Computer-Aided Specification of Quality Models for Machine Translation Evaluation

Eduard Hovy[*], **Margaret King**[**], **Andrei Popescu-Belis**[**]

[*]USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292-6695, USA
hovy@isi.edu

[**]ISSCO / TIM / ETI, University of Geneva
40 Bvd. du Pont d'Arve
CH–1211 Geneva 4, Switzerland
{margaret.king, andrei.popescu-belis}@issco.unige.ch

Abstract

This article describes the principles and mechanism of an integrative effort in machine translation (MT) evaluation. Building upon previous standardization initiatives, above all ISO/IEC 9126, 14598 and EAGLES, we attempt to classify into a coherent taxonomy most of the characteristics, attributes and metrics that have been proposed for MT evaluation. The main articulation of this flexible framework is the link between a taxonomy that helps evaluators define a context of use for the evaluated software, and a taxonomy of the quality characteristics and associated metrics. The article explains the theoretical grounds of this articulation, along with an overview of the taxonomies in their present state, and a perspective on ongoing work in MT evaluation standardization.

1. Introduction

Evaluating machine translation 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. 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 way, and that might help pave the way toward an eventual theory of MT evaluation.

Our main effort is to build a coherent overview of the various features and metrics used in the past, to offer a common descriptive framework and vocabulary, and to unify the process of evaluation design. Therefore, we present here a parameterizable taxonomy of the various attributes of an MT system that are relevant to its utility, as well as correspondences between the intended context of use and the desired system qualities, i.e., a quality model. Our initiative builds upon previous work in the standardization of evaluation, while applying to MT the ISO/IEC standards for software evaluation.

We first review (Section 2.) the main evaluation efforts in MT and in software engineering (ISO/IEC standards). Then we define our main theoretical stance, i.e., the need for two taxonomies, one relating the context of use (analyzed in Section 3.) to the quality characteristics, the other relating the quality characteristics to the metrics (Section 4.). In Section 5. we provide a brief overview of these taxonomies, together with a view on their dissemination and use. We finally outline (Section 6.) our perspectives on current and future developments.

2. Formalizing Evaluation: from MT to Software Engineering

2.1. Previous Approaches to MT Evaluation

The path to a systematic picture of MT evaluation is long and hard. 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 *quality* and *fidelity*—stand out. Particularly MT researchers often feel that if a system produces syntactically and lexically well-formed sentences (i.e., high quality output), and does not distort the meaning (semantics) of the input (i.e., high fidelity), then the evaluation is sufficient. System developers and real-world users often add evaluation measures, notably *system extensibility* (how easy it is for a user to add new words, grammar, and transfer rules), *coverage* (specialization of the system to the domains of interest), and *price*. 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 quality have been proposed, some focusing on specific syntactic constructions (relative clauses, number agreement, etc.) (Flanagan, 1994), others simply asking judges to rate each sentence as a whole on an *N*-point scale (White et al., 1992 1994; Doyon et al., 1998), and others automatically measuring the perplexity of a target text against a bigram or trigram language model of ideal translations (Papineni et al., 2001). The amount of agreement among such measures has never been studied. Fidelity requires bilingual judges, and is usually measured on an *N*-point scale by having judges rate how well each portion of the system's output expresses the content of an equivalent portion of one or more ideal (human) translations (White et al., 1992 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 above-mentioned perplexity measure also to evaluate fidelity (Papineni et al., 2001).

The Japanese JEIDA study of 1992 (Nomura, 1992; Nomura and Isahara, 1992), paralleling EAGLES, 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 to determine the degree of match, and hence to predict which system would be appropriate for which user. In 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 includes usability, customizability, application to total translation process, language coverage, terminology building, documentation, and others.

The variety of MT evaluations is enormous, from the influential ALPAC Report (Pierce et al., 1966) to the largest ever competitive MT evaluations, funded by the US Defense Advanced Research Projects Agency (DARPA) (White et al., 1992 1994) and beyond. Some influential contributions are (Kay, 1980; Nagao, 1989). Van Slype (1979) produced a thorough study reviewing MT evaluation at the end of the 1970s, and reviews for the 1980s can be found in (Lehrberger and Bourbeau, 1988; King and Falkedal, 1990). The pre-AMTA workshop on evaluation contains a useful set of papers (AMTA, 1992).

