Principles of Context-Based Machine Translation Evaluation *

Eduard Hovy (`hovy@isi.edu`)
USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292-6695, USA

Margaret King (`margaret.king@issco.unige.ch`)
University of Geneva, ISSCO/TIM/ETI
40 Bvd. du Pont d'Arve
CH–1211 Geneva 4, Switzerland

Andrei Popescu-Belis (`andrei.popescu-belis@issco.unige.ch`)
University of Geneva, ISSCO/TIM/ETI
40 Bvd. du Pont d'Arve
CH–1211 Geneva 4, Switzerland

Abstract.
This article defines a Framework for Machine Translation Evaluation (FEMTI) which relates the quality model used to evaluate a machine translation system to the purpose and context of the system. Our proposal attempts to put together, into a coherent picture, previous attempts to structure a domain characterized by overall complexity and local difficulties. In this article, we first summarize these attempts, then present an overview of the ISO/IEC guidelines for software evaluation (ISO/IEC 9126 and ISO/IEC 14598). As an application of these guidelines to machine translation software, we introduce FEMTI, a framework that is made up of two inter-related classifications or taxonomies. The first classification enables evaluators to define an intended context of use, while its links to the second classification provide a relevant quality model (quality characteristics and metrics) for the respective context. The second classification thus provides definitions of various metrics used by the community. Further on, as part of ongoing, long-term research, we describe the analysis of these metrics, first from the general point of view of 'meta-evaluation', then focusing on examples. Finally, we explain how consensus towards the present framework is sought for, and how feedback from the community is taken into account in the FEMTI life-cycle.

Keywords: MT Evaluation, Quality Model, Evaluation Metrics, Context-Based Evaluation

---

[*] Part of this work has been supported through the ISLE project (International Standards for Language Engineering) by the United States National Science Foundation, the European Union and the Swiss Government.

1. Introduction

Evaluating machine translation (MT) is important for everyone involved: researchers need to know if their theories make a difference, commercial developers want to impress customers, and users have to decide which system to employ. As a result, the literature is replete with MT evaluations and evaluation studies—it has even been said that more has been written about MT evaluation over the past 50 years than about MT itself! However, just as it is nonsensical to ask "what is the best house?", it is nonsensical to ask "what is the best MT system?". No simple answer can be expected or given when evaluating MT systems. As argued in (Church and Hovy, 1993), even poor MT can be useful, or even ideal, in the right circumstances.

Given the richness of the literature, and the complexity of the enterprise, there is a need for an overall perspective, something that helps the potential evaluator approach the problem in a more informed and standardized manner, and that will perhaps pave the way towards an eventual theory of MT evaluation.

In this paper, we advocate a global principle-based approach to MT evaluation, which takes into account the fact that no unique evaluation scheme is acceptable for all evaluation purposes.

Therefore, we introduce the Framework for MT Evaluation in the ISLE Project (FEMTI) which enables evaluators to parametrize the quality model (the set of desired system qualities and their metrics) depending on the intended context of use. Building upon previous work, our proposal also aims at applying to MT the ISO/IEC standards for software evaluation. However, we do not propose new metrics for any particular attribute, or attempt to automate the evaluation process, or to analyze performances of human judges. Our main effort is to build a coherent overview picture of the various features and metrics that have been used in the past, to offer a common descriptive framework and vocabulary, and to unify the process of evaluation design.

The paper proceeds as follows. First, we review the main evaluation efforts (Section 2), starting with MT evaluation, moving on to natural language processing (NLP) evaluation, and ending with the ISO/IEC standards. Then, we situate our proposal in this context, and define FEMTI's main theoretical stance, that is, the articulation between two classifications, one relating the context of use to the quality characteristics of the system and of its output, the other one relating the quality characteristics to their metrics (Section 3). In Section 4 we provide an overview of the contents of these classifications, focusing on output quality. Proposals for the dissemination and updating of this work are given in Section 5, while Section 6 provides long-term guidelines for further refinements and analyses related in particular to metrics. A conclusion and some perspectives end the article (Section 7).

2. Formalizing Evaluation: from MT to Software Engineering

The path to a systematic picture of MT evaluation is long and hard. One of the first quality judgments was the comparison of a source sentence with its reverse translation from the target sentence, as in the much quoted though anecdotal “the spirit is willing but the flesh is weak” *vs.* “the vodka is strong but the meat is rotten”. For a long time, such intuitive (counter-)examples, frequently tailored for a particular system, counted as evaluation. However, this approach is hardly systematic, and makes comparisons between systems difficult. Since then, many alternative evaluation schemes and even more metrics have been proposed. Unfortunately, however, the relations between them have usually remained unclear.

In this section, we present the lessons learned from previous research on evaluation, structured here as a bottom-up quest for standardization and interoperability of evaluation schemes. Indeed, not only are the initiatives for standardization more developed at the general level of software engineering (see Section 2.3), but as MT systems are pieces of software, any proposal at the MT level must conform to standards and best practice procedures at the general software level. A view of an intermediate level, evaluation of MT software *qua* NLP software, offers one important view on the origins of our proposal (Section 2.2).

2.1. Previous Approaches to MT Evaluation

While it is impossible to write a comprehensive overview of the MT evaluation literature, certain tendencies and trends should be mentioned. First, throughout the history of evaluation, two aspects—often called *fluency* and *fidelity*—stand out. Particularly MT researchers often feel that if a system produces lexically and syntactically well-formed sentences (i.e., high fluency), and does not distort the meaning (semantics) of the input (i.e., high fidelity), then the evaluation results are considered good enough. System developers and real-world users often add other evaluation measures, notably *price*, *system extensibility* (how easy it is for a user to add new words, grammar, and transfer rules), and *coverage* (specialization of the system to the domains of interest). In fact, as discussed in (Church and Hovy, 1993), for some real-world applications quality may take a back seat to these factors.

Various ways of measuring *fluency* have been proposed, some focusing on specific syntactic constructions, such as relative clauses, number agreement, etc. (Flanagan, 1994), others simply asking judges to rate each sentence as a whole on an *N*-point scale (White and O'Connell, 1994; Doyon et al., 1998), and others automatically measuring the perplexity of a target text against a bigram or trigram language model derived from a set of ideal translations (Papineni et al., 2001). The amount of agreement among such measures has never been studied.

Fidelity is usually measured on an *N*-point scale by having judges rate how well each portion of the system's output text expresses the content of an equivalent portion of the source text (in this case bilingual judges are required) or of one or more ideal (human) translations (White and O'Connell, 1994; Doyon et al., 1998). A proposal to measure fidelity automatically by projecting both system output and a number of ideal human translations into a vector space of words, and then measuring how far the system's translation deviates from the mean of the ideal ones, is an intriguing idea whose generality still needs to be proved (Thompson, 1992). In similar vein, it may be possible to use the abovementioned perplexity measure also to evaluate fidelity against a set of ideal translations (Papineni et al., 2001).

Paralleling the work on EAGLES, the Japanese JEIDA study of 1992 (Nomura, 1992; Nomura and Isahara, 1992) identified two sets of 14 parameters each: one that characterizes the desired context of use of an MT system, and the other that characterizes the MT system and its output. A mapping between these two sets of parameters allows one easily to determine the degree of match, and hence to predict which system would be appropriate for which user. In a similar vein, various companies published large reports in which several commercial MT systems are compared thoroughly on a few dozen criteria (Mason and Rinsche, 1995; Infoshop, 1999). The OVUM report (Mason and Rinsche, 1995) includes usability, customizability, application to total translation process, language coverage, terminology building, documentation, and others.

Other evaluations are even more explicitly extrinsic, that is, use-oriented, to use the term of Sparck-Jones and Galliers (1996). Tomita (1992) describes an evaluation in which various MT systems translated the texts used in a TOEFL (Test of English as a Foreign Language) test from English into Japanese. He then measured how well students answered TOEFL's comprehension questions, using the translated texts instead of the original English, and ranked the MT systems as passing or failing. A more general view of task-based evaluations is provided by White and Taylor (1998).

The variety of MT evaluations is enormous, and spans a range from the highly influential ALPAC Report (Pierce et al., 1966) to the largest ever (and highly competitive) MT evaluations, funded by the US Defense Advanced Research Projects Agency DARPA in 1992–1994 (White and O'Connell, 1994) and beyond. An ongoing evaluation campaign (2201–2002) organized by NIST and making use of both human and automated metrics is described in several reports available at `http://www.nist.gov/speech/tests/mt/mt2001/`. Several recent proposal towards the automation of MT evaluation, in addition to (Papineni et al., 2001; Thompson, 1992), have been made in (Nießen et al., 2000; Rajman and Hartley, 2002). Van Slype (1979) produced a thorough study reviewing MT evaluation at the end of the 1970s, while reviews for the 1980s can be found in (Lehrberger and Bourbeau, 1988; King and Falkedal, 1990), and in the influential papers by Kay (1980) and by Nagao (1989). The pre-AMTA workshop on evaluation contains a useful set of papers (AMTA, 1992).

2.2. The Eagles Guidelines for NLP Software Evaluation

The European Eagles initiatives came into being as an attempt to create standards for language engineering. They were born out of a perception that linguistic resources were essential to progress in the area, but were expensive and time consuming to create. Agreed standards for the form and content of resources would facilitate sharing resources across projects, product development and different applications. The first areas to be attacked in the first phase of the initiative (1993–95) were corpora, lexicons, grammar formalisms and evaluation methodologies. It was accepted that no single evaluation scheme could be developed even for a specific application, simply because what counted as a "good" system would depend critically on the use to which the system was to be put and on its potential users. However, it did seem possible to create what was called a general framework for evaluation design, which could guide the creation of individual evaluations and make it easier to understand and compare the results. An important influence here was a report by Sparck-Jones and Galliers (1993), later reworked and published in book form (Sparck-Jones and Galliers, 1996).

