Machine Translation **17:** 43–75, 2002.

Principles of Context-Based Machine Translation Evaluation

EDUARD HOVY
USC Information Sciences Institute, 4676 Admiralty Way, Marina del Rey, CA 90292-6695, USA
E-mail: hovy@isi.edu

MARGARET KING and ANDREI POPESCU-BELIS
University of Geneva, ISSCO/TIM/ETI, 40 Bvd. du Pont d'Arve, 1211 Geneva 4, Switzerland
E-mail: {Andrei.Popescu-Belis,Margaret.King}@issco.unige.ch

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 characterised by overall complexity and local difficulties. In this article, we first summarise 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 of two interrelated classifications or taxonomies. The first classification enables evaluators to define an intended context of use, while the links to the second classification generate a relevant quality model (quality characteristics and metrics) for the respective context. The second classification provides definitions of various metrics used by the community. Further on, as part of ongoing, long-term research, we explain how metrics are analyzed, first from the general point of view of "meta-evaluation", then focusing on examples. Finally, we show how consensus towards the present framework is sought for, and how feedback from the community is taken into account in the FEMTI life-cycle.

Key words: MT evaluation, quality models, evaluation metrics, context-based evaluation

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.[1] 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 Church and Hovy (1993) have argued, 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 standardised 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 a Framework for the Evaluation of MT (FEMTI) which enables evaluators to parameterise the quality model (the set of desired system qualities and their metrics) based on the intended context of use. Building upon previous work, our proposal also aims at applying to the domain of MT the standards for software evaluation jointly proposed by the International Organization for Standardization (ISO) together with the International Electrotechnical Commission (IEC). 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 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, then moving on to natural language processing (NLP) evaluation, and ending with the ISO/IEC standards. Next, 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 review the contents of our classifications, with a focus on output quality. Proposals for the dissemination and update of this work are given in Section 5, while Section 6 provides long-term guidelines for 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 comparison 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 standardisation and interoperability of evaluation schemes. Indeed, not only are the initiatives for standardisation more developed at the general level of software engineering (cf. 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 (Section 2.1). 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* (specialisation of the system to the domains of interest). In fact, as discussed by Church and Hovy (1993), for some real-world applications quality may even 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, 1994a, b), 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, 1994a, b). 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 (see next section), the Japanese JEIDA study of 1992 (Nomura, 1992; Nomura and Isahara, 1992) identified two sets of 14 parameters each: one set characterises the desired context of use of an MT system, and the other characterises 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, customisability, application to total translation process, language coverage, terminology building, documentation, and other criteria.

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 (Test of English as a Foreign Language) 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. Another study has made use of the TOEIC comprehension task to assess the utility of English–Japanese MT output to various types of users (Fuji et al., 2001). A more general view of task-based evaluations is provided in Taylor and White (1998) and White and Taylor (1998).

The variety of MT evaluations is enormous, from the highly influential ALPAC Report (ALPAC, 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, 1994a, b) and beyond. A recent evaluation campaign (2001–2002) organised by the US National Institute of Standards and Technology (NIST) and making use of both human and automated metrics is described in several reports available online.[2] Several recent proposals towards the automation of MT evaluation, in addition to Papineni et al. (2001) and Thompson (1992), have been made in Niessen et al. (2000) and 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 have been gathered by Lehrberger and Bourbeau (1988) and King and Falkedal (1990). Beyond the influential papers by Kay (1980) and by Nagao (1989), the pre-AMTA workshop on evaluation (Vasconcellos, 1992) and a special issue of this journal (Arnold et al., 1993) contain a useful set of papers.

2.2. THE EAGLES GUIDELINES FOR NLP SOFTWARE EVALUATION

The European EAGLES initiative (Expert Advisory Group on Language Engineering Standards) came into being as an attempt to create standards for language engineering. The initiative was 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 resource transfer across projects, product development, and different applications. The first areas to be attacked in the first phase of the initiative (1993–1995) 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), the EAGLES Evaluation Work Group (EWG) 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 measurements to be combined in different ways (EAGLES, 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 (EAGLES, 1999). 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. Also during the second phase, the EWG came into closer contact with the ISO/IEC work on the evaluation of software in general.

When the ISLE project 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 with the same formalism a taxonomisation of user needs, along the lines suggested by the JEIDA study (Nomura, 1992). The EWG 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.[3]

2.3. THE ISO/IEC STANDARDS FOR SOFTWARE EVALUATION

In this section, we summarise the ISO/IEC standards for software quality evaluation, a series that has become more and more detailed. We propose first an overview of the series, then summarise the main points that are relevant to our framework, with a focus on the three-stage evaluation process.

2.3.1. *A Growing Set of Standards*

ISO together with the IEC have initiated in the past decade an important effort towards the standardisation of software evaluation. The ISO/IEC 9126 standard (ISO/IEC, 1991) appears as a milestone: It defined 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, 1991: 2).

The 1991 version of ISO/IEC 9126 was a prelude to more detailed developments in evaluation standardisation, on one side concerning the definition of quality models and associated metrics, and on the other side concerning the organisation 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, 1999a) 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, made of four inter-related standards: software quality models (ISO/IEC, 2001b), 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, 2000a), on its documentation (ISO/IEC, 2001a), and its application to developers (ISO/IEC, 2000b), to acquirers (ISO/IEC, 1999b) and to evaluators (ISO/IEC, 1998). Very briefly, in the last three documents, different aspects are emphasised 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 decontextualised point of view. We will adopt the point of view of the evaluators in order to summarise the ISO/IEC 9126 and 14598 framework regarding the key issue of defining a quality model.