2.2. The EAGLES Guidelines for NLP Evaluation

The European EAGLES initiatives (1993-1996) came into being as an attempt to create standards for language engineering. 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 of the system. However, it did seem possible to create 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 the 1993 report by Sparck-Jones and Galliers, later published in book form (1996), and the ISO/IEC 9126 (cf. next section).

These first attempts proposed the definition of a general quality model for NLP systems in terms of a hierarchically structured set of features and attributes, where the leaves of the structure were measurable attributes, with which specific metrics were associated. 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 metrics to be combined in different ways in order to reflect differing needs. These attempts were validated by application to quite simple examples of language technology: spelling checkers, then grammar checkers (TEMAA, 1996) and translation memory systems (preliminary work), but the EAGLES methodology was also used outside the project for dialogue, speech recognition and dictation systems.

When the ISLE project (International Standards for Language Engineering) was proposed in 1999, 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 (one of the three ISLE working groups) therefore decided to concentrate on MT systems.

2.3. The ISO/IEC Standards for Software Evaluation

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. In 1991 appeared the ISO/IEC 9126 standard (ISO/IEC-9126, 1991), a milestone that proposed a definition of the concept of *quality*, and decomposed 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).

Subsequent efforts led to a set of standards, some still in draft versions today. It appeared that a new series was necessary for the evaluation process, of which the first in the series (ISO/IEC-14598, 1998 2001, Part 1) provides an overview. The new version of the ISO/IEC 9126 standard 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 (ISO/IEC-14598, 1998 2001), now completely published, volumes subsequent to ISO/IEC 14598-1 focus on the planning and management (14598-2) and documentation (14598-6) of the evaluation process, and apply the generic organization framework to developers (14598-3), acquirers (14598-4) and evaluators (14598-5).

2.3.2. The Definition of a Quality Model

This subsection situates our proposal for MT evaluation within the ISO/IEC framework. According to ISO/IEC 14598-1 (1998 2001, Part 1, p. 12, fig. 4), the software life-cycle starts with an analysis of user needs that will be answered by the software, which determine in their turn a set of 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 is 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).

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 addition, for MT software, there seems to be no straightforward link, in the conception phase, from the external quality requirements to the internal structure of a system. Therefore, the relation between external and internal qualities is quite loose.

Following mainly (ISO/IEC-9126-1, 2001), software quality results from six *quality characteristics*:

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

These characteristics have been refined into sub-characteristics that are still domain-independent (ISO/IEC 9126-1). These form a loose hierarchy (some overlappings are possible), but the terminal entries are always measurable features of the software, that is, *attributes*. Following (ISO/IEC-14598, 1998 2001, Part 1), "*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*".

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 are certainly different, since external quality is measured through external attributes (related to the behavior of a system) while internal quality is measured through internal attributes (related to intrinsic features of the system).

Finally, quality in use results from four characteristics: effectiveness, productivity, safety, and satisfaction. These can only be measured in the operating environment of the software, thus seeming less prone to standardization (see however (Daly-Jones et al., 1999) and ISO/IEC 9126-4).

2.3.3. Stages in the Evaluation Process

The five consecutive phases of the evaluation process according to (ISO/IEC-9126, 1991, p. 6) and (ISO/IEC-14598, 1998 2001, Part 5, p. 7) are:

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

During specification of the measurements, each required quality characteristic must be decomposed into the relevant sub-characteristics, and metrics must be specified for each of the attributes arrived at in this process. More precisely, three elements must be distinguished in the specification and design processes; these correspond to the following stages in execution:

- application of a metric (*a.*)
- rating of the measured value (*b.*)
- integration or assessment of the various ratings (*c.*)

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. Still, at the level of concrete evaluations of systems, the above distinction, advocated also by EAGLES (1996), seems particularly useful: to evaluate a system, a metric is applied for each of the selected attributes, yielding as a score a raw or intrinsic score; these scores are then transformed into marks or rating levels on a given scale; finally, during assessment, rating levels are combined if a single result must be provided for a system.

