A Model for Context-Based Evaluation of Language Processing Systems and its Application to Machine Translation Evaluation

Andrei Popescu-Belis, Paula Estrella, Margaret King, Nancy Underwood

ISSCO / TIM / ETI
University of Geneva
40, bd. du Pont-d'Arve
1211 Geneva 4, Switzerland
{andrei.popescu-belis, paula.estrella, margaret.king, nancy.underwood}@issco.unige.ch

Abstract

In this paper, we propose a formal framework that takes into account the influence of the intended context of use of an NLP system on the procedure and the metrics used to evaluate the system. We introduce in particular the notion of a context-dependent quality model and explain how it can be adapted to a given context of use. More specifically, we define vector-space representations of contexts of use and of quality models, which are connected by a *generic contextual quality model (GCQM)*. For each domain, experts in evaluation are needed to build a GCQM based on analytic knowledge and on previous evaluations, using the mechanism proposed here. The main inspiration source for this work is the FEMTI framework for the evaluation of machine translation, which implements partly the present model, and which is described briefly along with insights from other domains.

1. Introduction

The ISO/IEC standards for software evaluation, together with the EAGLES guidelines for human language technology (HLT) evaluation, propose a formal framework that standardizes evaluation procedures up to a certain point. The framework concerns the evaluation process and the top level types of quality characteristics, but specific quality attributes and evaluation metrics must be defined for each particular domain. Although the importance of user needs in establishing the evaluation requirements and the quality model has long been recognized, a generic account of this influence is not yet available. In this paper, we propose a framework that follows the ISO/IEC and EAGLES guidelines for HLT evaluation, in order to model the role of the intended context of use of an HLT system on the required quality characteristics, and on the attributes that will be evaluated through specific metrics.

The paper is organized as follows. Section 2 overviews the inspiration from ISO/IEC standards related to software quality, then Section 3 introduces the notion of a generic contextual quality model (GCQM) that relates the vector-space representation of the context of use and that of the quality model. Section 4 shows how the model could become operational, for evaluators and for experts in evaluation, through the definition of specific interfaces and workflows. The FEMTI framework for the evaluation of machine translation, which is the main inspiration source for the present proposal and its main practical test bed, is described briefly in Section 5, along with the implemented consultation tool. Finally, the application of the proposed model to other domains is discussed in Section 6.

2. Evaluation Requirements in the ISO/IEC Standards for Software Evaluation

The ISO/IEC 9126 and 14598 series of standards in the field of software evaluation provide a general definition of quality, and situate the process of evaluation within the software lifecycle (ISO/IEC, 1999a; ISO/IEC, 2001). The internal and external qualities of a software system can be decomposed into six broad classes of quality characteristics, related to functionality, reliability, usability, efficiency, maintainability, and portability; These must be refined into a hierarchy of sub-characteristics depending on each particular domain. Such a fully-fledged hierarchy, bottoming out in terminal nodes that represent quality attributes measurable by metrics, is called a *quality model*.

According to ISO/IEC 14598-1 (ISO/IEC 2000: p.12, fig.4), the software life-cycle starts with the analysis of the user requirements or user needs that will be answered by the software, which determine a set of software specifications, or external quality requirements. During initial development, software quality is internal, i.e. related to the implementation characteristics of the software. During the testing and operational phases, it becomes possible to assess both the internal quality and the external quality, i.e. the extent to which the software satisfies the specified requirements. Finally, turning back to the initial user needs, quality in use is the extent to which the software really helps users to fulfill their tasks.

The intended context of use thus appears to have considerable influence on the quality model that is applied to evaluate software. This influence was acknowledged in the case of HLT by the EAGLES evaluation working group (1996) and, in a specific domain, by the related TEMAA project (TEMAA, 1996: chap.4). However, no domain-independent proposal was put forward for a standardized formal representation of this influence, despite the following important, non-normative examples in the annexes of two ISO/IEC 14598 standards.

The first example concerns the link between the desired integrity of the evaluated software and the activities to be done for evaluation, in particular the choice of a quality model (ISO/IEC, 1999b: Annex B, p.21-22). For instance, if the risk from software malfunction is important (high desired integrity), then more evaluation activities are required. The importance of the six uppermost quality characteristics must be adjusted as well: for low desired integrity, functionality is the most important general characteristic and maintainability the least; whereas for high integrity, reliability becomes the most important and portability the least. Examples of external metrics (one for each quality characteristic) and of acceptance criteria are provided as well.