2.3.2. *Definition of a Quality Model*

According to ISO/IEC 14598-1 (1999a: 12, fig. 4), the software life-cycle starts with the analysis of the 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*. During the design and development phase software quality becomes an *internal* matter related to the characteristics of the software 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 its origins, *quality in use* is the extent to which the software really helps users fulfil their tasks (ISO/IEC, 2001b: 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 valid system that sells books online, based on the requirements that the system manage 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 100 languages. Research in MT (and in other branches of NLP and Artificial Intelligence) must deal with the specification-to-design phase empirically, with the result that evaluators have then to define their own contexts of use and select external quality requirements. Therefore, the relation between external and internal qualities is quite loose in the case of MT.

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

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

Already present in ISO/IEC (1991), these characteristics have been refined in the more recent version of the standard, thanks to a loose hierarchy of subcharacteristics (still domain-independent) that may contain some overlapping (ISO/IEC, 2001b, A.1.1: 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, 1999a, emphases original).

We will adopt in this paper the ISO/IEC term "metric", even though the scoring methods do not always have the mathematical properties of a metric.[4] The terms "measure" or "measurement" will be used, in agreement with ISO/IEC, for the application of a metric to a translation or to a system; several (alternative) metrics can often be used to assign a level to a quality attribute.

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) will certainly differ. 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, 2001b, A.1.1: 13). As noted above, such a connection is generally absent in the case of MT software, despite examples such as the relation between the "dictionary size" (internal quality) and the "proportion of (un)translated words" (external quality).

Quality in use results from four characteristics (ISO/IEC, 2001b: 11).

- effectiveness
- productivity
- safety
- satisfaction

These can only be measured in the operating environment of the software, and seem less prone to standardisation. 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*

The ISO/IEC standards also outline the evaluation process, generalizing a proposal already present in the first ISO/IEC 9126. The five consecutive phases are emphasised differently according to whom the initiators of the evaluation are, developers, acquirers, or evaluators.

1. Establish the quality requirements, i.e. the list of required quality characteristics.
2. Specify the evaluation, i.e. the measurements and their mapping to the requirements.
3. Design the evaluation, producing the evaluation plan, i.e. the documentation of the procedures used to perform measurements.
4. Execute the evaluation, producing a draft evaluation report.
5. 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, three stages must be distinguished in the specification, design, and execution of an evaluation.[5] 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: The integration of measurements towards a *final score* is not an ISO/IEC priority, unlike the case of 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). The three-stage distinction above, advocated also by EAGLES (1996, 1999), seems to us particularly useful regarding the concrete evaluations of systems.

2.3.4. *Formal Definition of the Stages*

More formally, let S be a system for which several attributes must be evaluated, say $A_1, A_2, \ldots, A_n$ (Popescu-Belis, 1999a, b). First, the system is measured with a metric m_{A_i} for each attribute, generating a value on a scale that depends on the metric m_{A_i}, not necessarily tailored to reflect a quality value. If the set of all systems is noted Σ and the scale associated to m_{A_i} is the interval $[\inf(m_{A_i}), \sup(m_{A_i})]$, we define application of a metric as (1).

$$\begin{aligned} m_{A_i} : \ & \Sigma \longrightarrow [\inf(m_{A_i}), \sup(m_{A_i})] \\ & S \longmapsto m_{A_i}(S) \end{aligned} \tag{1}$$

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 normalises 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 set (for example, a numeric scale). The mapping between the measured values and the ratings reflects now the human judgment of an attribute's quality.

The rating of the measured value is given by (2).

$$\begin{aligned} r_{A_i} : \ & [\inf(m_{A_i}), \sup(m_{A_i})] \longrightarrow \{r_1, r_2, \ldots, r_p\} \\ & m_{A_i}(S) \longmapsto r_{A_i}(S) \end{aligned} \tag{2}$$

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 of the ratings (3).

$$\begin{aligned} \alpha : \ & \{r_1, r_2, \ldots, r_p\}^n \longrightarrow \{r_1, r_2, \ldots, r_p\} \\ & (r_{A_1}(S), r_{A_2}(S), \ldots, r_{A_n}(S)) \longmapsto \alpha(S) \end{aligned} \tag{3}$$

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, 1996: 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 particularise 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.

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

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 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, 1999a: 12) and affects 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 on Evaluation*

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, 1999b, Annex B: 21–22). The ISO/IEC tables are only examples, 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, and 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 development process.

Closer to our focus on quality models, ISO/IEC 14598-4 also proposes a table with the prioritised 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 in this example, 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, 1998, Annex B: 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 levels range from A (most critical) to D (least critical) and concern four classes of risks: environment, safety (people), economy (companies) and security (data). The standard 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, one should carry out: execution time measurements; benchmark testing; an analysis of the design to determine the algorithmic complexity. It is obvious that only very specific factors are taken into account here (those associated with risks). But usability studies point out 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. This proposal for customisable quality models echoes work in the TEMAA project on a "Parametrisable Test Bed", in which user profiles determine the weighting of partial scores (TEMAA, 1996, chap. 4).

The Framework for MT Evaluation in the ISLE Project (FEMTI) contains the following elements:

i. A classification of the main features defining the context of use: the type of *user* of the MT system, the type of *task*, and the nature of the *input* to the system.
ii. A classification of the MT software quality characteristics, into hierarchies of sub-characteristics, with internal and/or external attributes (i.e., metrics) at the bottom level. The upper levels match the ISO/IEC 9126 characteristics.