2.3.4. Formal Definition of the Stages

More formally, following previous work (Popescu-Belis, 1999), let S be a system for which several attributes must be evaluated, say $A_1, A_2, \ldots, A_n$. First, the system is subjected to a metric m_{A_i} for each attribute, producing a value on a scale that is intrinsic to the metric m_{A_i}, and is in general not tailored to reflect whether the result will be considered satisfactory. More formally, if the set of all systems is Σ and the scale associated to the metric m_{A_i} is the interval $[\inf(m_{A_i}), \sup(m_{A_i})]$, the m_{A_i} function has the following type:

a. application of a metric:

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

Each measured value is then rated with respect to the desired values, giving a set of satisfaction 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 a value set (for example, a numeric scale). The mapping between the measured values and the ratings reflects the human judgment of the attribute's quality. The rating function has the following type:

b. rating of the measured value:

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

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:

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

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

3. Relation between the Context of Use, Quality Characteristics, and 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 user/evaluator. The importance of the context for effective system deployment and use has been long understood, 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

While a good definition of the context of use is essential for accurate evaluation, in ISO/IEC the context of use plays a somewhat lesser role. The context of use is considered at the beginning of the software’s life-cycle (ISO/IEC-14598, 1998 2001, Part 1), and appears in the definition of quality in use. No obvious connection between quality in use metrics and internal or external ones is provided. There is thus no overall indication how to take into account the context of use in evaluating a product.

There are however two interesting mentions of the context of use in ISO/IEC. First, the ISO/IEC standard for acquirers (ISO/IEC-14598, 1998 2001, Part 4, Annex B, pp. 21-22) exemplifies the link between the desired *integrity* of the evaluated software (integrity pertains to the risk of using the software) and the evaluation activities, in particular the choice of a quality model: for higher integrity, more evaluation procedures have to be fulfilled. The six ISO/IEC 9126 characteristics are also ordered differently according to the required integrity. Second, (ISO/IEC-14598, 1998 2001, Part 5, Annex B, pp. 22-25) gives another relation between “evaluation techniques” and the acceptable risk level. These proposals attempt thus to fill the gap between concrete contexts of use and generic quality models.

3.2. Relating the Context of Use to the Quality Model

When specifying an evaluation, the external evaluator—a person or group in charge of estimating the quality of MT software—must mainly provide a quality model based on the expected context of use of the software. Guidelines for MT evaluation must therefore contain 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 and attributes, with internal and/or external attributes (i.e., metrics) at the bottom level. The upper levels coincide with 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 or metrics that are the most relevant for each context of use.

This broad view of evaluation is still, by comparison to ISO/IEC, focused on the technical aspect of evaluation. Despite the proximity between the taxonomy of contexts of use and quality in use, 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 measures. Therefore, in what follows, we will first propose a formal model of the mapping at point (3) above (next section), then outline the contents of points (1) and (2) above (Section 5.).

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

Building upon the definitions in Section 2.3.3., the set of all possible attributes for MT software is noted $\{A_1, A_2, \ldots, A_n\}$, and the process of evaluation is defined using three stages and the corresponding mappings: m_{A_i} (application of metrics), r_{A_i} (rating of measured value), and α (assessment of ratings).

From this point of view, the correspondence described at point (3) above is between a context of use and the assessment or averaging function α. Point (3) is thus addressed by providing, for each context of use, the corresponding assessment function, i.e. the function that assigns a greater weight to the attributes relevant to that particular context.

4.1. Definitions

If the role of the context of use is to modulate the assessment function that integrates the ratings of the measured values of attributes, our long term goal is to define such a correspondence $\mathcal{M}$:

- context / quality model correspondence $\mathcal{M}$:

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

One can imagine, of course, an endless variety of assessment functions α and therefore of mappings $\mathcal{M}$. Hence, we will 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 $\mathcal{S}_C$ providing a multiplicative coefficient $\sigma_C(A_i)$ for each attribute A_i, which depends on the desired context of use C. In other words, point (3) above amounts to defining two functions:

a. fixed averaging function α_0:

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

b. linear selection function for each context C:

$$\begin{array}{l|l} & \mathcal{S}_C : \Re^n \longrightarrow \Re^n \\ & \qquad (r_1, \ldots, r_n) \longmapsto (\sigma_C(A_1) \cdot r_1, \ldots, \sigma_C(A_n) \cdot r_n) \\ & \text{where } \sigma_C(A_i) \in [0,1],\ \forall\, 1 \leq i \leq n. \end{array}$$

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 $\sigma_C(A_i)$ represent the importance of each quality attribute in context of use C, and may rule out irrelevant attributes.