Influenced by earlier work in evaluation, including the ISO/IEC 9126 standard published in 1991 (see next section), these first attempts proposed the creation of a quality model for NLP systems in general in terms of a hierarchically structured classification of features and attributes, where the leaves of the hierarchy were measurable attributes, to which specific metrics were associated. The quality model in itself was intended to be very general, covering any feature which might potentially be of interest to any user. The specific needs of a particular user or class of users were catered for by extracting from the general model just those features relevant to that user, and by allowing the results of applying metrics to be combined in different ways (EAGLES-Evaluation-Workgroup, 1996).

These first attempts at providing a theoretical framework were validated by application to quite simple examples of language technology: spelling checkers were examined fairly thoroughly, and preliminary work was done on drawing up quality models for grammar checkers and translation memory systems (TEMAA, 1996). Whilst the case studies tended to confirm the utility of the theoretical framework, they also stressed the attention that had to be paid to sheer meticulous detail in designing a valid evaluation.

In the second phase of the EAGLES initiative (1995–1996), work on evaluation was essentially limited to consolidation and dissemination of the guidelines. During this time, the EAGLES methodology was used outside the project to design evaluations of a dialog system (Blasband, 1999) and of a speech recognition system (in a private company), as well as a comparative evaluation of a number of dictation systems (Canelli et al., 2000). The designers of these evaluations provided useful feedback and encouragement to the EAGLES evaluation work group. Also during the second phase, the group came into closer contact with the ISO/IEC work on the evaluation of software in general.

When the ISLE project (International Standards for Language Engineering) was proposed in 1999, it transpired that the American partners had also been working along the lines of taxonomies of features (Hovy, 1999), focusing explicitly on MT and developing in the same formalism a taxonomization of user needs, along the lines suggested by the JEIDA study (Nomura, 1992). The Evaluation Working Group of the ISLE project therefore decided to concentrate on MT systems, refining and extending the taxonomies that had been proposed. It is essentially this work which is described here[1].

2.3. THE ISO/IEC STANDARDS FOR SOFTWARE EVALUATION

In this section, we summarize the ISO/IEC standards for software quality, a series that has become more and more detailed. We propose

---

[1] It should be noted that the ISLE project also covers considerable work in other areas, especially on standards for NLP lexicons and multimodal human-computer interaction, which is not reported on here. For more information about the Evaluation Working Group, please visit our web site at: `http://www.issco.unige.ch/projects/isle/ewg.html`.

first an overview of the series, then summarize the main points that are relevant to our framework, with a final focus on a three-stage evaluation process.

2.3.1. *A Growing Set of Standards*

The International Organization for Standardization (ISO) together with the International Electrotechnical Commission (IEC) have initiated in the past decade an important effort towards the standardization of software evaluation. The ISO/IEC 9126 standard (ISO/IEC-9126, 1991) appears as a milestone that attempted to define the concept of *quality*, by decomposing software quality into six generic *quality characteristics*. Evaluation is the measure of the quality of a system, in a given *context*, as stated by the definition of quality as

> the totality of features and characteristics of a product or service that bear on its ability to satisfy stated or implied needs (ISO/IEC-9126, 1991, p. 2).

The 1991 version of ISO/IEC 9126 was a prelude to more detailed developments in evaluation standardization, on one side concerning the definition of quality models and associated metrics, and on the other concerning the organization of the evaluation process. Subsequent efforts led to a set of standards, some still in draft versions at the time of writing. It appeared that a new series was necessary for the evaluation process, of which the first in the series (ISO/IEC-14598-1, 1999) provides an overview. Although this overview covers all aspects of the evaluation process, including the definition of a quality model, this latter point is elaborated upon in a new version of the ISO/IEC 9126 standard, which will finally comprise four inter-related standards: standards for software quality models (ISO/IEC-9126-1, 2001), for external, internal, and quality in use metrics (ISO/IEC 9126-2 to 4, unpublished).

Regarding the 14598 series, now completely published, volumes subsequent to ISO/IEC 14598-1 focus on the planning and management of the evaluation process (ISO/IEC-14598-2, 2000), on its documentation (ISO/IEC-14598-6, 2001), and apply the generic organization framework to developers (ISO/IEC-14598-3, 2000), acquirers (ISO/IEC-14598-4, 1999) and evaluators (ISO/IEC-14598-5, 1998). Very briefly, in the last three documents, different aspects are emphasized depending on the goal of the evaluation: it is assumed that developers evaluate their products at early stages, whereas acquirers deal with one or more end-products that must answer specific needs. Evaluators represent in this view a more independent body, assessing the quality of software from a more generic or decontextualized point of view. In the following section, we summarize from the point of view of the evaluators the basic framework that emerges from the ISO/IEC 9126 and 14598 series of standards, concentrating on the key issue of defining a quality model.

2.3.2. *Definition of a Quality Model*

According to ISO/IEC 14598-1 (1999, p. 12, fig. 4), the software life-cycle starts with the analysis of user needs that will be answered by the software, which determine a set of software specifications. From the point of view of quality, these are the *external quality requirements*. Then, the software is built during the design and development phase, when quality becomes an *internal* matter related to the characteristics of the system itself. Once a product is obtained, it becomes possible to assess its internal quality, then the external quality, i.e., the extent to which it satisfies the specified requirements. Finally, turning back to the user needs that were at the origin of the software, *quality in use* is the extent to which the software really helps users fulfill their tasks (ISO/IEC-9126-1, 2001, p. 11). According to ISO/IEC, quality in use does not follow automatically from external quality since it is not possible to predict all the results of using the software before it is completely operational.

In the case of MT software, an important element of the life-cycle must be taken into account: there is no straightforward link, in the conception phase, from the external quality requirements to the internal structure of a system. To use an example, it may be possible to design a system to sell books online, based on the requirements that the system handle a database of at least one billion titles and 100,000 customers. However, it is at present impossible to infer the design of a system that translates the books, from requirements that the target must be a hundred languages. Quite frequently, research in MT (and in other branches of NLP and Artificial Intelligence) shortcuts the specification-to-design phase, with the result that evaluators must then define their own contexts of use and select external quality requirements. As a consequence, the relation between external and internal qualities is quite loose in the case of MT.

According to (ISO/IEC-9126-1, 2001), software quality results in general from six *quality characteristics*:

- functionality
- reliability
- usability
- efficiency
- maintainability
- portability

Already present in (ISO/IEC-9126, 1991), these characteristics have been refined in the more recent version of the standard, through a loose hierarchy of sub-characteristics (still domain-independent) that may contain some overlappings (ISO/IEC-9126-1, 2001, A.1.1, p. 13). The terminal entries are always measurable features of the software, that is, *attributes*. Conversely,

> a *measurement* is the use of a *metric* to assign a value (i.e., a measure, be it a number or a category) from a scale to an attribute of an entity (ISO/IEC-14598-1, 1999).

Therefore, several (alternative) metrics can be used to assign a level to a quality attribute. We will adopt in this paper the term '*metric*', as proposed by ISO/IEC, even though the scoring methods do not always have the mathematical properties of a metric[2]. The term 'measure' or 'measurement' is used, in agreement with ISO/IEC, for the application of a 'metric' to a translation or to a system.

The six top level quality characteristics are the same for external as well as for internal quality. The hierarchy of sub-characteristics may be different, whereas the attributes (the leaves of the hierarchy) are certainly different. Indeed, external quality is measured through external attributes (related to the behavior of the system) and internal quality is measured through internal attributes (related to intrinsic features). When there is a close link between specification and design of a system, a similar connection holds between internal and external attributes (ISO/IEC-9126-1, 2001, A.1.1, p. 13). As noted above, this is unclear in the case of MT software, despite examples such as the relation between the 'dictionary size' and the 'number of (un)translated words'.

Quality in use results from four characteristics (ISO/IEC-9126-1, 2001, p. 11):

- effectiveness
- productivity
- safety
- satisfaction

---

[2] A metric is a function that associates to two elements A and B of a set a positive number or distance $d(A, B)$, with the following properties: (a) the distance from a point to itself is zero ($d(A, A) = 0$); (b) the distance from A to B is the same as from B to A ($d(A, B) = d(B, A)$); (c) the distance from A to C is always shorter than the distance between A and B plus the distance between B and C, whatever A, B, C are ($d(A, C) \leq d(A, B) + d(B, C)$). Evaluation 'metrics' can sometimes be conceived as the distance between a system's response and a set of ideal responses, but depending on the scoring method, property (c) is not always satisfied.

These can only be measured in the operating environment of the software, and seem therefore less prone to standardization. Part 4 of ISO/ IEC 9126 will be dedicated to quality in use metrics, and for the moment other usability studies shed light on this matter (Daly-Jones et al., 1999).