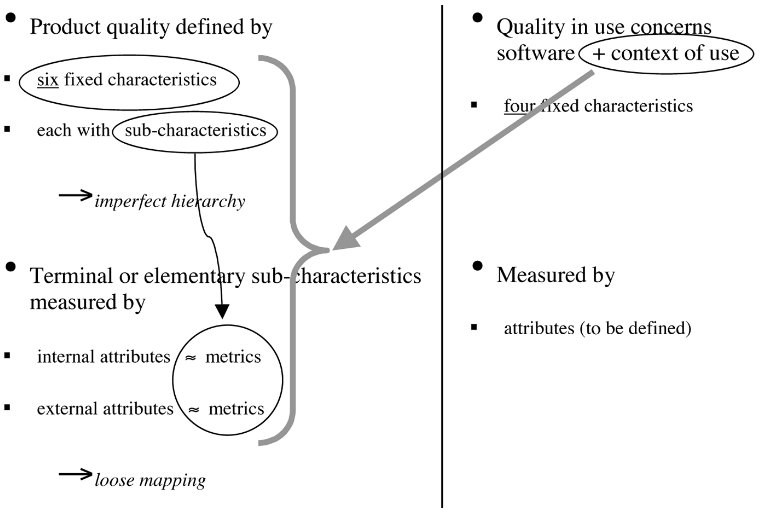
Product quality defined by
six fixed characteristics
each with sub-characteristics
imperfect hierarchy
Terminal or elementary sub-characteristics measured by
internal attributes ≈ metrics
external attributes ≈ metrics
loose mapping
Quality in use concerns software + context of use
four fixed characteristics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

Figure 1. Proposed systematic link between context of use (right) and quality model, i.e. quality characteristics plus metrics (left).

iii. A mapping from the first classification to the second, which defines (or at least suggests) the characteristics, sub-characteristics and attributes/metrics that are relevant to 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 theory related to the ISO/IEC quality in use, but 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 correspondence uses the definitions in Section 2.3.3. Remember that the set of all possible attributes of 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) (1), r_{A_i} (rating of measured value) (2), and α (assessment of ratings) (3). The mapping described at point (iii) of the previous section holds between a context of use and the assessment or averaging function α – the function that assigns a greater weight to the metrics/scores relevant to the respective context. Point (iii) is thus addressed simply by providing, for each context of use, the corresponding assessment function. We explain now how this function can be computed.

3.3.1. *Definitions*

The context of use modulates the assessment function that integrates the ratings of the measured attributes. Therefore, our goal is to define such a correspondence $\mathcal{M}$ between a context C and an assessment function α (determining a quality model) (4).

$$\begin{aligned} \mathcal{M} :\ & \mathfrak{C} \longrightarrow (\Re^n \longrightarrow \Re) \\ & C \longmapsto \alpha_{\mathcal{M}}(C) \end{aligned} \tag{4}$$

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 symmetric averaging function α_0 independent from the context (5), 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 (6).

$$\begin{aligned} \alpha_0 :\ & \Re^n \longrightarrow \Re \\ & (r_1, r_2, \dots, r_n) \longmapsto \alpha_0(r_1, r_2, \dots, r_n) \end{aligned} \tag{5}$$

$$\begin{aligned} \mathcal{S}_C :\ & \Re^n \longrightarrow \Re^n \\ & (r_1, \dots, r_n) \longmapsto (w_C(A_1) \cdot r_1, \dots, w_C(A_n) \cdot r_n) \\ & \text{where } w_C(A_i) \in [0, 1],\ \forall\, 1 \le i \le n \end{aligned} \tag{6}$$

The assessment function for a chosen context of use C is then $\alpha_0 \circ \mathcal{S}_C$, that is, the selection function followed by the averaging one. Coefficients $w_C(A_i)$ represent the importance of each quality attribute in the context of use C; for instance, when set at zero value, they can even rule out irrelevant attributes.

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 emphasised 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, numeric weights must be provided for relevant quality attributes (as a "weighting tuple"), but not for the total set of quality attributes.[6]

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

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

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,

$$w'_C(A_i) = \sum\nolimits_k v_k(A_i) \tag{7}$$

4. Normalise the final weight list for quality attributes by the highest weight in the list. Therefore, for each A_{i_0}, the associated weight is

$$w_C(A_{i_0}) = \frac{w'_C(A_{i_0})}{max_i(w'_C(A_i))} \tag{8}$$

The weights equal 1 for the essential attribute(s) and have smaller values for less important ones. Null weights appear for irrelevant attributes.

The weight list, combined with the assessment function α_O, 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; their scales must also be normalised. 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 summarise it here, since the full version covers more than 50 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.[7] The second subsection exemplifies in detail a fragment of our framework.

4.1. GENERAL OVERVIEW

Following the recommendations of ISO/IEC 14598-1 (ISO/IEC, 1999a), the FEMTI framework begins by asking the evaluation designer to make explicit why the evaluation is to be done, and what exactly must be evaluated. Whilst these two elements are in a way independent of the rest of the taxonomy, they obviously play a critical role in determining what elements of the taxonomy will be of interest in the case of a specific evaluation. The purpose of the evaluation may range from trying to assess whether a particular algorithm really does what it is meant to do, at one extreme, to providing the basis for a decision on whether to make a purchase implying a considerable investment at the other, with many and varied possibilities between the two. The object to be evaluated might be, for example, a component of a system, some other form of intermediate software product, an entire system considered in isolation from the contexts in which the system will be used, a system which is itself embedded in a larger system or a system considered from the particular perspective created by a specific context in which it will be used.

As suggested by this variety of possibilities, these preliminary considerations would deserve a great deal more development. But here we are mainly concerned with the relation between a generic quality model which tries to take into account all the quality characteristics which might be of interest – the superset, as it were, of the characteristics relevant in any particular evaluation – and a general statement of possible user requirements which is again a superset from which the evaluation designers may choose those pertinent to their case. Let it suffice here, then, to say that determining the purpose of the evaluation and clarifying exactly what must be evaluated are logically prior to choosing pertinent quality characteristics and metrics by which to measure them.

