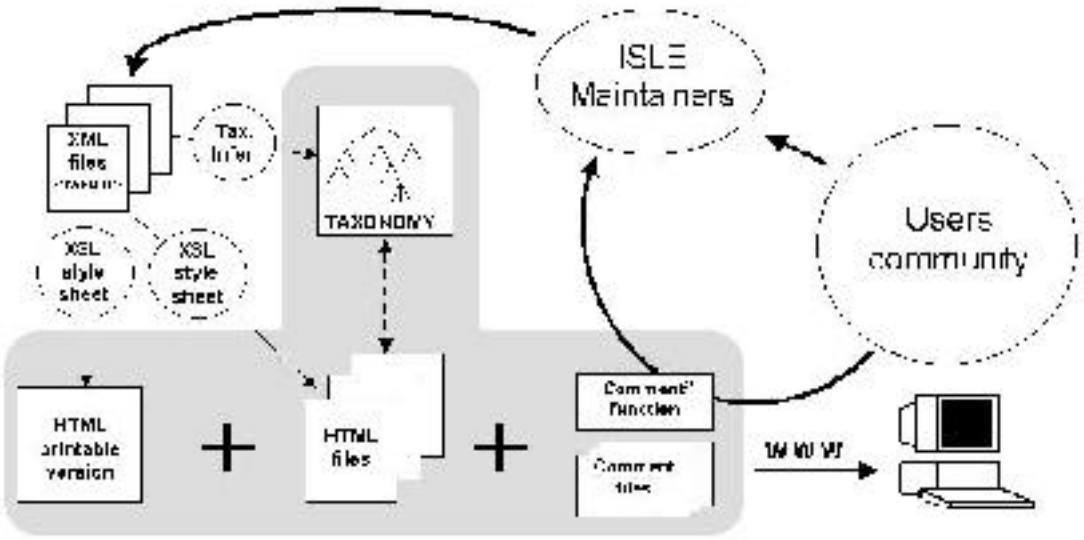

Figure 1. Life-cycle of the taxonomy with feed-back loop