2.3.3. *Stages in the Evaluation Process*

Apart from the preceding concepts, the ISO/IEC standards also outline the evaluation process, generalizing a proposal already present in the first ISO/IEC 9126[3]. The five consecutive phases of the process are emphasized differently according to whom the initiators of the evaluation are, developers, acquirers or evaluators:

- establish the quality requirements (the list of required quality characteristics)
- specify the evaluation (i.e., specify the measurements and map them to the requirements)
- design the evaluation, producing the evaluation plan (i.e., document the procedures used to perform measurements)
- execute the evaluation, producing a draft evaluation report
- conclude the evaluation

According to ISO/IEC 14598-5, the specification of measurements starts with a distribution of the evaluation requirements over the components of the evaluated system. Then, each required quality characteristic must be decomposed into the relevant sub-characteristics, and so on. Metrics must be specified for each of the attributes arrived at in this decomposition process. More precisely[4], three elements must be distinguished in the specification, design, and execution of an evaluation; the following order applies to execution:

a. application of a metric

b. rating of the measured value

c. integration or assessment of the various ratings

It must be noted that *a.* and *b.* may be merged in the concept of 'measure', as in ISO/IEC 14598-1, and that integration *c.* is optional. Indeed, in the ISO/IEC 14598 series, the integration of measurements towards a *final score* is not a priority, unlike the case of

---

[3] The standards become indeed more and more general, hence abstract—compare (ISO/IEC-9126, 1991, p. 6) and (ISO/IEC-14598-5, 1998, p. 7)

[4] We follow here (ISO/IEC-9126, 1991, p. 6) as well as (ISO/IEC-14598-1, 1999, p. 15–17).

many NLP and MT evaluation campaigns. One has to turn back to the first ISO/IEC 9126 standard to find a mention of the integration stage. Therefore, at the level of concrete evaluations of systems, the above three stage distinction, advocated also by EAGLES (1996), seems to us particularly useful.

2.3.4. *Formal Definition of the Stages*

More formally, following previous work (Popescu-Belis, 1999a; Popescu-Belis, 1999b), let S be a system for which several attributes must be evaluated, say $A_1, A_2, \ldots, A_n$. First, the system is measured by a metric m_{A_i} for each attribute, producing a value on a scale that is intrinsic to the metric m_{A_i}, in general not tailored to reflect a quality value. If the set of all systems is noted Σ and the scale associated to the metric m_{A_i} is the interval $[\inf(m_{A_i}), \sup(m_{A_i})]$, we define:

a. application of a metric:

$$\left| \begin{array}{r} m_{A_i} : \Sigma \longrightarrow [\inf(m_{A_i}), \sup(m_{A_i})] \\ S \longmapsto m_{A_i}(S) \end{array} \right.$$

The metric may be *scaled* in order to obtain a numeric value that is easier to apprehend by the human evaluator, for instance a scaling that normalizes a metric to a given interval of values.

In any case, each measured value must be rated with respect to the desired values, say a set of scores or ratings $\{r_1, r_2, \ldots, r_p\}$. This set may be discrete (as in the notation chosen here) or continuous; some metrics may require a unique set, while others may share the same value set (for example, a numeric scale). The mapping between the measured values and the ratings reflects the human judgment of an attribute's quality.

b. rating of the measured value:

$$\left| \begin{array}{r} r_{A_i} : [\inf(m_{A_i}), \sup(m_{A_i})] \longrightarrow \{r_1, r_2, \ldots, r_p\} \\ m_{A_i}(S) \longmapsto r_{A_i}(S) \end{array} \right.$$

If integration of the ratings is needed—that is, in order to reduce the number of ratings at the conclusion of the evaluation—then an assessment criterion should be used, typically some weighted sum α between the ratings.

c. assessment of several ratings:

$$\left| \begin{array}{r} \alpha : \{r_1, r_2, \ldots, r_p\}^n \longrightarrow \{r_1, r_2, \ldots, r_p\} \\ (r_{A_1}(S), r_{A_2}(S), \ldots, r_{A_n}(S)) \longmapsto \alpha(S) \end{array} \right.$$

A single final rating is often less informative, but more adapted to comparative evaluation, while an expandable rating, in which a single value can be decomposed on demand into multiple components, is conceivable when the relative strengths of the component metrics are understood. This is a possible future development within the ISLE project. However, the EAGLES methodology considers the set of ratings to be the final result of the evaluation (EAGLES-Evaluation-Workgroup, 1996, p. 15).

To conclude, it is quite apparent that the ISO/IEC standards have been conceived as an abstract framework that suits the needs of many communities that develop or use software. We particularize hereafter this framework to MT evaluation, starting with an essential factor that influences the choices that are made among quality characteristics, namely the context of use. The link with previous standardization efforts in MT evaluation is thus visible.

3. A Double Articulation: Context of Use / Quality Characteristics / Metrics for Machine Translation

Just as one cannot determine what is "the best house", one cannot expect to determine the best MT system without further specifications. Just like a house, an MT system is intended for certain users, located in specific circumstances, and required for specific functions. Which parameters to pay attention to, and how much weight to assign each one, remains the prerogative of the user/evaluator. The major role of the context for effective system evaluation has been long understood (cf. Section 2.2 on EAGLES), and has been a focus of study for MT specifically in the JEIDA report (Nomura, 1992).

3.1. THE CONTEXT OF USE IN THE ISO/IEC STANDARDS

Despite the ISO/IEC definition of quality as the extent to which a system meets various user needs, the context of use plays a somewhat lesser role in ISO/IEC. The standards mention the context of use when defining quality in use, but discuss only quality in use metrics, without providing any link with other internal or external metrics. Also according to ISO/IEC, the analysis of the context is done at the beginning of the software's life-cycle (ISO/IEC-14598-1, 1999, p.12) and conditions software specifications.

There is however no overall indication how to take into account the context of use in evaluating a product, apart from the two points outlined hereafter.

3.1.1. *Influence of the Target Software Integrity*

The ISO/IEC standard for acquirers exemplifies the link between the desired *integrity* of the evaluated software and the activities related to evaluation, in particular the choice of a quality model (ISO/IEC-14598-4, 1999, Annex B, pp. 21-22). The tables used as examples are not normative, and even the notion of software integrity is not fully defined—roughly, the higher the risk from software malfunction, the higher the desired integrity. For low integrity, only a few activities related to evaluation are required, such as the preparation of the quality requirements, the external evaluation proper, the study of the product's operating history. For medium and high integrity, other procedures must be carried on, such as assessing supplier capability or evaluating the supplier's software process.

Closer to our focus on quality models, ISO/IEC 14598-4 (*ibid.*) also proposes a table with the prioritized quality characteristics for low *vs.* high target software integrity. The six ISO/IEC 9126 characteristics are ordered differently in the two cases, from functionality to maintainability for low integrity, and from reliability to portability for high integrity. Only one sub-characteristic is selected for each characteristic, together with one external metric and an example of acceptance criterion.

3.1.2. *Influence of the Evaluation Levels*

The guidelines for evaluators (ISO/IEC-14598-5, 1998, Annex B, pp. 22-25) provide a similar, though less developed example of the influence of the intended context of use on evaluation choices. Here, a parameter somewhat parallel to integrity is defined, namely *evaluation levels.* These range from A (most critical) to D (least critical) and concern four classes of risks: environment, safety (people), economy (companies) and security (data). Then, the Annex provides a ranking of "evaluation techniques" for each of the six quality characteristics based on the required evaluation level. More demanding techniques should be used for higher levels. For instance, for efficiency, from less to more demanding levels, on should carry out: execution time measurements, benchmark testing, or an analysis of the design to determine the algorithmic complexity.

To conclude, it should be noted that only very specific factors are taken here into account (those associated with risks, in fact), despite the fact that usability studies and the notion of quality in use point out to the strong influence of the context of use on the quality model.

3.2. Relating the Context of Use to the Quality Model

Our main point is that the external evaluator—a person or group in charge of estimating the quality of MT software—must essentially determine a quality model based on the expected context of use of the software, since no unique quality model suits all needs for MT. Our proposal for customizable quality models echoes suggestions in the TEMAA project for a 'Parametrizable Test Bed', in which user profiles determine the weighting of partial scores (TEMAA, 1996, chap. 4).

Our Framework for MT Evaluation in the ISLE project (Femti) is based on the following elements:

1. A classification of the main features defining a context of use: the *user* of the MT system, the *task*, and the nature of the *input* to the system.

2. A classification of the MT software quality characteristics, detailed into hierarchies of sub-characteristics, with internal and/or external attributes (i.e., metrics) at the bottom level. The upper levels must of course match the ISO/IEC 9126 characteristics.

3. A mapping from the first classification to the second, which defines (or at least suggests) the characteristics, sub-characteristics and attributes/metrics that are the most relevant for each context of use.

This broad view of evaluation, which is, by comparison to ISO/IEC, focused on the technical aspect of evaluation, is represented in Figure 1. The taxonomy of contexts of use is in fact closely related to the ISO/IEC quality in use. However, we do not extend our guidelines to quality in use, since this must be measured fully in context, using metrics that have less to do with MT evaluation than with ergonomics and productivity measure.

3.3. A Formal Model of the Context-to-Quality Relation

The following model is built upon the definitions in Section 2.3.3. Remember that the set of all possible attributes for MT software was noted $\{A_1, A_2, \ldots, A_n\}$, and that three stages were identified in the process of evaluation: m_{A_i} (application of metrics), r_{A_i} (rating of measured value), and α (assessment of ratings).