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) following the purpose and the object of the evaluation. 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. Evaluation requirements
    - 1.1 Purpose of evaluation
    - 1.2 Object of evaluation
    - 1.3 Characteristics of the translation task
        - 1.3.1 Assimilation
        - 1.3.2 Dissemination
        - 1.3.3 Communication
    - 1.4 User characteristics
        - 1.4.1 Machine translation user
        - 1.4.2 Translation consumer
        - 1.4.3 Organisational user
    - 1.5 Input characteristics (author and text)
        - 1.5.1 Document type
        - 1.5.2 Author characteristics
        - 1.5.3 Characteristics related to sources of error
2. System characteristics to be evaluated
    - 2.1 MT system-specific characteristics

- 2.1.1 Translation process models (rule-based – example-based – statistical – translation memory)
- 2.1.2 Linguistic resources and utilities
- 2.1.3 Characteristics of process flow

2.2 System external characteristics

- 2.2.1 Functionality
  - 2.2.1.1 Suitability (readability – comprehensibility – coherence – cohesion – coverage – style)
  - 2.2.1.2 Accuracy (fidelity – consistency – terminology)
  - 2.2.1.3 Well-formedness (punctuation – lexical choice – grammar – morphology)
  - 2.2.1.4 Interoperability
  - 2.2.1.5 Compliance
  - 2.2.1.6 Security
- 2.2.2 Reliability
- 2.2.3 Usability
- 2.2.4 Efficiency
  - 2.2.4.1 Time behavior (pre-processing time – input-to-output translation speed – post-processing time)
  - 2.2.4.2 Resource utilisation
- 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, as well as a classification of the *purposes* and *objects* of evaluation.

4.2. A FOCUS ON FLUENCY OR "OUTPUT QUALITY"

In this section we focus on what is commonly called "output quality" in MT evaluation. Although in many evaluations it is one of the basic aspects to be measured, quality is really a composite concept, and has sometimes also been called "fluency", "correctness", "intelligibility", "readability", "style", and "clarity". In this section we illustrate the richness and complexity of this taxon, developing it somewhat deeper, and thereby highlighting some of the problems of taxonomisation in general, an issue we take up again in Section 7.

The subtree rooted at *functionality* (taxon 2.2.1) is decomposed into three major parts:

- 2.2.1.1 Suitability: the general understandability of the form of the translation, without regard to its meaning

- 2.2.1.2 Accuracy: the fidelity of the translation (semantic correspondence of the input and output), without regard to its form
- 2.2.1.3 Well-formedness: the linguistic correctness of the output words and sentences as individual entities

In addition, three other subsidiary parts *interoperability*, *compliance*, and *security* pertain to general software characteristics.

Since space is limited, we focus on taxon 2.2.1.1 *Suitability*. This we separate into aspects that can be measured by reference to the output alone (2.2.1.1.1) and those that require a comparison of source and target texts (2.2.1.1.2). The former include *readability* "the extent to which each sentence reads naturally", *comprehensibility* "the extent to which the text as a whole is easy to understand", *coherence* "the extent to which the reader can grasp and explain the structure (internal interrelatedness) of the text", and *cohesion* "the extent to which text-internal links such as lexical chains are maintained in the translation".

Readability, the aspect of translation people seem to come to first, has been the subject of numerous MT evaluations. This aspect has also been called fluency, clarity, intelligibility, quality, etc. Most commonly, readability measures require the assessor to rate each sentence more or less subjectively on a scale of between 3 and 9 points. The system's average over a set of sentences is then reported. As listed in the taxonomy (2.2.1.1.1.1), this method has been used by Crook and Bishop (1965), Pfafflin (1965), ALPAC (1966),[8] Leavitt et al. (1971), Sinaiko (1979), Vauquois (1979), Miller and Vanni (2001) and others.

A second method to measure readability is the Cloze test. First, every *n*th word of the translation is (automatically) blanked out. Then, the assessor's task is to replace it. According to this measure, the degree of correctness of replacement reflects the readability/interpretability of the translation. Examples of this method appear in Crook and Bishop (1965), Halliday and Briss (1977), and Sinaiko (1979).

Other methods to measure readability are reading time – the longer it takes an assessor to read the translation, the worse it is (ALPAC, 1966; Dostert, 1973; van Slype, 1979) – and edit distance – the more insert/delete/swap operations it takes to correct the translation, the worse it is (Niessen et al., 2000).

Comprehensibility, *coherence*, and *cohesion* all measure how well the translation can be understood as a whole text. *Comprehensiblity* focuses on the extent to which valid inferences can be drawn by combining information from different parts of the document. For example, if one source sentence describes city A as the largest city, and another implies that city B is larger than city C, can the reader of the translation infer which of the three is the smallest city? Focusing on information that is not stated explicitly on the surface, this aspect is typically measured using a multiple-choice questionnaire (Orr and Small, 1967; Leavitt et al., 1971; White and O'Connell, 1994b). Following Rhetorical Structure Theory (RST) (Mann and Thompson, 1988) and similar theories of discourse structure, *coherence* measures how well the assessor can identify what role(s) a clause plays with respect to its surrounding clauses and ultimately in the text as a whole. For example, a system that translates each sentence perfectly but scrambles their order will score high on readability, possibly high on comprehensibility, but low on coherence. *Cohesion* pertains to the correct maintenance in the translation of lexical chains, that is, the identity or relatedness of concepts that re-appear throughout the text (Halliday and Hasan, 1976; Morris and Hirst, 1991). This aspect plays a large role in MT because it includes coreference/anaphora; relevant measures focus on whether pronouns in particular are properly handled by the system.

When the taxonomy bottoms out into specific measures, the essential subjectivity of MT evaluation measures stands out most clearly.[9] However, by specifying as precisely as possible the assessor's task and scoring options, one can help ensure a reasonable degree of inter-assessor consistency. Indeed, despite individual variations of measures of readability, it is fairly obvious that relative rankings of translation will be maintained across them – a very bad translation will score worse on all the measures than a partially bad one, etc.

5. Dissemination, Use and Update of the Taxonomy

5.1. A SERIES OF HANDS-ON EXERCISES

The EWG of the ISLE project concentrated 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 organised: in October 2000 (at AMTA 2000), April 2001 (stand-alone hands-on workshop at ISSCO, University of Geneva), June 2001 (at NAACL 2001), September 2001 (at MT Summit VIII), May 2002 (at LREC 2002). Another workshop is in preparation for September 2003 (at MT Summit IX).