The second ISO/IEC example introduces the notion of evaluations levels, from most to least critical, concerning four classes of threats: to environment, to people's safety, to the economy, and to data security (ISO/IEC, 1998: Annex B, p.22-25). Again, more demanding evaluation techniques should be used if more critical evaluation levels are required.

The 9126 and 14598 series of ISO/IEC standards are currently being restructured under the SQuaRE model – *Software Quality Requirements and Evaluation* (Azuma, 2001). The factors that influence the definition of quality requirements will be the subject of the 2503n series, the "Quality Requirements Division", which is still under way. Preliminary indications show that our proposal is compatible with the current ISO/IEC considerations on quality requirements (Suryn, Abran & April, 2003).

3. Context of Use and Qualities

The following framework is a formal representation of quality models, of the contexts of use, and of the relation between them, which applies to software evaluation in domains where a somewhat large range of quality metrics is available. In this case, evaluators need to choose the most important metrics given the intended use of the software – since it would be impossible to carry out all the possible tests, and, even if it weren't, the scores would have to be weighted according to some classification of the importance of the metrics. The present framework is a generalization of a previous proposal in the field of machine translation evaluation (see Section 5), applicable also in other complex domains of HLT, where many metrics co-exist (see Section 6).

3.1. Vector-space Representation of Contexts and of Quality Models

In previous work (Hovy, King & Popescu-Belis, 2002: p.55-56), we argued that a quality model could be represented abstractly by a linear averaging function applied to the scores provided by quality metrics chosen from a domain-specific hierarchy. Null coefficients in this function are attributed to metrics that are not part of the quality model. Conversely, the higher the coefficient, the greater the importance of the metric in the quality model. If a final score is averaged from the evaluation results, then the higher the coefficient, the higher the weight of the respective score in the average. Since a quality model for a given domain and evaluation is a hierarchy of characteristics, sub-characteristics and attributes, its representation as a vector is obtained by a pre-order traversal of the leaves of the taxonomy.

Similarly, the context of use or a set of user requirements can also be described as a list of features which characterize the specific task of the evaluated system, the type of users, and the type of input to the system. In the corresponding context vector, the positions corresponding to the features that apply to the particular context are '1' or 'true' and the others '0' or 'false', if a Boolean representation is used. Using numeric representation would allow the coding of the importance of each context feature with respect to the others.

The quality vector can be Boolean if it simply consists of a list of relevant qualities, or numeric, if the list is weighted. In the latter case, the weights are those used in the linear assessment or averaging function that generates the final score, i.e. they encode the contribution of the score obtained by the system for each measured quality attribute to the overall score of the system, if such a unique score is desired. It is still the subject of investigation as to whether, during the final analysis and reporting phase of a specific evaluation, this kind of weighting is sufficient in itself or whether, additional factors arising during the course of the evaluation may also come into play.

In what follows, for clarity reasons, we will only consider Boolean vectors for context and quality. That is, the context features are simply considered to be 'applicable' or 'not applicable' to a given context of use, and quality attributes are simply 'relevant' or 'irrelevant' within a given quality model. A more complex option under study is the use of a Boolean context vector and a numeric quality vector, that is, with weighted quality attributes and metrics.

3.2. Generic Contextual Quality Model (GCQM)

We propose to relate the vector-space representation of the context characteristics to that of the quality model through the use of a specific matrix. The goal is to express in a straightforward manner how the quality model has to be modulated depending on the intended context of use of the evaluated software. In other words, the goal is to design a procedure that associates to any context vector the most appropriate quality vector based on the previous experience of evaluation experts in that field, indicating the qualities that are relevant to a given context.

We propose to compute the influence of the context of use (vector) on the quality model (vector) using a matrix which we baptize "generic contextual quality model" or GCQM[1]. We call this matrix M, and propose that the quality vector Q corresponding to the context vector C is simply the *matrix product* of M and C.

To clarify, suppose that C is an m-dimensional vector (the hierarchy of possible context characteristics has a total of m leaves) and that the quality vector Q is an n-dimensional vector (there are n quality attributes). Then, M is a matrix of m columns and n rows, which can be Boolean or numeric. If both vectors and the matrix are Boolean, then the matrix product must also be computed in a Boolean way (multiplication meaning 'and' and sum means 'or'). If the matrix is not Boolean, then the product can be computed numerically and the resulting quality vector is necessarily a numeric one.