The correspondence described at point (3) above holds between a context of use and the assessment or averaging function α (the function that assigns a greater weight to the attributes relevant to the respective

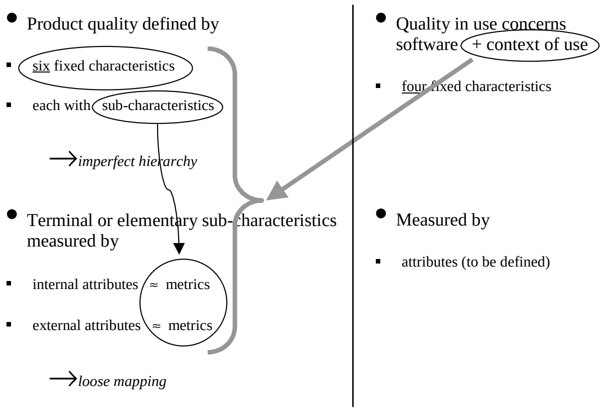
Product quality defined by
six fixed characteristics
each with sub-characteristics
→imperfect hierarchy
Terminal or elementary sub-characteristics measured by
internal attributes ≈ metrics
external attributes ≈ metrics
→loose mapping
Quality in use concerns software + context of use
four fixed characteristics
Measured by
attributes (to be defined)

Figure 1. Proposed systematic link between context of use and quality model

context). Point (3) is thus addressed by providing, for each context of use, the corresponding assessment function. We explain now how this function is computed.

3.3.1. *Definitions*

Since the context of use modulates the assessment function that integrates the ratings of the various measured values of attributes, our goal is to define such a correspondence $\mathcal{M}$:

- correspondence $\mathcal{M}$ between a context C and a quality model (assessment function α):

$$\begin{array}{|l} \mathcal{M} : \mathcal{C} \longrightarrow (\Re^n \longrightarrow \Re) \\ \quad C \longmapsto \alpha_{\mathcal{M}}(C) \end{array}$$

One can imagine, of course, an endless variety of assessment functions α and therefore of mappings $\mathcal{M}$. Hence, we further constrain our formal description by choosing assessment or averaging functions defined by the composition of two functions: a constant averaging function α_0 and a 'linear selection function' $\mathcal{S}_C$ that provides a weight $w_C(A_i)$ for each attribute A_i, depending on the desired context of use C.

a. fixed averaging function α_0:

$$\begin{array}{|l} \alpha_0 : \Re^n \longrightarrow \Re \\ \quad (r_1, r_2, \ldots, r_n) \longmapsto \alpha_0(r_1, r_2, \ldots, r_n) \end{array}$$

b. linear selection function for each context C:

$$\begin{array}{|l} \mathcal{S}_C : \Re^n \longrightarrow \Re^n \end{array}$$

$$
\begin{array}{ll}
| & (r_1, \ldots, r_n) \longmapsto (w_C(A_1) \cdot r_1, \ldots, w_C(A_n) \cdot r_n) \\
| & \text{where } w_C(A_i) \in [0,1],\ \forall\ 1 \leq i \leq n.
\end{array}
$$

Once these are defined, the assessment function for a chosen context of use C is given by $\alpha_0 \circ \mathcal{S}_C$, that is, the selection function followed by the averaging one. Coefficients $w_C(A_i)$ represent the importance of each quality attribute in context of use C, for instance ruling out irrelevant attributes when set at zero value.

3.3.2. *An Algorithm for Specifying Evaluations*

An attempt to put theory into practice shows quickly however that defining each context of use and its selection function in part is a burdensome task. Many contexts of use share a significant number of quality requirements, and only a few attributes are emphasized differently in each of them. Some constraints can then be dropped from the model:

- It is sufficient that the taxonomy of contexts contain a hierarchy of non mutually exclusive characteristics.
- For each context characteristic, weights must be provided for relevant quality attributes (as a 'weighting tuple'), but not for the whole set of quality attributes[5].

Therefore, the assessment function for a given context is constructed by integrating the various weights from the various sub-characteristics that hold for that context. Evaluation specification obeys then the following algorithm:

1. Start with null weights for all quality attributes, $\{A_1, A_2, \ldots, A_n\}$.

2. Go through every branch of the taxonomy of contexts of use, and decide for each node and leaf whether the situation it describes applies to the actual context.

3. If a leaf k applies, then add the weights $v_k(A_i)$ it provides for the relevant arguments (from its weighting tuple). If a node in the context hierarchy carries itself a weighting tuple, then reflect that tuple onto all leaves below that node (i.e., add the weights to all weights below). Therefore, $w'_C(A_i) = \sum_k v_k(A_i)$.

---

[5] For the time being, the taxonomy of contexts of use provides for each sub-characteristic a list of relevant attributes, by order of relevance. In the near future, we will associate numeric weights between 0 and 1 to the attributes.

4. Normalize the final weight list for quality attributes by the highest weight in the list. Therefore, for each A_{i_0}, the associated weight is $w_C(A_{i_0}) = w'_C(A_{i_0})/max_i(w'_C(A_i))$. The weights equal 1 for the essential attribute(s) and smaller values for less important ones. Quite often, many null weights will appear for irrelevant attributes.

The weight list, combined with the assessment function α_0, constitutes the final step of the measurement/rating/assessment process. The second classification that is part of the FEMTI guidelines must be consulted at this point to find the metrics associated to the attributes that have non-null weights. The evaluation can, at this point, be finally specified and executed.

4. Classifications of Contexts and Quality Models: Contents

It is now time to overview the content of our framework, keeping in mind that we cannot but summarize it here, since the full version covers about 30 pages. The first subsection provides an overview of the upper parts of the two classifications (or taxonomies) and contains only the titles of the items without their descriptive content[6]. The second subsection exemplifies in-depth a fragment of our framework.

4.1. GENERAL OVERVIEW

The schema below gives a general view of the contents of our framework. The first part (1.) enumerates non-exclusive characteristics of the context of use, grouped in three complementary subparts (task, user, input). The second part (2.) develops the quality model, its starting point being the six ISO/IEC quality characteristics. The reader will notice that our efforts towards a synthesis have not yet succeeded in unifying internal and external attributes under these six characteristics. As discussed in Section 2.3.2, the link between internal features and external performance is still an object of research for MT systems. So, the internal attributes are structured here in a branch that is separate from the six ISO/IEC characteristics that are measured by external metrics.

1. Specifying the context of use

   1.1 Characteristics of the translation task

---

[6] The Appendix lists all the titles of the entries of the classifications. The contents of each entry are available through the browsable/printable version of our framework at: `http://www.issco.unige.ch/projects/isle/taxonomy2/`.

    - 1.1.1 Assimilation
    - 1.1.2 Dissemination
    - 1.1.3 Communication
  - 1.2 Characteristics of the user of the MT system
    - 1.2.1 Linguistic education
    - 1.2.2 Language proficiency in source language
    - 1.2.3 Language proficiency in target language
    - 1.2.4 Present translation needs
  - 1.3 Input characteristics (author and text)
    - 1.3.1 Document / text type
    - 1.3.2 Author characteristics
    - 1.3.3 Sources of error in the input
      - 1.3.3.1 Intentional error sources
      - 1.3.3.2 Medium-related error sources
      - 1.3.3.3 Performance-related errors

2. Quality characteristics, sub-characteristics and attributes
  - 2.1 System internal characteristics
    - 2.1.1 MT system-specific characteristics (translation process)
    - 2.1.2 Model of translation process (rule-based / example-based / statistical / translation memory)
    - 2.1.3 Linguistic resources and utilities
    - 2.1.4 Characteristics related to the intended mode of use
      - 2.1.4.1 Post-editing or post-translation capacities
      - 2.1.4.2 Pre-editing or pre-translation capacities
      - 2.1.4.3 Vocabulary search
      - 2.1.4.4 User performed dictionary updating
      - 2.1.4.5 Automatic dictionary updating
  - 2.2 System external characteristics
    - 2.2.1 Functionality
      - 2.2.1.1 Suitability (coverage — readability — fluency or style — clarity — terminology)
      - 2.2.1.2 Accuracy
        - Text as a whole (fidelity — comprehensibility — consistency — coherence)
        - Individual sentence level (morphology — syntax: sentence and phrase structure)

        - Types of errors (diction errors — punctuation errors — lexical errors — syntax errors — stylistic errors)
    - 2.2.1.3 Interoperability
    - 2.2.1.4 Compliance
    - 2.2.1.5 Security
- 2.2.2 Reliability
- 2.2.3 Usability
- 2.2.4 Efficiency
    - 2.2.4.1 Time behavior (production time / speed of translation — reading time — revision and post-editing / correction time)
    - 2.2.4.2 Resource behavior
- 2.2.5 Maintainability
- 2.2.6 Portability
- 2.2.7 Cost

These classifications represent a snapshot of the actual state of our proposal, and may be revised under feedback from the community, as explained in Section 5.3.2. Besides, certain branches may be developed in further detail along with the progress of research, and a classification of the *purposes* of evaluation and of the *objects* of evaluation is also under study.

4.2. A Focus on Fluency or "Output Quality"

In this section we focus on what is commonly called *output quality* in MT evaluation, but has been also known as 'fluency', 'correctness', 'intelligibility', 'readability' or 'clarity'—namely, the taxon 2.2.1.1, *Suitability* (under *Functionality*). The idea is to illustrate the richness and complexity of this taxon, to develop it somewhat deeper, and thereby to highlight some of the problems of taxonomization of the features, an issue we take up again in Section 7.