All of the workshops included hands-on exercises in MT supported 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 by 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 are 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.

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. With this in mind, they 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) 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 involves finding out which FEMTI metrics best correlate with the "suitability of output for information processing" metric (Vanni and Miller, 2002). 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 produces feedback enabling future work.

5.3. MANAGEMENT OF THE FRAMEWORK

For the sake of simplicity, FEMTI can be accessed and browsed through a computer interface. The mechanism that supports this function also ensures that the various taxons (nodes and leaves) of the two classifications are stored in a coherent XML format, and simplifies considerably the periodic update of the classifications

(Popescu-Belis et al., 2001; Hovy et al., 2002). The current version of the framework is visible at `http://www.issco.unige.ch/projects/isle/taxonomy3/`.

5.3.1. *Data Structures*

One of the main advantages of using XML to store the contents of the two classifications is the separation between form and content. The conceptual format used for storage is quite independent of the displayed format. Each taxon receives a unique index number, and is stored in a separate computer file. The structure of the taxons varies between the first and the second classification. In both cases, the structure is defined using a DTD (document type definition); automatic *validation* ensures that each taxon conforms to the respective DTD. Figure 2 shows the essence of the DTD for part 1 (context/qualities):

```
<!ELEMENT taxon (index-number,
                 child-index-number*,
                 parent-index-number,
                 name,
                 definition,
                 relevant-qualities,
                 stakeholders?,
                 references?,
                 comments?)>
```

Figure 2. DTD for context and qualities.

The taxon structure for part 2 (qualities/metrics) follows a similar DTD, in which `relevant-qualities` are replaced with `metrics` and there are no stakeholders. Possible improvements of this mechanism include 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 encode formally in the part 1 taxons (context of use) the weighting tuples for the relevant quality attributes (these tuples are noted $v_k(A_i)$ in Section 3.3, equation (7)). 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 the 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

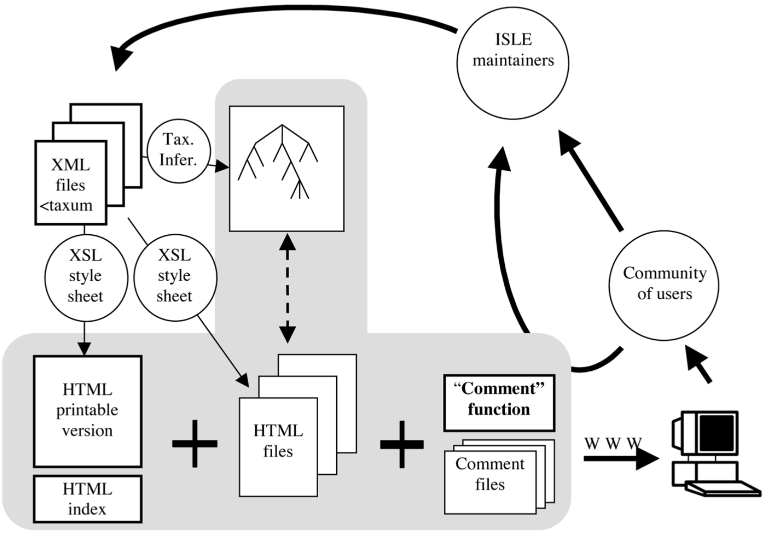

Figure 3. Life-cycle of the Framework for the Evaluation of MT in ISLE.

of a single HTML file for the whole framework (printer-friendly version), as well as a glossary. 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 need not be modified for each update of the framework). The mechanism that allows evaluators to parameterise the MT quality model depending on the context of use is currently under development.

The separation between the FEMTI taxons and the XML/XSL formatting mechanism facilitates the separate update of both form and content, and enables us to receive and log comments. FEMTI's life-cycle is summarised in Figure 3. Starting with the XML files for the individual taxons, the XSL stylesheets and other scripts generate the web site through which FEMTI is consulted, i.e. browsed or printed. A link for submitting comments is also automatically embedded in the HTML files. These comments and those received directly, e.g. during the workshops organised through the ISLE project, are gradually fed back to the taxon files. Once these suggestions have been validated, a new version of the web site is then generated.

6. Studies Towards the Refinement of the Taxonomies

FEMTI forms just 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 a set of coherence criteria for evaluation metrics, then lists the (unfortunately very few) examples of analyses 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, b). The main goal was to enable evaluators to choose the most suitable metric for a given attribute and to help them interpret the measures. One of the main hypotheses was that a given quality attribute could be measured by running the system with a relevant set of input data, then measuring the quality level for each of the system's responses. 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 for the quality of a given attribute without necessarily comparing the system's response with a correct one.

A metric for a given quality 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 (1) in Section 2.3.3, the metric (plus the rating) quantifies a system's position in the quality space for A_i. So, before analyzing metrics, evaluators must poll the experts to get an idea of what the best and the worst quality levels are for quality A_i.

It is often easy to define the best quality level, but there are at least two types 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 100% for perfect quality of the respective attribute A_i and (reciprocally) only reach 100% 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 always 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 and testing whether they really receive low scores.

LL–2′: A simpler necessary condition for LL–2 is that 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 criticise 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 generates 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 also be used in an absolute way, meaning “more severe (or lenient) than most of other metrics”.

6.2. ANALYZING THE BEHAVIOR OF METRICS

As exemplified 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. Recent promising work using the DARPA 1994 evaluation results (White and O'Connell, 1994a) has analysed 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 generalised 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 optimum 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 post-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 overall requirements of the evaluation. Armed with this knowledge, an evaluator would be able to make a much more informed selection of what to evaluate and how to interpret the obtained results.

7. Further Developments and Conclusion

A general theme running throughout this paper is that MT evaluation is only 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, dialog 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 type of 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 did 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 had 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 is 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 realise 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 of useful qualities. 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 organisation at the time of writing of the two FEMTI taxonomies, quoting here for each taxon only its title, but not its contents (see explanations in Section 4).