4.2. An Algorithm for Specifying Evaluation

An attempt to put theory into practice shows quickly that defining each context of use and its selection function is a burdensome task. Many contexts of use share a significant number of requirements, and only a few attributes are emphasized differently in each of them. Therefore, the previous model must be adapted twice:

1. The taxonomy of contexts of use must allow *non exclusive* characteristics. At the bottom level, weights are provided for quality attributes pertaining to the relevant (sub-)characteristics of the context, but not for the whole range of possible attributes.

2. The assessment function for a given context must integrate weights from various non exclusive (sub-)characteristics that hold for that context.

For the time being, the taxonomy of contexts of use provides for each (sub-)characteristic a set of attributes by order of relevance. In the near future, we will associate numeric weights between 0 and 1 to the attributes. Evaluation specification will then obey the following algorithm:

1. Start with null weights for all quality attributes.

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 applies, then add the weights $s_k(A_i)$ it provides for its relevant argument(s). 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, ${\sigma'}_C(A_i) = \sum_k s_k(A_i)$.

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 $\sigma_C(A_{i_0}) = {\sigma'}_C(A_{i_0})/max_i({\sigma'}_C(A_i))$. The weights will 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, obtained at this point, constitute the final step of the measurement–rating–assessment process. The second taxonomy that is part of these guidelines must be consulted at this point to find metrics for the attributes that have non-null weights. The evaluation can then be executed.

5. The Contents of the Two Taxonomies

The schema below gives a general view of the contents of the two taxonomies. The first one enumerates non exclusive characteristics of the context of use grouped in three complementary parts (task, user, input). The second one develops the quality model, and its starting point is 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 mentioned in Section 2.3.2., the link between internal features and external performance is not yet completely clear for MT systems. So, the internal attributes are structured here in a branch separate from the six ISO/IEC characteristics, which are measured by external metrics.

For lack of space, the hierarchies below represent a brief snapshot of the actual state of our proposal, which may be revised under feedback from the community. The full version available over the Internet (see below), has about 30 pages, and expands each 'taxum' with the corresponding metrics extracted from the literature.

- Specifying the context of use
  - Characteristics of the translation task
    - Assimilation
    - Dissemination
    - Communication
  - Characteristics of the user of the MT system
    - Linguistic education
    - Language proficiency in source language
    - Language proficiency in target language
    - Present translation needs
  - Input characteristics (author and text)
    - Document / text type
    - Author characteristics
    - Sources of error in the input
      - Intentional error sources
      - Medium-related error sources
      - Performance-related errors
- Quality characteristics, sub-characteristics and attributes
  - System internal characteristics
    - MT system-specific characteristics (translation process)
    - Model of translation process (rule-based / example-based / statistical / translation memory)
    - Linguistic resources and utilities
    - Characteristics related to the intended mode of use
      - Post-editing or post-translation capacities
      - Pre-editing or pre-translation capacities
      - Vocabulary search
      - User performed dictionary updating
      - Automatic dictionary updating
  - System external characteristics
    - Functionality
      - Suitability (coverage — readability — fluency / style — clarity — terminology)
      - Accuracy (text as a whole — individual sentence level — types of errors)
      - Interoperability
      - Compliance
      - Security
    - Reliability
    - Usability
    - Efficiency
      - Time behavior (production time / speed of translation — reading time — revision and post-editing / correction time)
      - Resource behavior
    - Maintainability
    - Portability
    - Cost

Practical work using the present taxonomy was the object of a series of workshops organized by the Evaluation Work Group of the ISLE Project. There has been considerable continuity between workshops, with the result that the most recent in the series offered 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, ways to take into account different user needs, novel scenarios both for the evaluation and for the ultimate use of an MT system and ways to automate MT evaluation. The four workshops took place in October 2000 (at AMTA 2000), April 2001 (stand-alone hands-on workshop at ISSCO, Geneva), June 2001 (at NAACL 2001) and September 2001 (at MT Summit VIII).

Among the first conclusions drawn from the workshops is the fact that evaluators tend to favor some parts of the second taxonomy—especially attributes related to the quality of the output text—and to neglect some others—for instance the definition of a user profile. It appears that the sub-hierarchy related to the "hard problem", i.e. the quality of output text, should be better developed. Subcharacteristics such as the translation quality for noun phrases (which is further on split into several attributes) attracted steady interest.