Numerous ways of defining and measuring translation quality have been proposed in the literature. But even with essentially the same definition and measure, the specific metric (measuring system and scale) often differs. For example:

- *Def:* The degree to which words are correctly inflected (for example, tense, number, gender, case, aspect, etc.).
  Source: ALPAC Report (Pierce et al., 1966)
  Metric: Human rating of sentences on a 5-point scale

- *Def:* The syntactic correctness of a sentence: is the translation fluent English?
  Source: DARPA evaluation, Fluency measure (White and O'Connell, 1994)
  Metric: Human rating of sentences on a 5-point scale, normalized to the interval [0,1]
- *Def:* The clarity (understandability) of the sentence.
  Sources: Pfafflin (Van Slype, 1979) and Sinaiko (Van Slype, 1979)
  Metric: Human rating of sentences on a 3-point scale
- *Def:* Comprehensibility reflects the degree to which a complete translation can be understood (whereas the intelligibility is based on the general clarity of translation, whether this is considered in its entirety or by segments out of context).
  Source: Halliday (Van Slype, 1979)
  Metric: Noise test
- *Def:* Subjective evaluation of the degree of comprehensibility and clarity of the translation. In general usage, these terms are considered synonymous. For the purpose of analysis, these terms are differentiated. In the context of MT evaluation, intelligibility refers to the ease with which a mechanical translation can be understood, i.e., how clear is it to the reader? (Van Slype, 1979)
  The following measures and metrics, all from (Van Slype, 1979), can be classified under this somewhat broader definition:
  Source: Leavitt: multiple-choice questionnaire
  Source: Orr: multiple-choice questionnaire
  Source: Sinaiko: knowledge test
  Source: Carroll: rating of sentences on a 9-point scale
  Source: Carroll and Bishop: rating of sentences on a 7-point scale
  Source: Crook and Bishop: rating of sentences on a 7-point scale
  Source: Leavitt: rating of texts on a 9-point scale
  Source: Van Slype: rating of sentences in their context on a 4-point scale
  Source: Vauquois: rating of sentences on a 2-point and 3-point scale

While the abovementioned definitions can all be formulated in terms of syntactic correctness, and hence be applied to individual sentences, or even clauses, out of context, it is clear that translation quality also has an aspect that transcends sentence boundaries. For even when each sentence is grammatical, that does not mean the text as a whole is comprehensible. We are therefore faced with a choice: should we taxonomize multi-sentence readability/comprehensibility together with single-sentence readability/comprehensibility (and give up the measures based on syntactic correctness), or divide them into two taxons (and thereby make the taxonomy more complex)? Given the number of different measures proposed for each case, we decided to split them up. Therefore the above measures of quality can be called:

2.2.1.1.1 **Single-sentence quality:** This test measures the syntactic correctness of the given fragment (a full syntactic unit such as a sentence, clause, or noun phrase). Please rate the fragment on a 5-point scale, where:

- 1: fragment is ungrammatical,
- 2: fragment is grammatical only in places,
- 3: fragment is reasonably grammatical, but has one or two major problems,
- 4: fragment is almost perfect, but has one or two small problems,
- 5: fragment is perfectly grammatical.

while another taxon, its sibling, is:

2.2.1.1.2 **Text-level multi-sentence quality:** This test measures the textual coherence of the given fragment (either a whole text of a connected part of it, and at least three sentences). A text is coherent when the reader can infer how each separate sentence fits into the whole, in other words, what role each sentence plays with respect to the others. Please rate this fragment in terms of a 5-point scale, where:

- 1: fragment is incoherent, nonsense,
- 2: fragment hangs together only in parts, but more than half of it seems disconnected,
- 3: fragment hangs together / coheres reasonably well, but still has one or two major problems,
- 4: fragment is almost perfect, but has one or two small problems,
- 5: fragment is perfectly coherent and each portion fits.

With respect to taxon 2.2.1.1.2, the literature includes the following examples, all from (Van Slype, 1979):

- *Source*: Crook and Bishop: Cloze test (every eighth word is deleted; the evaluator must replace them; the number of correct replacements is the metric).
- *Source*: Halliday: Clozentropy test.

- *Source*: Sinaiko: Multiple-choice questionnaire; a Cloze test (every fifth word); clarity measurement; time measurement.

Even at this level of definitional delicacy it remains unclear exactly what this taxon should include. For example, it might be argued that the DARPA Informativeness Test (White and O'Connell, 1994) belongs in this class as well, because a translation could hardly score well on it unless it was also coherent:

- *Def:* DARPA Informativeness test: Does the translation contain *useful* information?
  Metric: The evaluator performs a comprehension test for each text by answering 6 multiple-choice questions (each with 6 candidate answers). The questions were created by DARPA consultants from expert human translations.

However, no study has ever been done of the correlation between scores of Coherence and Informativeness. We therefore cannot motivate our intuition that Informativeness is but another metric for coherence, and cannot place this test into taxon 2.2.1.1.2.

Separately from these taxons, but also intuitively related to translation quality, stands the question of terminology translation. In many domains, domain-specific terminology (jargon) is not an issue. However, for commercial translation, which is often very domain-oriented and makes heavy use of MT, it can be extremely important. It can also play a major role in low-quality assimilation-oriented translation, in which the user needs only an indication of the main topic of the text; accurately translated terminology may be more useful than polished sentences (Hirschman et al., 2000). We can therefore distinguish another taxon:

> 2.2.1.1.3 **Terminology translation quality:** This test measures how correctly important domain terms are translated. Please highlight each domain term in the input and determine whether it has been translated properly, either by a single term or a description. Please count the percentage of correct cases.

Space limitation preclude us from working through other portions of the taxonomy in more detail. The interested reader is referred to previous papers leading up to this one, including (Hovy, 1999) and the web site related to our framework (`http://www.issco.unige.ch/projects/isle/taxonomy2/`).

5. Dissemination, Use and Update of the Taxonomy

5.1. A Series of Hands-on Exercises

The Evaluation Working Group of the ISLE project concentrates its research on the development of the present framework for MT evaluation, Femti. In order to disseminate results and receive feedback, a series of five workshops was organized: in October 2000 (at AMTA 2000), April 2001 (stand-alone hands-on workshop at ISSCO, Geneva), June 2001 (at NAACL 2001), September 2001 (at MT Summit VIII), May 2002 (at LREC 2002).

All of the workshops included hands-on exercises in MT, supported to a variable extent by the present classifications. The feedback brought to us by the workshops serves directly our goal of developing proposals of concrete usefulness to the whole community. Among the first conclusions drawn from the workshops is the fact that evaluators tend to favor certain parts of the second classification (the quality model)—especially attributes related to the quality of the output text—without paying enough attention to the first classification (the context of use)—for instance to the definition of a user profile. This was however somehow expected, since the links from the first classification to the second were not yet worked out.

It appears globally that the sub-hierarchy related to the “hard problem”, i.e. the quality of output text, should be developed in more detail. Sub-characteristics such as the translation quality for noun phrases attracted steady interest and were further on split into several attributes. Conversely, taking into account the whole range of possible quality characteristics leads to finer-grained evaluations, which bring into light interesting uses of MT systems for which excellent output quality is not the main priority, as pointed out in (Church and Hovy, 1993).

5.2. Examples of Use

There has been considerable continuity between workshops, with results from previous workshops being reported on at subsequent ones, among which a number of interesting examples of using the taxonomy in practice. A very wide range of topics was covered, including the development of new metrics, investigations into possible correlations between metrics, comparisons with the evaluation of human translations, ways to take into account different user needs, novel scenarios both for the evaluation and for the ultimate use of the MT system, and ways to automate MT evaluation. It seems almost invidious to pick out any one of these papers and offer it as an example of how work on the taxonomy proceeds in practice: we hope the other authors will forgive us for choosing a paper which, because of world events, was not actually presented at the September 2001 workshop—see however (Vanni and Miller, 2002).

Vanni and Miller (2001) set out to use the FEMTI framework in an attempt to select and validate metrics which might ultimately be automated, at least in part. They situate their work firmly in a perspective focusing on what MT can be good for, rather than looking at any kind of absolute notion of quality (see (Church and Hovy, 1993) for a first statement of this approach). With this in mind, Vanni and Miller pick out from the ISLE taxonomy a number of features, including coherence, clarity, syntax, morphology and dictionary update/terminology. Metrics and assessment functions (scores) are then developed partly by consulting the taxonomy, partly by consulting the literature, partly by developing original metrics. For example, the coherence measure is based on Mann and Thompson's (1988) Rhetorical Structure Theory (RST): the test involves counting the number of sentences in the text, reading the individual sentences and trying to assign an RST function to each sentence. A sentence where a function can be assigned scores 1, otherwise it scores 0. The final score is obtained by adding up the sentence scores for the text and dividing by the number of sentences in the text. The authors critically discuss the process of validating the measures, suggesting changes they would make on the basis of an initial round of testing.

The next step of their work will involve finding out which FEMTI metrics best correlate with the 'suitability of output for information processing' metric. In exploring that hypothesis, the authors hope also to discover which of the features is most predictive of the usability of MT output in the performance of each specific task.—In any case, we hope that this summary, despite its brevity, gives a taste of how our framework can be used to guide evaluation research, which in turn produce feedback enabling future work.

5.3. MANAGEMENT OF THE FRAMEWORK

For the sake of simplicity, the proposed FEMTI framework can be accessed and browsed through a computer interface. The mechanism that supports this function also ensures that the various nodes and leaves (taxons) of the two classifications are stored in a common XML format, and simplifies considerably the periodic update of the classifications (Popescu-Belis et al., 2001; Hovy et al., 2002). The current version of our framework is visible at `http://www.issco.unige.ch/projects/isle/taxonomy2/`.