1. Evaluation requirements
   1.1 Purpose of evaluation
      1.1.1 Feasibility evaluation
      1.1.2 Requirements elicitation
      1.1.3 Internal evaluation
      1.1.4 Diagnostic evaluation
      1.1.5 Declarative evaluation
      1.1.6 Operational evaluation
      1.1.7 Usability evaluation
   1.2 Object of evaluation
      1.2.1 A component of an MT system
      1.2.2 An MT system considered as a whole
      1.2.3 An MT system considered as a component of a larger system
   1.3 Characteristics of the translation task
      1.3.1 Assimilation
         1.3.1.1 Document routing/sorting
         1.3.1.2 Information extraction/summarisation
         1.3.1.3 Search
      1.3.2 Dissemination
         1.3.2.1 Internal/in-house publication
            – Routine
            – Experimental/research
         1.3.2.2 External publication
            – Single-client
            – Multi-client
      1.3.3 Communication
         1.3.3.1 Synchronous
         1.3.3.2 Asynchronous
   1.4 User characteristics
      1.4.1 Machine translation user
         1.4.1.1 Education
         1.4.1.2 Proficiency in source language
         1.4.1.3 Proficiency in target language
         1.4.1.4 Computer literacy

1.4.2 Translation consumer
1.4.2.1 Proficiency in source language
1.4.2.2 Proficiency in target language
1.4.3 Organisational user
1.4.3.1 Quantity of translation
1.4.3.2 Number of personnel
1.4.3.3 Time allowed for translation
1.5 Input characteristics (author and text)
1.5.1 Document type
1.5.1.1 Genre
1.5.1.2 Domain/field of application
1.5.2 Author characteristics
1.5.2.1 Proficiency in source language
1.5.2.2 Professional training
1.5.3 Characteristics related to sources of error
1.5.3.1 Intentional error sources
1.5.3.2 Medium related error sources
1.5.3.3 Performance related errors

2. System characteristics to be evaluated
2.1 MT system-specific characteristics
2.1.1 Translation process models
2.1.1.1 Methodology

- Rule-based models
- Statistically-based models
- Example-based models
- Translation memory models

2.1.1.2 Models

- Direct
- Transfer
- Interlingua

2.1.2 Linguistic resources and utilities
2.1.2.1 Languages
2.1.2.2 Dictionaries
2.1.2.3 Word lists, glossaries
2.1.2.4 Comparable and parallel corpora
2.1.2.5 Grammars
2.1.3 Characteristics of process flow
2.1.3.1 Translation preparation activities
2.1.3.2 Post-translation activities
2.1.3.3 Interactive translation activities
2.1.3.4 Dictionary updating
2.1.3.5 Process management
2.2 System external characteristics
2.2.1 Functionality
2.2.1.1 Suitability

- Target-language only
  - Readability (or fluency, intelligibility, clarity)

        - Comprehensibility
        - Coherence
        - Cohesion
      - Cross-language/contrastive
        - Coverage of corpus-specific phenomena
        - Style

2.2.1.2 Accuracy
- Fidelity
- Consistency
- Terminology

2.2.1.3 Well-formedness
- Punctuation
- Lexis/lexical choice
- Grammar/syntax
- Morphology

2.2.1.4 Interoperability

2.2.1.5 Compliance

2.2.1.6 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

2.2.2.5 Reliability compliance

2.2.3 Usability

2.2.3.1 Understandability

2.2.3.2 Learnability

2.2.3.3 Operability

2.2.3.4 Documentation

2.2.3.5 Attractiveness

2.2.3.6 Usability compliance

2.2.4 Efficiency

2.2.4.1 Time behavior
- Pre-processing time
  - Pre-editing time
  - Code-set conversion
  - Preparation time
- Input-to-output translation speed
- Post-processing time
  - Post-editing time
  - Code-set conversion
  - Update time

2.2.4.2 Resource utilisation
- Memory
- Lexicon
- Clean-up
- Program size

2.2.5 Maintainability

2.2.5.1 Analyzability

2.2.5.2 Changeability

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

2.2.5.3 Stability

2.2.5.4 Testability

2.2.5.5 Maintainability compliance

2.2.6 Portability

2.2.6.1 Adaptability

2.2.6.2 Installability

2.2.6.3 Conformance

2.2.6.4 Replaceability

2.2.6.5 Co-existence

2.2.7 Cost

2.2.7.1 Introduction cost

2.2.7.2 Maintenance cost

2.2.7.3 Other costs

Notes

[1] 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.

[2] At `http://www.nist.gov/speech/tests/mt/`.

[3] 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 EWG, please visit our web site at: `http://www.issco.unige.ch/projects/isle/ewg.html`.

[4] 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.

[5] Following (ISO/IEC, 1991: 6) and (ISO/IEC, 1999a: 15–17).

[6] 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 plan to associate numeric weights between 0 and 1 to the attributes.

[7] The Appendix provides the full list of all the titles in 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/taxonomy3/`.

[8] See Appendix 10.

[9] Even so-called objective, fully automated evaluation measures are subject to this problem. All automated measures developed so far require one or more human translations to compare against. To

avoid individual bias and provide small enough margins of error, automated evaluation methods have to test a large enough set of candidate translations; in this they are similar to human scoring. In both automated and human evaluations, considering a large enough test set helps mitigate subjectivity through simple regression to the mean.

References

ALPAC: 1966, *Language and Machines: Computers in Translation and Linguistics*, A report by the Automatic Language Processing Advisory Committee, Division of Behavioral Sciences, National Research Council, Washington, DC: National Academy of Sciences. [Available at: `http://www.nap.edu/openbook/ARC000005/html/`]

Arnold, Doug, R. Lee Humphreys and Louisa Sadler (eds): 1993, Special Issue on Evaluation of MT Systems, *Machine Translation* **8**.1–2.