A simple interpretation of the GCQM matrix is that the values in row i indicates which quality attributes are relevant to the context characteristic i. Namely, if coefficient j in this row i is non-zero, then quality attribute j is relevant to the context characteristic i. Therefore, a GCQM is not *per se* a quality model, but a generic correspondence between all context and all quality characteristics.

[1] Note that the present notion of GCQM has no direct relation with the existing GQM model (Goal – Question – Metric), which is a method used to define quality measurements related to a software project, process, or product (more information at http://sel.gsfc.nasa.gov/website/exp-factory/gqm.htm). The similarity of the two acronyms is totally unintended.

3.3. The construction of a GCQM

The construction of the generic correspondence between contexts of use and qualities is of course a complex task, particular to a given domain. The GCQM should embody the knowledge of evaluation experts, who are able to state the quality characteristics and attributes that are relevant to specific aspects of the context of use.

We propose that each expert is offered the possibility of defining a GCQM matrix (using an appropriate interface, cf. 4.2), and then to average the matrices of a pool of experts into an aggregate GCQM. If Boolean GCQMs are considered, they can be aggregated either using a logical *and* (only qualities selected by all experts are kept), or using an *or* (all selected qualities are kept).

The GCQM is thus the main data structure that embodies the knowledge of the relation between context of use and qualities. This proposal elaborates on a previous algorithm (Hovy, King & Popescu-Belis, 2002), which it reformulates in a more elegant and easier to implement fashion. However, the status of nodes with respect to leaves in the ISO-based taxonomies of context and quality characteristics is still subject to analysis. According to the above description, only the leaves of the taxonomies could appear in the context/quality vectors, while the nodes would be considered only as "representative" of the sets of leaves below them. This abstraction may conflict with the possibility that some quality nodes contain particular metrics, which do not appear in the leaves – therefore more analysis is needed.

4. Access Interfaces and Workflow

The users of the proposed context-dependent evaluation guidelines are definitely not supposed to understand the mechanism outlined above in order to start using them. Even evaluation experts should not bother about matrices and coefficients: they should only enter relevant qualities for relevant context features. Therefore, we propose the following workflow and associated interfaces. We first present separate workflows for evaluators (called "users") and for experts, then outline a unifying perspective for these two categories.

4.1. User's View

The evaluators in search of a quality model need first to properly define the context of use of the system that will be evaluated, then to retrieve a suggested quality model depending on the context features they entered.

The user is presented by the tool with a list of characteristics of the context, among which she has to select the ones that describe the intended context of use. Once the context is defined, clicking on a "submit" button in the interface displays the relevant quality attributes (based on the internal computation of Q = M x C by the tool), which could for instance be highlighted among all the possible qualities. At this point, the user has to choose the metric(s) that she wants to apply to each attribute, if several metrics are proposed by the general quality model – a stage that seems difficult to automate. However, this selection can be done based on descriptions and comments attached to the metrics, and depends also on the resources available to the evaluator. Once metrics are selected, the result constitutes the draft evaluation plan, which can be refined and tailored by the user to suit her own needs even more precisely.

4.2. Expert's View

Experts are required to define links between context and qualities, i.e. relevant qualities for each characteristic of the context, which are embodied in a GCQM. It is clearly not the experts' task to figure out how the matrix has to be filled – an assistant interface must be proposed.

The expert's workflow requires the expert to work on each context characteristic separately. The first stage is the choice of a context characteristic, which is done using an interface similar to the one designed for the general user, except that only one context characteristic at the time can be selected. Then, the whole list of quality characteristics and attributes is displayed, allowing the expert to select the relevant qualities for the context feature that was selected. The expert can then save these links, and proceed to another context characteristic, immediately or in a later session[2]. Once several experts have entered GCQMs, these can be averaged in a reference GCQM which can be used by regular users (evaluators), or even "tuned" as shown in the following section.

4.3. Unified View

It appears from the two previous sections that the general user (evaluator) and the expert share part of the workflow and interfaces, with the main difference that experts are asked to define a quality model (working on context characteristics one by one), while users are presented as a result with a proposed quality model.