The proposed taxonomies can be accessed and browsed through a computer interface. The mechanism that supports this function also ensures that the various nodes and leaves of the categories are stored in a common format (based on XML), and simplifies considerably the periodic update of the classifications (Popescu-Belis et al., 2001). A first version of our taxonomies is visible at `http://www.isi.edu/natural-language/mteval` and a second one at `http://www.issco.unige.ch/projects/isle/taxonomy2`—the two sites will soon mirror a third, updated version.

6. Towards the Refinement of the Taxonomies

The taxonomies form but the first step in a larger programme— listing the essential parameters of importance to MT evaluation. But for a comprehensive and systematic understanding of the problem, one also has to analyze 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 defined coherence criteria for NLP evaluation metrics in an EAGLES-based framework (Popescu-Belis, 1999). The following criteria, applied to a case where there is no golden standard to compare a system's response to, enable evaluators to choose the most suitable metric for a given attribute and help them interpret the measures.

A metric m_{A_i} for a given attribute A_i is a function from an abstract 'quality space' onto a numeric interval, say [0,1] or [0%, 100%]. With respect to definition (*a.*) in Section 2.3.3., each system occupies a place in the quality space of A_i, quantified by that metric. Since the goal of evaluators is to quantify the quality level using a metric, they 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 find the best quality of a response, 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, for the capacity to translate polysemous words, 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 limits are identified, the following coherence criteria should be tested for:

- **UL — upper limit**: A metric for attribute A_i must reach 1 for best quality of a system, and (reciprocally) only reach 1 when the quality is perfect.
- **LL — lower limit**: A metric for attribute A_i must reach 0 for the worst possible quality of a system, and only reach 0 when the quality is extremely low. Since it is not easy to identify the set of lowest quality cases, one can alternatively check that:
  - receiving a 0 score corresponds to low quality;
  - all the worst quality responses receive a 0 score;
  - the lowest theoretical scores are close or equal to 0 (a necessary condition for the previous requirement).
- **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, one can also compare two metrics, stating that m_1 is more severe than m_2 if it yields lower scores for each possible quality level.

6.2. Analyzing the Behavior of Measures

Since our taxonomy gathers numerous quality attributes and metrics, there are basic aspects of MT that may be rated through several attributes, and each attribute may be scored using several metrics. This uncomfortable state of affairs calls for investigation. If it should turn out, for a given characteristic, that one specific attribute correlates perfectly with human judgments, subsumes most or all of the other proposed measures, can be expressed easily through one or more metrics, and is cheap to apply, we should have no reason to look further: that aspect of the taxonomy would be settled.

The full list of desiderata for a measure is not immediately clear, but there are some obvious ones. The measure:

- must be easy to define, clear and intuitive;

- must correlate well with human judgments under all conditions, genres, domains, etc.;
- must be 'tight', 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 for training data or ideal examples);
- must be cheap to apply;
- should be automated if possible.

Unexpectedly, the literature contains rather few methodological studies of this kind. Few evaluators have bothered to try someone else's measures too, and correlate the results. However, there are some advances. In recent promising work using the DARPA 1994 evaluation results (White et al., 1992 1994), White and Forner have studied the correlation between intelligibility (syntactic fluency) and fidelity (White, 2001) and between fidelity and noun compound translation (Forner and White, 2001). As one would expect with measures focusing on aspects as different as syntax and semantics, some correlation was found, but not a clear one. Papineni et al. (2001) compared the scores given by BLEU, an algorithm mentioned above, with human judgments of the fluency and fidelity of translations. They found a very high level of agreement, with correlation coefficients of 0.99 (with monolingual judges) and 0.96 (bilingual ones).

Another important matter is inter-evaluator agreement, reported on by most careful evaluations. Although the way one formulates instructions has a major effect on subjects' behavior, we still lack 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- or 7-point. 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). Still another important issue is the number of measure points required by each metric before the evaluation can be trusted, a figure that can be inferred from the confidence levels of past evaluation studies.

In the ISLE research we are now embarking on the design of a programme that will help address these questions. Our very ambitious goal is to know, for each taxon in the taxonomy, which measure(s) are most appropriate, which metric(s) to use for them, how much work and cost is involved in applying each measure, and what final level of score should be considered acceptable (or not). 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.