Blasband, Marc: 1999, 'Practice of Validation: The ARISE Application of the EAGLES Framework', in *Proceedings of the European Evaluation of Language Systems Conference (EELS)*, Hoevelaken, The Netherlands. [Available at: http://www.computeer.nl/eels.htm]

Canelli, Maria, Daniele Grasso and Margaret King: 2000, 'Methods and Metrics for the Evaluation of Dictation Systems: A Case Study', in *LREC 2000: Second International Conference on Language Resources and Evaluation*, Athens, pp. 1325–1331.

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

Crook, M. and H. Bishop: 1965, Evaluation of Machine Translation, Final report, Institute for Psychological Research, Tufts University, Medford, MA.

Daly-Jones, Owen, Nigel Bevan and Cathy Thomas: 1999, Handbook of User-Centred Design, Deliverable 6.2.1, INUSE European Project IE-2016. [Available at: `http://www.ejeisa.com/nectar/inuse/`]

Dostert, B. H.: 1973, User's Evaluation of Machine Translation: Georgetown MT System, 1963–1973, Report RADC-TR-73-239, Rome Air Development Center, Grifiss Air Force Base, NY, and Report AD-768-451, Texas A&M University, College Station, TX.

EAGLES-EWG: 1996, EAGLES Evaluation of Natural Language Processing Systems, Final Report EAG-EWG-PR.2, Project LRE-61-100, Center for Sprogteknologi, Copenhagen, Denmark. [Available at: `http://www.issco.unige.ch/projects/ewg96/`]

EAGLES-EWG: 1999, EAGLES Evaluation of Natural Language Processing Systems, Final Report EAG-II-EWG-PR.2, Project LRE-61-100, Center for Sprogteknologi, Copenhagen, Denmark. [Available at: `http://www.issco.unige.ch/projects/eagles/`]

Flanagan, Mary A.: 1994, 'Error Classification for MT Evaluation', in *Technology Partnerships for Crossing the Language Barrier: Proceedings of the First Conference of the Association for Machine Translation in the Americas*, Columbia, Maryland, pp. 65–72.

Forner, Monika and John S. White: 2001, 'Predicting MT Fidelity from Noun-Compound Handling', in *MT Summit VIII Workshop on MT Evaluation "Who did what to whom?"*, Santiago de Compostela, Spain, pp. 45–48.

Fuji, M., N. Hatanaka, E. Ito, S. Kamei, H. Kumai, T. Sukehiro and H. Isahara: 2001, 'Evaluation Method for Determining Groups of Users Who Find MT "Useful"', in *MT Summit VIII: Machine Translation in the Information Age*, Santiago de Compostela, Spain, pp. 103–108.

Halliday, M. A. K. and R. Hasan: 1976, *Cohesion in English*, London: Longman.

Halliday, T. and E. Briss: 1977, The Evaluation and Systems Analysis of the Systran Machine Translation System, Report RADC-TR-76-399, Rome Air Development Center, Griffiss Air Force Base, NY.

Hovy, Eduard H.: 1999, 'Toward Finely Differentiated Evaluation Metrics for Machine Translation', in *Proceedings of EAGLES Workshop on Standards and Evaluation*, Pisa, Italy.

Hovy, Eduard H., Margaret King and Andrei Popescu-Belis: 2002, 'Computer-Aided Specification of Quality Models for MT Evaluation', in *LREC 2002: Third International Conference on Language Resources and Evaluation*, Las Palmas de Gran Canaria, Spain, pp. 1239–1246.

Infoshop: 1999, *Language Translations: World Market Overview, Current Developments and Competitive Assessment*, Kawasaki, Japan: Infoshop Japan/Global Information Inc.

ISO/IEC: 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, December, 1991.

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

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

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

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

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

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

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

Kay, Martin: 1980, The Proper Place of Men and Machines in Language Translation, Research Report CSL-80-11, Xerox PARC, Palo Alto, CA; repr. in *Machine Translation* **12** (1997), 3–23.

King, Margaret and Kirsten Falkedal: 1990, 'Using Test Suites in Evaluation of Machine Translation Systems', in *COLING-90: Papers presented to the 13th International Conference on Computational Linguistics*, Helsinki, vol. 2, pp. 211–216.

Leavitt, A., J. Gates and S. Shannon: 1971, Machine Translation Quality and Production Process Evaluation, Report RADC-TR-71-206, Rome Air Development Center, Griffiss Air Force Base, NY.

Lehrberger, John and Laurent Bourbeau: 1988, *Machine Translation: Linguistic Characteristics of MT Systems and General Methodology of Evaluation*, Amsterdam: John Benjamins Press.

Mann, William C. and Sandra A. Thompson: 1988, 'Rhetorical Structure Theory: A Theory of Text Organization', *Text* **8**, 243–281.

Mason, Jane and Adriane Rinsche: 1995, *Translation Technology Products*, London: OVUM Ltd.

Miller, Keith J. and Michelle Vanni: 2001, 'Scaling the ISLE Taxonomy: Development of Metrics for the Multi-Dimensional Characterisation of MT Quality', in *MT Summit VIII: Machine Translation in the Information Age*, Santiago de Compostela, Spain, pp. 229–234.

Morris, J. and G. Hirst: 1991, 'Lexical Cohesion, the Thesaurus, and the Structure of Text', *Computational Linguistics* **17**, 21–48.

Nagao, Makoto: 1980, *A Japanese View on Machine Translation in Light of the Considerations and Recommendations Reported by ALPAC, USA*, Tokyo: Japan Electronic Industry Development Association (JEIDA).

Niessen, Sonja, Franz Josef Och, Gregor Leusch and Hermann Ney: 2000, ‘An Evaluation Tool for Machine Translation: Fast Evaluation for MT Research’, in *LREC 2000: Second International Conference on Language Resources and Evaluation*, Athens, Greece, pp. 39–45.

Nomura, Hirosato: 1992, *JEIDA Methodology and Criteria on Machine Translation Evaluation*, Tokyo: Japan Electronic Industry Development Association (JEIDA).