We believe that the possibility for an informed user to tune or adjust a quality model should also be taken into account, beyond the aforementioned choice of metrics, including the option to include or not certain quality attributes. This makes the distinction between users and experts less obvious. We propose therefore to allow both user and experts to adjust quality models, following a common workflow described in Figure 1 below. Of course, being able to adjust and to save a quality model does not necessarily mean that the result will be validated as expertise and stored into the guidelines: in technical terms, the GCQM saved by a user is not necessarily stored in the global repository used for global averaging.

The common workflow starts with the selection of the GCQM to be used: non-expert users select the reference one (the average of the individual GCQMs entered by experts), while an expert can start with a blank GCQM or continue working on a previously saved one. Users/experts are then prompted to select contextual features: as many as needed to characterize the actual content of use for the user, and in principle only one at the time for the expert. When proceeding to the quality model resulting from the matrix product, users are presented with the quality model corresponding to their context, while experts are shown a blank quality model to which they *must* add qualities.

However, if they find it useful, users can also add/remove qualities from the model they received. In this case, a pop-up window will prompt them to specify the context characteristic to which the added quality is relevant. If the quality model is thus modified,

[2] The identity of the expert is preserved via the personal GCQM file that stores all the links previously entered. When starting a new session, the expert only needs to reload her GCQM file into the interface.

users/experts can save the "tuned" GCQM, and add it to the global pool if allowed to do so. Users can also save/print the result: context, quality model, and metrics.

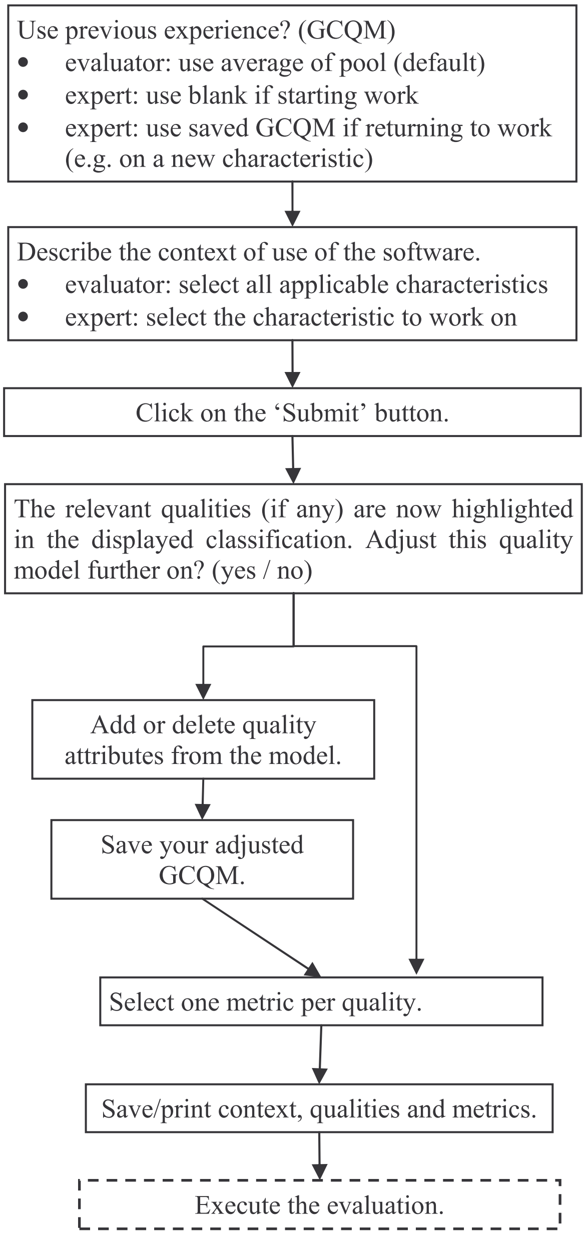

Use previous experience? (GCQM)
• evaluator: use average of pool (default)
• expert: use blank if starting work
• expert: use saved GCQM if returning to work (e.g. on a new characteristic)
Describe the context of use of the software.
• evaluator: select all applicable characteristics
• expert: select the characteristic to work on
Click on the 'Submit' button.
The relevant qualities (if any) are now highlighted in the displayed classification. Adjust this quality model further on? (yes / no)
Add or delete quality attributes from the model.
Save your adjusted GCQM.
Select one metric per quality.
Save/print context, qualities and metrics.
Execute the evaluation.

Figure 1: Integrated workflow for context-based evaluation: users and experts

This unified view enables the input of expertise from previous evaluations. Such evaluations must first be analyzed in terms of context characteristics and qualities. Then, the context characteristics can be entered, and a default quality model (using the current GCQM) can be computed. Based on the comparison with the qualities/metrics used in the past evaluation, the quality model could be tuned, and the result can be saved into a new GCQM, which can be added to the global GCQM pool.