6.3. A View to the Future

It can be appreciated that building a taxonomy 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 taxonomy structure, and it is unfortunately very difficult to argue for the correctness of one against another. We therefore explicitly do not claim in this work that the present taxonomy 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 taxonomy, collecting additional references, and classifying additional measures, we feel that the most pressing work is only now being started. The taxonomy is but the first step toward a more comprehensive and systematic understanding of MT evaluation in all its complexity, including a dedicated programme of systematic comparison between metrics.

The dream of a magic test that makes everything easy—preferably an automated process—always remains. A recent candidate, proposed by (Papineni et al., 2001), has these desirable characteristics. Should it be true that the method correlates very highly with human judgments, and that it really requires only a handful of 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, it still does not cover more than a small portion of the taxonomy; all the other aspects of MT that people have wished to measure in the past remain to be measured.

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. However, as the systems to be evaluated grow more complex, the contexts of use become potentially almost infinite. Trying to imagine them all and to draw up a descriptive scheme as we are doing for MT systems becomes a challenging problem, that must be addressed in the future. It is nevertheless our belief that the basic ISO notion of building a quality model and associating appropriate metrics to it should carry over to almost any application.

7. References

AMTA. 1992. MT evaluation: Basis for future directions (Proceedings of a workshop held in San Diego, CA). Technical report, Association for Machine Translation in the Americas (AMTA).

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

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

J. Doyon, K. Taylor, and J.S. White. 1998. The DARPA MT 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, October 1996.

M. Flanagan. 1994. Error classification for MT evaluation. In *Proceedings of the AMTA Conference*, Columbia, Maryland.

M. Forner 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.

E.H. Hovy. 1999. Toward finely differentiated evaluation metrics for MT. 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. 1998-2001. *ISO/IEC 14598 — Information technology — Software product evaluation — Part 1: General overview (1999), Part 2: Planning and management (2000), Part 3: Process for developers (2000), Part 4: Process for acquirers (1999), Part 5: Process for evaluators (1998), Part 6: Documentation of evaluation modules (2001)*. ISO/IEC, Geneva.

ISO/IEC-9126-1. 2001. *ISO/IEC 9126-1:2001 (E) — Software engineering — Product quality — Part 1: Quality model*. ISO/IEC, Geneva, June.

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

M. Kay. 1980. The proper place of men and machines in language translation. Research Report CSL-80-11, XEROX PARC.

M. King and K. Falkedal. 1990. Using test suites in evaluation of MT systems. In *18th Coling Conference*, volume 2, Helsinki, Finland.

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

J. Mason and A. Rinsche. 1995. Translation technology products. Report, OVUM Ltd.

M. Nagao. 1989. A Japanese view on MT in light of the considerations and recommendations reported by ALPAC, USA. Technical report, Japan Electronic Industry Development Association (JEIDA).

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

H. Nomura. 1992. JEIDA methodology and criteria on MT evaluation. Technical report, Japan Electronic Industry Development Association (JEIDA).

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. 2001. BLEU: a method for automatic evaluation of MT. Research Report, Computer Science RC22176 (W0109-022), IBM Research Division, T.J.Watson Research Center, 17 September 2001.

J.R. Pierce, J.B. Carroll, E.P. Hamp, D.G. Hays, C.F. Hockett, A.G. Oettinger, and A. Perlis. 1966. Computers in translation and linguistics (ALPAC report). report 1416, National Academy of Sciences / National Research Council, 1966.

A. Popescu-Belis, S. Manzi, and M. King. 2001. Towards a two-stage taxonomy for MT evaluation. In *Workshop on MT Evaluation ”Who did what to whom?” at Mt Summit VIII*, pages 1–8, Santiago de Compostela, Spain.

A. Popescu-Belis. 1999. Evaluation of natural language processing systems: a model for coherence verification of quality measures. In Marc Blasband and Patrick Paroubek, editors, *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).

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

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.

H. S. Thompson, editor. 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).

G. VanSlype. 1979. Critical study of methods for evaluating the quality of MT. Technical Report BR 19142, European Commission / Directorate for General Scientific and Technical Information Management (DG XIII).

J.S. White et al. 1992-1994. ARPA workshops on MT (series of four workshops on comparative evaluation). Technical report, PRC Inc., McLean, Virginia.

J.S. White. 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.