Nomura, Hirosato and Hitoshi Isahara: 1992, ‘Evaluation Surveys: The JEIDA Methodology and Survey’, in *MT Evaluation: Basis for Future Directions, Proceedings of a workshop sponsored by the National Science Foundation*, San Diego, CA, pp. 11–12.

Orr, D. and V. Small: 1967, ‘Comprehensibility of Machine-Aided Translations of Russian Scientific Documents’, *Mechanical Translation and Computational Linguistics* **10**, 1–10.

Papineni, Kishore, Salim Roukos, Todd Ward and Wei-Jing Zhu: 2001, BLEU: a Method for Automatic Evaluation of Machine Translation, Computer Science Research Report RC22176 (W0109-022), IBM Research Division, T. J. Watson Research Center, Yorktown Heights, NY. [Available at: `http://domino.watson.ibm.com/library/Cyberdig.nsf/home`]

Pfafflin, S.: 1965, ‘Evaluation of Machine Translations by Reading Comprehension Tests and Subjective Judgments’, *Mechanical Translation* **8**, 2–8.

Popescu-Belis, Andrei: 1999a, ‘Evaluation of Natural Language Processing Systems: a Model for Coherence Verification of Quality Measures’, in Marc Blasband and Patrick 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*, Deliverable D1.1, Project LE-4-8340, LIMSI-CNRS, Orsay, France.

Popescu-Belis, Andrei: 1999b, ‘L’évaluation en Génie Linguistique: Un Modèle pour Vérifier la Cohérence des Mesures’ [Evaluation in Language Engineering: A Model for Coherence Verification of Measures], *Langues* **2**, 151–162.

Popescu-Belis, Andrei, Sandra Manzi and Margaret King: 2001, ‘Towards a Two-stage Taxonomy for Machine Translation Evaluation’, in *MT Summit VIII Workshop on MT Evaluation “Who did what to whom?”*, Santiago de Compostela, Spain, pp. 1–8.

Rajman, Martin and Anthony Hartley: 2002, ‘Automatic Ranking of MT Systems’, in *LREC 2002: Third International Conference on Language Resources and Evaluation*, Las Palmas de Gran Canaria, Spain, pp. 1247–1253.

Sinaiko, H. W.: 1979, ‘Measurement of Usefulness by Performance Test’, in van Slype (1979), pp. 91ff.

Sparck Jones, Karen and Julia Rose Galliers: 1993, Evaluating Natural Language Processing Systems, Technical Report 291, University of Cambridge Computer Laboratory.

Sparck Jones, Karen and Julia Rose Galliers: 1996, *Evaluating Natural Language Processing Systems: An Analysis and Review*, Berlin: Springer-Verlag.

Taylor, Kathryn B. and John S. White: 1998, ‘Predicting What MT is Good for: User Judgements and Task Performance’, in David Farwell, Laurie Gerber and Eduard H. Hovy (eds), *Machine Translation and the Information Soup*, Berlin: Springer-Verlag, pp. 364–373.

TEMAA: 1996, TEMAA Final Report, LRE-62-070, Center for Sprogteknologi, Copenhagen, Denmark. [Available at: `http://cst.dk/temaa/D16/d16exp.html`]

Thompson, Henry S. (ed.): 1992, *Proceedings of the Workshop on The Strategic Role of Evaluation in Natural Language Processing and Speech Technology*, HCRC, University of Edinburgh.

Tomita, Masaru: 1992, ‘Application of the TOEFL Test to the Evaluation of English-Japanese MT’, in *MT Evaluation: Basis for Future Directions, Proceedings of a workshop sponsored by the National Science Foundation*, San Diego, CA, p. 59.

Vanni, Michelle and Keith J. Miller: 2001, ‘Scoring Methods for Multi-Dimensional Measurement of Machine Translation Quality’, in *MT Summit VIII Workshop on MT Evaluation “Who did what to whom?”*, Santiago de Compostela, Spain, pp. 21–28.

Vanni, Michelle and Keith J. Miller: 2002, ‘Scaling the ISLE Framework: Use of Existing Corpus Resources for Validation of MT Metrics across Languages’, in *LREC 2002: Third International*

Conference on Language Resources and Evaluation, Las Palmas de Gran Canaria, Spain, pp. 1254–1262.

van Slype, Georges: 1979, Critical Study of Methods for Evaluating the Quality of Machine Translation, Final report BR 19142, Brussels: Bureau Marcel van Dijk. [Available at: http://issco-www.unige.ch/projects/isle/van-slype.pdf]

Vasconcellos, Muriel (ed.): 1992, *MT Evaluation: Basis for Future Directions, Proceedings of a workshop sponsored by the National Science Foundation*, San Diego, CA.

Vauquois, Bernard: 1979, 'Measurement of Intelligibility of Sentences on Two Scales', in van Slype (1979), pp.71ff.

White, John S.: 2001, 'Predicting Intelligibility from Fidelity in MT Evaluation', in *MT Summit VIII Workshop on MT Evaluation "Who did what to whom?"*, Santiago de Compostela, Spain, pp. 35–38.

White, John S. and Theresa A. O'Connell: 1994a, *ARPA Workshops on Machine Translation: a Series of Four Workshops on Comparative Evaluation, 1992–1994*, McLean, VA: Litton PRC Inc.

White, John S. and Theresa A. O'Connell: 1994b, 'The ARPA MT Evaluation Methodologies: Evolution, Lessons, and Future Approaches', in *Technology Partnerships for Crossing the Language Barrier: Proceedings of the First Conference of the Association for Machine Translation in the Americas*, Columbia, Maryland, pp. 193–205.

White, John S. and Kathryn B. Taylor: 1998, 'A Task-Oriented Evaluation Metric for Machine Translation', in *LREC 1998: First International Conference on Language Resources and Evaluation*, Granada, Spain, pp. 21–25.