5. FEMTI: an Implementation of the Model for the Evaluation of MT

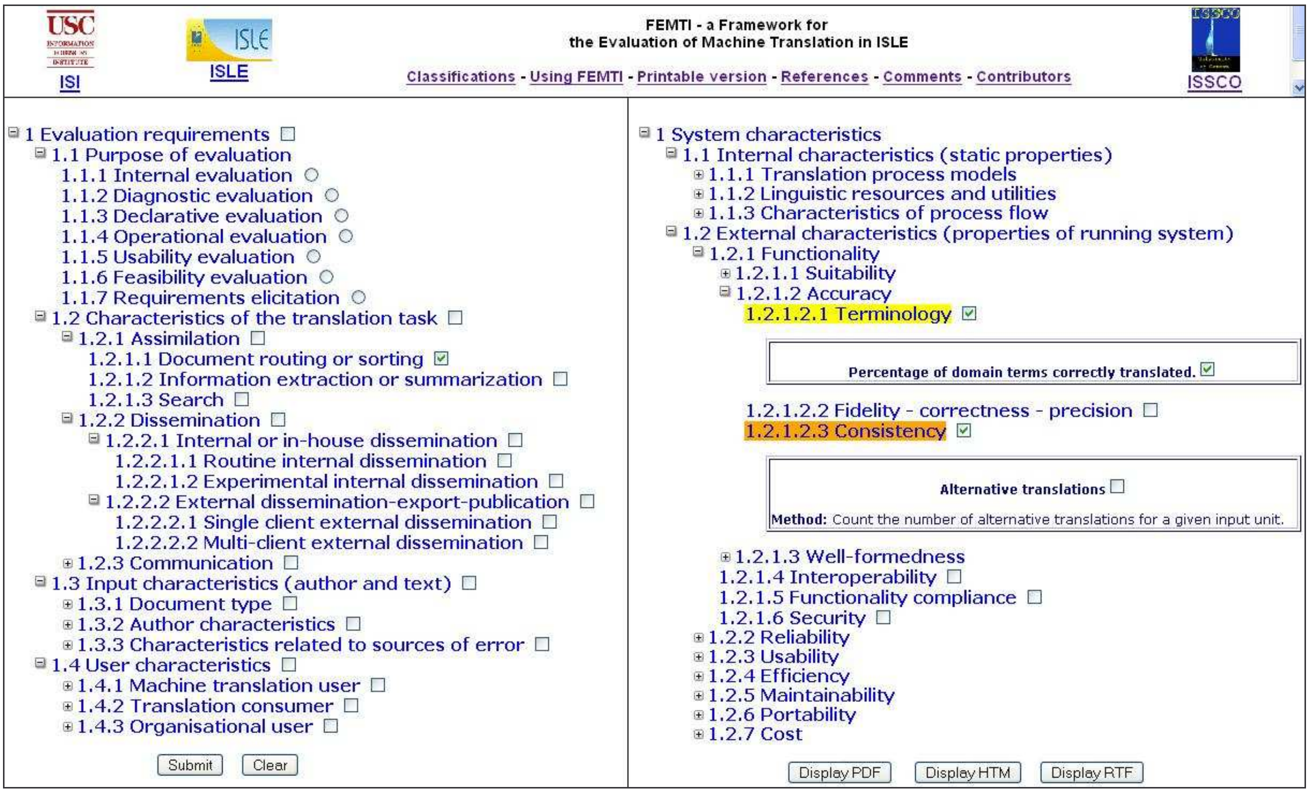
FEMTI, the Framework for the Evaluation of Machine Translation in ISLE (Hovy, King & Popescu-Belis, 2002) is rooted in more general considerations of HLT evaluation put forward in the EAGLES project (King & Maegaard, 1998), in relation to ISO/IEC standards on software evaluation. Given that machine translation (MT) systems fall under the scope of the ISO/IEC guidelines for software evaluation, it is natural that the FEMTI guidelines particularize them.