5.3.1. *Data Structures*

One of the main advantages of using XML to store the contents of the two classifications (the taxons, the hierarchical structure, the weighting tuples) is the separation between form and content. The classifications are stored in a conceptual format that is quite independent of how they are displayed, this format being defined using an XML DTD (document type definition). Each of the taxons is assigned to a separate computer file. Here is a simplified version of the DTD for a taxon:

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

For the time being, the same taxon structure is used in both classifications (context of use and quality characteristics), despite their different roles in the specification of evaluations. The `how-to-measure` fields of the taxons in the first classification point towards quality characteristics taxons in the second classification, while the `how-to-measure` fields of the quality attributes (in the second classification) point towards metrics.

Future improvements of this mechanism include the adoption of different DTDs for taxons in the first *vs.* the second classification, and the use of the XPointer standard to encode the links from the first classification to the second one. Furthermore, the possibility of declaring in the DTD all the MT quality attributes is under study, in order to formally encode in the 'context of use' taxons the weighting tuples for the relevant quality attributes (noted $v_k(A_i)$ in Section 3.3). The overall goal is to automate the evaluation specification through an interface that allows the evaluators to select the relevant characteristics of the context, then proposes to them a set of relevant quality characteristics (including attributes and metrics) with the proper assessment function (weighting tuple), a proposal that can be modified and finally validated by the evaluators to obtain the draft specification of their evaluation.

5.3.2. *Life-cycle of the Framework*

The automatic generation of a readable version of our framework relies essentially on the XSL mechanism (eXtensible Stylesheet Language). Stylesheets allow generation of HTML files for each taxon (browsable version of FEMTI), or alternatively of a single HTML file for the whole

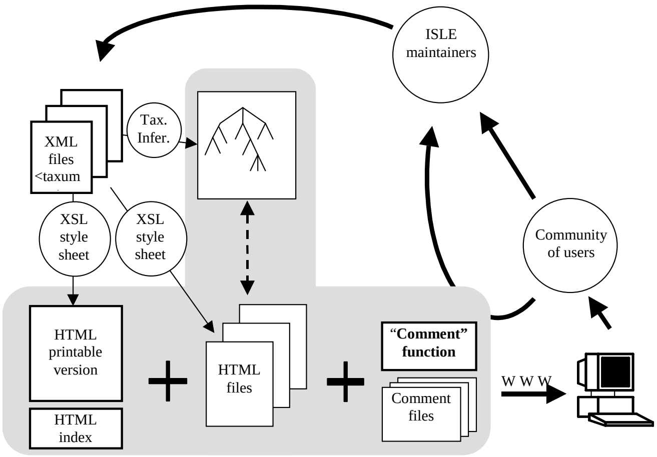

Figure 2. Life-cycle of the Framework for MT Evaluation in ISLE

framework (printer-friendly version), both of them being made available through a web server. All the information necessary to these transformations is contained in the individual `<taxon>` files, the hierarchical structure of the classifications being encoded using indexes and pointers. External support files are added to the output HTML files in order to build a user-friendly, informative web site (the support files are not modified for each update of the framework). Finally, the mechanism that allows evaluators to parametrize the MT quality model depending on the context of use is currently under development.

The separation between the FEMTI content files (taxons) and the XML/XSL formatting mechanism facilitates the update of both form and content, and enables us to receive and log comments. The life-cycle of our framework is summarized in Figure 2. Starting with the XML files for the individual taxons, the XSL stylesheets and other scripts generate the web site through which FEMTI is consulted (browsed or printed). A 'comment function' (form plus button) is automatically embedded in the HTML files, allowing the users to post comments upon individual taxons or upon the whole classification. These comments and those received directly, e.g. in the series of workshops organized through the ISLE project, are gradually fed back to the taxon files. Once these suggestions have been validated, a new version of the HTML files is then generated.

6. Studies Towards the Refinement of the Taxonomy

The taxonomy forms but the first step in a larger program—it lists the essential parameters of importance to MT evaluation. But for a comprehensive and systematic understanding of the problem, one also has to know the nature and results of the actual evaluation measures used. In our current work, a primary focus is the analysis of the measures and metrics: their variation, correlation, expected deviation, reliability, cost to perform, etc. This section outlines first a theoretical framework featuring coherence criteria for the metrics, then lists the (unfortunately very few) examples from previous research.

6.1. Coherence Criteria for Evaluation Metrics

We have previously proposed coherence criteria for NLP evaluation metrics in an Eagles based framework (Popescu-Belis, 1999a; Popescu-Belis, 1999b). The main goal was to propose criteria enabling evaluators to choose the most suitable metric for a given attribute and to help interpret the measures. One of the main hypotheses was that a given quality attribute could be measured by running the system on a relevant set of data, then measuring the quality level for each of the system's responses. Therefore, the criteria applied to metrics were defined in terms of the distance between the actual and the desired responses. In the ISO/IEC vocabulary these are external metrics. We extend now these considerations to the general case of a metric that estimates the quality of a given attribute without necessarily comparing the system's response with a correct one.

A metric for a given attribute A_i is a function from an abstract quality space onto a numeric interval—say [0%, 100%] in the rest of this section (scaling can be used to bring measured values to such an interval). With respect to definition *a.* in Section 2.3.3, the metric quantifies a system's position in the quality space for A_i. The goal of evaluators using a metric (plus rating) is to quantify the quality level. So, before analyzing metrics, evaluators must poll the experts to get an idea of what the best and the worst quality levels are for A_i.

It is often easy to define the best quality, but there are at least two kinds of very poor quality levels: (a) the worst imaginable ones (which a system may rarely actually descend to) and (b) the levels attained by simplistic or baseline systems. For instance, if the capacity to translate polysemous words is evaluated, a system that always outputs the most frequent sense of source words does far better than the worst possible system (the one that *always* gets it wrong) or than a random system. Once these bounds are identified, the following coherence criteria should be tested for:

UL—upper limit A metric must reach 1 for perfect quality (the respective attribute A_i) and (reciprocally) only reach 1 when the quality is perfect.

LL—lower limit A metric must reach 0 for the worst possible quality (of the respective attribute A_i) and only reach 0 when the quality is extremely low. As noted above, since it is not easy to identify the situations of lowest quality, the following reformulations of the LL criterion are easier to analyze:

LL–1 All the translations (or systems) receiving a 0 score must indeed have very poor quality. Since it is obviously difficult to identify such cases, this criterion can be studied using (counter-)examples.

LL–2 Reciprocally, all the worst quality cases must receive a 0 score. This criterion is studied by finding examples of poor quality systems (for a given attribute) and testing whether they receive low scores.

LL–2′ A simpler necessary condition for LL–2 is: the lowest possible scores of a metric must be close or equal to 0. If this is not true, then LL–2 cannot be true either.

M — monotonicity A metric must be monotonic, that is, if the quality of system A is higher than that of system B, then the score of A must be higher than the score of B.

One should note that it is difficult to *prove* that a metric does satisfy these coherence criteria, and much easier to use (counter-)examples to criticize a measure on the basis of these criteria.

Finally, to compare two metrics, we can say that m_1 is more severe (or less lenient) than m_2 if it yields lower scores for each possible quality level. This is not a coherence criterion, but a comparison criterion between two metrics that should help evaluators select the most appropriate metric according to the expected quality level. Of course, not every two metrics of the same attribute can be compared. Also, since the rating function can affect the severity of a metric, it is rather the rating functions that are chosen on the basis of estimated severity than the other way round. Severity (and its inverse, leniency) are intrinsically comparative concepts, but they can be used in an absolute way, meaning “more severe (resp., lenient) than most of other metrics”.

6.2. Analyzing the Behavior of Metrics

As described in Section 4.2, the same MT quality characteristic may be grasped using several attributes, and each attribute may be measured through various metrics. This uncomfortable state of affairs calls for investigation. If it should turn out, for example, that one specific attribute for single-sentence quality correlates perfectly with human judgments, subsumes most or all of the other proposed attributes, can be expressed easily into one or more metrics, and is cheap to apply, we should have no reason to look further: that quality (sub-)characteristic or attribute would be settled.

The full list of desiderata for a metric is not immediately clear. We can however list some obvious ones. The metric:

- must be easy to define: clear and intuitive
- must correlate well with human judgments under all conditions, genres, domains, etc.
- must be reliable, exhibiting as little variance as possible across evaluators, or for equivalent inputs
- must be cheap to prepare (i.e., not require a great deal of human effort to prepare training data or ideal examples)
- must be cheap to apply
- should be automated if possible

Unexpectedly, the literature contains rather few methodological studies of the kind we need. Many people have applied their own evaluations, but few have bothered to try someone else's, and then to correlate the two.

However, there are some advances. In recent promising work using the DARPA 1994 evaluation results (White and O'Connell, 1994), White and Forner have studied the correlation between intelligibility (i.e, syntactic fluency) and fidelity (White, 2001) and between fidelity and noun compound handling by the system (Forner and White, 2001). As one would expect with measures focusing on aspects as different as syntax and semantics, they find some correlation, but not a simple one. Studies of the relation between the automatic Bleu scores and the judgments of two sets of 10 human judges (one set monolingual, the other bilingual), over the same texts (some by human, some by machine), show a very high level of agreement, namely correlation coefficients of 0.99 with the monolingual group and 0.96 with the bilingual group (Papineni et al., 2001). These results support the validity of Bleu scores and should be generalized to other automated methods.