MT is a domain with a rich range of quality characteristics and metrics. In the realm of quality models for MT software, functionality plays the leading role, especially through two quality attributes generally called fluency (the capacity to produce lexically and syntactically well-formed sentences) and fidelity (the capacity to preserve the meaning of the source text). System developers and real-world users often add other quality attributes, notably price, system extensibility, or coverage. For example, the OVUM report (Mason & Rinsche, 1995) includes usability, customizability, application to entire translation process, language coverage, terminology building, and documentation. In fact, as discussed by Church and Hovy (1993), for some real-world applications, functionality-related attributes may even take a back seat to these sorts of factors. Among the direct forerunners of FEMTI, the work of the Japanese Electronic Industry Development Association (JEIDA) argued that user needs are essential in assessing the quality and usefulness of an MT system (Nomura & Isahara, 1992a; 1992b).

5.1. Contents of FEMTI

FEMTI emphasizes the central influence of the context of use of an MT system on the qualities that should be measured in order to evaluate the system. FEMTI is intended to help evaluators construct a quality model based on the expected context of use of a particular MT software. The 2003 version of FEMTI is a freely available web-based resource (http://www.issco.unige.ch/projects/isle/femti/) which was implemented through a large scale cooperative effort involving a significant part of the MT evaluation community. The website created by the ISLE Evaluation Work Group acknowledges all the contributions to FEMTI (see the "About FEMTI" section). Originating in Hovy's (1999) hierarchical representation of both context and quality characteristics of MT systems, FEMTI is made of two interrelated classifications or taxonomies, informally called part I and part II.

The first taxonomy enables evaluators to define the intended context of use of the MT system(s) that must be evaluated, or, in other words, a set of user requirements. The main aspects to be considered here are the type of user of the MT system, the type of task, and the nature of the input to the system. Initially, the purpose of evaluation must also be taken into account.



ISI ISLE

FEMTI - a Framework for
the Evaluation of Machine Translation in ISLE

Classifications - Using FEMTI - Printable version - References - Comments - Contributors

ISSCO

⊟ 1 Evaluation requirements □
- ⊟ 1.1 Purpose of evaluation
  - 1.1.1 Internal evaluation ○
  - 1.1.2 Diagnostic evaluation ○
  - 1.1.3 Declarative evaluation ○
  - 1.1.4 Operational evaluation ○
  - 1.1.5 Usability evaluation ○
  - 1.1.6 Feasibility evaluation ○
  - 1.1.7 Requirements elicitation ○
- ⊟ 1.2 Characteristics of the translation task □
  - ⊟ 1.2.1 Assimilation □
    - 1.2.1.1 Document routing or sorting ☑
    - 1.2.1.2 Information extraction or summarization □
    - 1.2.1.3 Search □
  - ⊟ 1.2.2 Dissemination □
    - ⊟ 1.2.2.1 Internal or in-house dissemination □
      - 1.2.2.1.1 Routine internal dissemination □
      - 1.2.2.1.2 Experimental internal dissemination □
    - ⊟ 1.2.2.2 External dissemination-export-publication □
      - 1.2.2.2.1 Single client external dissemination □
      - 1.2.2.2.2 Multi-client external dissemination □
  - ⊞ 1.2.3 Communication □
- ⊟ 1.3 Input characteristics (author and text) □
  - ⊞ 1.3.1 Document type □
  - ⊞ 1.3.2 Author characteristics □
  - ⊞ 1.3.3 Characteristics related to sources of error □
- ⊟ 1.4 User characteristics □
  - ⊞ 1.4.1 Machine translation user □
  - ⊞ 1.4.2 Translation consumer □
  - ⊞ 1.4.3 Organisational user □

Submit Clear

⊟ 1 System characteristics
- ⊟ 1.1 Internal characteristics (static properties)
  - ⊞ 1.1.1 Translation process models
  - ⊞ 1.1.2 Linguistic resources and utilities
  - ⊞ 1.1.3 Characteristics of process flow
- ⊟ 1.2 External characteristics (properties of running system)
  - ⊟ 1.2.1 Functionality
    - ⊞ 1.2.1.1 Suitability
    - ⊟ 1.2.1.2 Accuracy
      - 1.2.1.2.1 Terminology ☑
        - Percentage of domain terms correctly translated. ☑
      - 1.2.1.2.2 Fidelity - correctness - precision □
      - 1.2.1.2.3 Consistency ☑
        - Alternative translations □
        - **Method:** Count the number of alternative translations for a given input unit.
    - ⊞ 1.2.1.3 Well-formedness
    - 1.2.1.4 Interoperability □
    - 1.2.1.5 Functionality compliance □
    - 1.2.1.6 Security □
  - ⊞ 1.2.2 Reliability
  - ⊞ 1.2.3 Usability
  - ⊞ 1.2.4 Efficiency
  - ⊞ 1.2.5 Maintainability
  - ⊞ 1.2.6 Portability
  - ⊞ 1.2.7 Cost