Such studies are important. Also important are studies of other aspects of the metrics. Most careful evaluations report on inter-evaluator agreement, which can differ quite widely. Although it is well known by psychologists that the way one formulates instructions can have a major effect on subjects' behavior, we have no guidelines for formulating the instructions for evaluators, and no idea how variations would affect systems' scores. Similarly, we do not know whether a 3-point scale is more effective than a 5-point scale or a 7-point scale; experiments are needed to determine the optimal point between inter-evaluator consistency (higher on a shorter scale) and evaluation informativeness (higher on a longer scale).

Another very important issue for evaluation is the number of measure points (e.g., texts translated by a system) required by each metric before the evaluation can be trusted. Again, here, all that is available are the confidence levels of past evaluation studies.

In the ISLE research we are now embarking on the design of a program that will help address these questions. Our very ambitious goal is to know, for each taxon in the classification of quality characteristics, which attributes are most relevant, which metric(s) are most appropriate to measure them, how much work and cost is involved in applying each metric, and what final level of system score should be considered acceptable (or not) given the purpose of the evaluation. Armed with this knowledge, a would-be evaluator would be able to make a much more informed selection of what to evaluate and how to go about it.

7. Further Developments and Conclusion

A general theme running throughout this paper is that MT evaluation is simply a special, although rather complex, case of software evaluation in general. An obvious question then is whether the work described here can be extended to other fields. Some previous experience has shown that it applies relatively straightforwardly to some domains, for example, dialogue systems in a specific context of use. It is our belief that the basic ISO/IEC notion of building a quality model in relation to a desired context of use, then associating appropriate metrics to it, should carry over to almost any application. Where we are less confident is in the definition of user needs outside specific contexts. With the applications considered so far, it has been possible to imagine users or classes of users and describe their needs fairly exhaustively. As the system to be evaluated grows more complex, this may well become less realistic. For an application like data mining or information management, the uses of such tools are potentially almost infinite. Trying to imagine them all and to draw up a descriptive scheme as we are doing for MT systems is likely to prove impossible. Nonetheless, it should still prove possible to describe specific needs in a specific context of use, and derive from such a description the choice of quality characteristics and metrics that are relevant.

It can also be appreciated that building classifications of features is an arduous task, made more difficult by the fact that few external criteria for correctness exist. It is easy to think of features and to create taxonomies; we therefore have several suggestions for our classifications. It is unfortunately very difficult to validate the correctness of one's decisions, hence to justify one's intuitions; it it thus possible that we always have multiple copies of our framework, each one reflecting the author's own experiences and biases. As with semantic ontology building, this is probably a fact we may have to live with.

We therefore explicitly do not claim here that the FEMTI framework is correct, complete, or not subject to change. We expect it to grow, to become more refined, and to be the subject of discussion and disagreement—that is the only way in which it will show its relevance.

Nonetheless, while it is possible to continue refining the FEMTI framework, collecting additional references, and classifying additional metrics, we feel that the most pressing work is only now being started. Our framework is but the first step toward a more comprehensive and systematic understanding of MT evaluation in all its complexity. As discussed in Section 6, the work needed to make it truly useful is a careful study of the various metrics, of their individual strengths and weaknesses, and especially of their correlations. If we are ever to realize the wish of knowing precisely which minimal set of evaluations to perform in each situation, we have to know which metrics are the most central, trustworthy, and cost-effective. This we can only determine by a dedicated program of systematic comparison.

The dream of a magic test that makes everything easy—preferably an automated process—always remains. One of the latest candidates, recently proposed by (Papineni et al., 2001), seems to have these desirable characteristics, which prompted its use in NIST's recent evaluation campaign (see Section 2.1). Should it be true that the BLEU metric correlates very highly with human judgments about a certain quality characteristic, and that it really requires only a handful of reference (expert) translations, then we will be spared much work. But we will not be done. For although the existence of a quick and cheap evaluation measure is enough for many people (system developers, for instance), it still does not cover more than a small portion of the taxonomy. All the other aspects of machine translation that people have wished to measure in the past remain to be measured.

Appendix

A. Developed View of the Two Classifications

This is the organization at the time of writing of the two FEMTI classifications, quoting here for each taxon only its title, but not its contents (cf. more explanations in Section 4).

1. Specifiying the context of use
    - 1.1 Characteristics of the translation task
        - 1.1.1 Assimilation
            - 1.1.1.1 Document routing / sorting
            - 1.1.1.2 Information extraction / summarization
        - 1.1.2 Dissemination
            - 1.1.2.1 Internal / in-house publication
                - Routine
                - Experimental / research
            - 1.1.2.2 External publication
                - Single-client
                - Multi-client
        - 1.1.3 Communication
            - 1.1.3.1 Interactive
            - 1.1.3.2 Delayed
    - 1.2 Characteristics of the user of the MT system
        - 1.2.1 Linguistic education
        - 1.2.2 Language proficiency in source language
        - 1.2.3 Language proficiency in target language
        - 1.2.4 Present translation needs
            - 1.2.4.1 Quantity of translation
            - 1.2.4.2 Number of personnel
            - 1.2.4.3 Time allowed for translation
    - 1.3 Input characteristics (author and text)
        - 1.3.1 Document / text type
            - 1.3.1.1 Genre
            - 1.3.1.2 Domain / field of application
        - 1.3.2 Author characteristics
            - 1.3.2.1 Proficiency in source language
            - 1.3.2.2 Professional training
        - 1.3.3 Characteristics related to sources of error in the input
            - 1.3.3.1 Intentional error sources

      - 1.3.3.2 Medium related error sources (speech recognition, OCR, etc.)
      - 1.3.3.3 Performance related errors

2. Quality characteristics, sub-characteristics and attributes
   - 2.1 System internal characteristics
     - 2.1.1 MT system-specific characteristics (translation process)
       - 2.1.1.1 Language specific characteristics
       - 2.1.1.2 Internal linguistic sophistication and level of processing
     - 2.1.2 Model of translation process (rule-based / example-based / statistical / translation memory)
     - 2.1.3 Linguistic resources and utilities
       - 2.1.3.1 Languages
       - 2.1.3.2 Dictionaries
       - 2.1.3.3 Word lists, glossaries, parallel corpora
       - 2.1.3.4 Grammars
     - 2.1.4 Characteristics related to the intended mode of use
       - 2.1.4.1 Post-editing or post-translation capacities
       - 2.1.4.2 Pre-editing or pre-translation capacities
       - 2.1.4.3 Vocabulary search (unknown word identification)
       - 2.1.4.4 User performed dictionary updating
       - 2.1.4.5 Automatic dictionary updating
   - 2.2 System external characteristics
     - 2.2.1 Functionality
       - 2.2.1.1 Suitability
         - – Coverage
           - Coverage of cross-language phenomena
           - Coverage of corpus based problems
         - – Readability
         - – Fluency / style
         - – Clarity
         - – Terminology
         - – Utility of output
       - 2.2.1.2 Accuracy
         - – Text as a whole
           - Fidelity
           - Comprehension / comprehensibility
           - Consistency
           - Coherence
         - – Individual sentence level
           - Morphology

        - Syntax: [phrase and sentence structure
      - Types of errors
        - Diction errors
        - Punctuation errors
        - Lexical errors
        - Syntax errors
        - Stylistic errors

2.2.1.3 Interoperability
2.2.1.4 Compliance
2.2.1.5 Security
2.2.2 Reliability
2.2.2.1 Maturity
2.2.2.2 Fault tolerance
2.2.2.3 Crashing frequency
2.2.2.4 Recoverability
2.2.3 Usability
2.2.3.1 Understandability
2.2.3.2 Learnability
2.2.3.3 Operability
2.2.3.4 Documentation
2.2.4 Efficiency
2.2.4.1 Time behavior

- Production time / speed of translation
- Reading time
- Revision and post-editing / correction time

2.2.4.2 Resource behavior
2.2.5 Maintainability
2.2.5.1 Analyzability
2.2.5.2 Changeability

- Ease of upgrading multilingual aspects of system
- Improveability
- Ease of dictionary updating
- Ease of modifying grammar rules

2.2.5.3 Stability
2.2.5.4 Testability
2.2.6 Portability
2.2.6.1 Adaptability
2.2.6.2 Installability
2.2.6.3 Conformance
2.2.6.4 Replaceability

- 2.2.7 Cost
  - 2.2.7.1 Introduction cost
  - 2.2.7.2 Maintenance cost
  - 2.2.7.3 Other costs

References

AMTA: 1992, 'MT Evaluation: Basis for Future Directions (Proceedings of a workshop held in San Diego)'. Technical report, Association for Machine Translation in the Americas (AMTA).

Blasband, M.: 1999, 'Practice of Validation: The ARISE Application of the EAGLES Framework'. In: *EELS (European Evaluation of Language Systems) Conference*. Hoevelaken, The Netherlands.

Canelli, M., D. Grasso, and M. King: 2000, 'Methods and Metrics for the Evaluation of Dictation Systems: A Case Study'. In: *Second International Conference on Language Resources and Evaluation (LREC 2000)*, Vol. 3. Athens, Greece, pp. 1325–1331.

Church, K. W. and E. H. Hovy: 1993, 'Good Applications for Crummy MT'. *Machine Translation* **8**, 239–258.

Daly-Jones, O., N. Bevan, and C. Thomas (eds.): 1999, *Handbook of User-Centred Design: INUSE 6.2*. http://www.ejeisa.com/nectar/inuse.

Doyon, J., K. Taylor, and J. S. White: 1998, 'The DARPA Machine Translation Evaluation Methodology: Past and Present'. In: *Proceedings of the AMTA Conference*. Philadelphia, PA.

EAGLES-Evaluation-Workgroup: 1996, 'EAGLES Evaluation of Natural Language Processing Systems'. Final report, Center for Sprogteknologi, Denmark.

Flanagan, M.: 1994, 'Error Classification for MT Evaluation'. In: *Proceedings of the AMTA Conference*. Columbia, Maryland.

Forner, M. and J. S. White: 2001, 'Predicting MT Fidelity from Noun-Compound Handling'. In: *Workshop on MT Evaluation "Who did what to whom?" at Mt Summit VIII*. Santiago de Compostela, Spain.

Hirschman, L., F. Reeder, J. Burger, and K. Miller: 2000, 'Name Translation as a Machine Translation Evaluation Task'. In: *Workshop on MT Evaluation at the LREC 2000 Conference*. Athens, Greece.

Hovy, E., M. King, and A. Popescu-Belis: 2002, 'Computer-Aided Specification of Quality Models for MT Evaluation'. In: *Third International Conference on Language Resources and Evaluation (LREC 2002)*, Vol. 4. Las Palmas de Gran Canaria, Spain, pp. 1239–1246, ELRA.

Hovy, E. H.: 1999, 'Toward Finely Differentiated Evaluation Metrics for Machine Translation'. In: *EAGLES Workshop on Standards and Evaluation*. Pisa, Italy.

Infoshop: 1999, 'Language Translations: World Market Overview, Current Developments and Competitive Assessment'. Technical report, Infoshop Japan, Global Information Inc., Kawasaki, Japan, http://www.infoshop-japan.com/study/ab3365_languagetranslation_toc.html.

ISO/IEC-14598-1: 1999, *ISO/IEC 14598-1:1999 (E) — Information technology — Software product evaluation — Part 1: General overview*. Geneva: International Organization for Standardization & International Electrotechnical Commission.

ISO/IEC-14598-2: 2000, *ISO/IEC 14598-2:2000 (E) — Software engineering — Product evaluation — Part 2: Planning and management*. Geneva: International Organization for Standardization & International Electrotechnical Commission.

ISO/IEC-14598-3: 2000, *ISO/IEC 14598-3:2000 (E) — Software engineering — Product evaluation — Part 3: Process for developers*. Geneva: International Organization for Standardization & International Electrotechnical Commission.

ISO/IEC-14598-4: 1999, *ISO/IEC 14598-4:1999 (E) — Software engineering — Product evaluation — Part 4: Process for acquirers*. Geneva: International Organization for Standardization & International Electrotechnical Commission.

ISO/IEC-14598-5: 1998, *ISO/IEC 14598-5:1998 (E) — Software engineering — Product evaluation — Part 5: Process for evaluators*. Geneva: International Organization for Standardization & International Electrotechnical Commission.

ISO/IEC-14598-6: 2001, *ISO/IEC 14598-6:2001 (E) — Software engineering — Product evaluation — Part 6: Documentation of evaluation modules*. Geneva: International Organization for Standardization & International Electrotechnical Commission.

ISO/IEC-9126: 1991, *ISO/IEC 9126:1991 (E) — Information Technology — Software Product Evaluation — Quality Characteristics and Guidelines for Their Use*. Geneva: International Organization for Standardization & International Electrotechnical Commission.

ISO/IEC-9126-1: 2001, *ISO/IEC 9126-1:2001 (E) — Software engineering — Product quality — Part 1: Quality model*. Geneva: International Organization for Standardization & International Electrotechnical Commission.

Kay, M.: 1980, 'The Proper Place of Men and Machines in Language Translation'. Research Report CSL-80-11, XEROX PARC.

King, M. and K. Falkedal: 1990, 'Using Test Suites in Evaluation of Machine Translation Systems'. In: *18th Coling Conference*, Vol. 2. Helsinki, Finland.

Lehrberger, J. and L. Bourbeau: 1988, *Machine Translation: Linguistic Characteristics of MT Systems and General Methodology of Evaluation*, Lingvisticæ Investigationes Supplementa 15. Amsterdam: John Benjamins Press.

Mann, W. C. and S. A. Thompson: 1988, 'Rhetorical Structure Theory: A Theory of Text Organization'. *Text* **8**(3), 243–281.

Mason, J. and A. Rinsche: 1995, 'Translation Technology Products'. Report, OVUM Ltd.

Nagao, M.: 1989, 'A Japanese View on Machine Translation in Light of the Considerations and Recommendations Reported by ALPAC, U.S.A.'. Technical report, Japan Electronic Industry Development Association (JEIDA).

Nießen, S., F. J. Och, G. Leusch, and H. Ney: 2000, 'An Evaluation Tool for Machine Translation: Fast Evaluation for MT Research'. In: *Second International Conference on Language Resources and Evaluation (LREC 2000)*, Vol. 1. Athens, Greece, pp. 39–45, ELRA.

Nomura, H.: 1992, 'JEIDA Methodology and Criteria on Machine Translation Evaluation'. Technical report, Japan Electronic Industry Development Association (JEIDA).

Nomura, H. and J. Isahara: 1992, 'The JEIDA Report on machine Translation". In: *Workshop on MT Evaluation: Basis for Future Directions*. San Diego, CA, Association for Machine Translation in the Americas (AMTA).

Papineni, K., S. Roukos, T. Ward, and W.-J. Zhu: 2001, 'BLEU: a Method for Automatic Evaluation of Machine Translation'. Research Report, Computer Science RC22176 (W0109-022), IBM Research Division, T.J.Watson Research Center, http://domino.watson.ibm.com/library/Cyberdig.nsf/home.

Pierce, J., J. Carroll, E. Hamp, D. Hays, C. Hockett, A. Oettinger, and A. Perlis: 1966, 'Computers in Translation and Linguistics (ALPAC Report)'. report 1416, National Academy of Sciences / National Research Council.

Popescu-Belis, A.: 1999a, 'Evaluation of natural language processing systems: a model for coherence verification of quality measures'. In: M. Blasband and P. Paroubek (eds.): *A Blueprint for a General Infrastructure for Natural Language Processing Systems Evaluation Using Semi-Automatic Quantitative Black Box Approach in a Multilingual Environment.* ELSE Project LE4-8340 (Evaluation in Language and Speech Engineering).

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

Popescu-Belis, A., S. Manzi, and M. King: 2001, 'Towards a Two-stage Taxonomy for Machine Translation Evaluation'. In: *Workshop on MT Evaluation "Who did what to whom?" at Mt Summit VIII.* Santiago de Compostela, Spain.

Rajman, M. and A. Hartley: 2002, 'Automatic Ranking of MT Systems'. In: *Third International Conference on Language Resources and Evaluation (LREC 2002)*, Vol. 4. Las Palmas de Gran Canaria, Spain, pp. 1247–1253, ELRA.

Sparck-Jones, K. and J. R. Galliers: 1993, 'Evaluating Natural Language Processing Systems'. Technical Report 291, University of Cambridge Computer Laboratory.

Sparck-Jones, K. and J. R. Galliers: 1996, *Evaluating Natural Language Processing Systems: An Analysis and Review*, Lecture Notes in Artificial Intelligence 1083. Berlin / New York: Springer-Verlag.

TEMAA: 1996, 'TEMAA Final Report'. Technical Report LRE-62-070 (March 1996), Center fo Sprogteknologi, Copenhagen, Danemark, http://www.cst.ku.dk/projects/temaa/D16/d16exp.html.

Thompson, H. S. (ed.): 1992, *The Strategic Role of Evaluation in Natural Language Processing and Speech Technology (Record of a workshop sponsored by DANDI, ELSNET and HCRC).* University of Edinburgh (Technical Report, May 1992).

Tomita, M.: 1992, 'Application of the TOEFL Test to the Evaluation of Japanese-English MT'. In: *Proceedings of MT Evaluation Workshop at the AAMT Conference.*

Vanni, M. and K. Miller: 2001, 'Scoring Methods for Multi-Dimensional Measurement of Machine Translation Quality'. In: *Workshop on MT Evaluation "Who did what to whom?" at Mt Summit VIII.* Santiago de Compostela, Spain.

Vanni, M. and K. Miller: 2002, 'Scaling the ISLE Framework: Use of Existing Corpus Resources for Validation of MT Metrics across Languages'. In: *Third International Conference on Language Resources and Evaluation (LREC 2002)*, Vol. 4. Las Palmas de Gran Canaria, Spain, pp. 1254–1262, ELRA.

Van Slype, G.: 1979, 'Critical Study of Methods for Evaluating the Quality of Machine Translation'. Technical Report BR 19142, European Commission / Directorate for General Scientific and Technical Information Management (DG XIII), http://issco-www.unige.ch/projects/isle/van-slype.pdf.

White, J. S.: 2001, 'Predicting Intelligibility from Fidelity in MT Evaluation'. In: *Workshop on MT Evaluation "Who did what to whom?" at Mt Summit VIII.* Santiago de Compostela, Spain.

White, J. S. and T. O'Connell: 1992-1994, 'ARPA Workshops on Machine Translation: Series of four workshops on comparative evaluation'. Technical report, Litton PRC Inc., McLean, VA.

White, J. S. and K. B. Taylor: 1998, 'A Task-Oriented Evaluation Metric for Machine Translation'. In: *First International Conference on Language Resources and Evaluation (LREC 1998)*, Vol. 1. Granada, Spain, pp. 21–25, ELRA.