Display PDF Display HTM Display RTF

Figure 2: FEMTI's redesigned interface, with a sample context characteristic checked and two quality attributes with metrics

The second taxonomy lists the MT software quality characteristics as hierarchies of sub-characteristics, with internal and/or external quality attributes at the bottom level. The upper levels match the ISO/IEC 9126 characteristics, while the lower levels are made of MT-specific attributes. For each attribute, definitions and references to the metrics used by the community are also provided.

The most original aspect of FEMTI, fully implemented only in its most recent version (see Figure 2 and the next section), is a mapping from the first part to the second part, which states the quality characteristics, sub-characteristics and metrics that are relevant to each feature of the context of use. For instance, as shown in Figure 2, 'terminology precision' (an attribute of functionality in part II) is an important quality for MT systems aimed at 'document routing / sorting' (a context of use from part I). When the links for all the context characteristics from part I that apply to a given context are followed, the result is a set of quality attributes from part II, which constitutes a quality model. Although defined from a theoretical point of view using a GCQM, as explained above, these links from part I to part II are not fully worked out yet.

5.2. FEMTI: Recent Implementation Updates

An important goal in the development of FEMTI is its usability, for both users and experts. Therefore, the most recent version provides new functionalities and a more interactive interface for evaluators, as shown in Figure 2. An interface for experts is also under development. Although some features of FEMTI's new implementation are currently under development, a test version is already visible at the new URL: http://www.issco.unige.ch/femti.

One of the most innovative changes since the 2003 version is the use of a dynamic document server architecture named Cocoon[3], which generates web pages on-the-fly from the XML files that contain the information related to context and quality characteristics. The combination of XSP extensible server pages, XSL stylesheets and Javascript allows the generation of expandable hierarchies, and the behind-the-scenes processing of the forms, including the vectorial representation of contexts of use and quality models, and the matrix product involving the GCQM.

Evaluators can define and submit their context of use through the corresponding hierarchy, visible in the left-hand frame of the interface in Figure 2. Then, the qualities related to the specific context appear highlighted in part II (right-hand frame of the interface in Figure 2). After selecting qualities and metrics the model can be saved in HTML, PDF or RTF format thanks to Cocoon's XSLT and XSL-FO processors.

The linking mechanism implementing the aforementioned connections between parts I and II is also operational: as explained, this mechanism computes the relevant qualities from part II. The links present are those gathered from the existing FEMTI's content and should be enriched in the future with experts' knowledge.

An expert interface will be provided to input links in individual GCQMs, which will then be combined into a reference GCQM used by evaluators. The management of expert identities and rights, based on individual GCQM data structures, is currently under study, as is the averaging of GCQMs into a unique FEMTI matrix.

[3] An open source system available at http://cocoon.apache.org.

6. Perspectives

The structured approach to context-based evaluation adopted here has proved a great success in the field of MT – possibly the oldest field of language technology with a wealth of experience of previous evaluations to draw on. It seems clear to us that other mature disciplines in the field of language technology could also benefit from such evaluation frameworks. In order to test our model of context-dependent evaluation, the best potential application domains would be those with a large range of qualities and metrics, the importance of which depends on the context of use. Application domains of this sort include at least information retrieval systems, for which a proposal for task-based evaluation was made by Sparck Jones (2001), and dialogue systems. For dialogue systems, while the PARADISE methodology (Walker *et al.*, 1997) remains close to the three ISO/IEC aspects of quality-in-use assessed *in situ*, the more recent ITU-T Recommendation P.851 (Möller, 2004) acknowledges the need for contextual quality models, with internal and external qualities pertaining to spoken dialogue systems.

The general purpose framework we have presented not only supports users/evaluators in choosing appropriate metrics to design a quality model for specific evaluations but also allows experts to collaborate and share their knowledge in a structured way. It essentially provides a way of modeling the three main pillars of any user-centered evaluation: the context in which the system is to be used, quality characteristics of the software (and their associated metrics) and the mapping between the two.

We believe that it will be fruitful to apply the same modeling techniques to the even more complex but much younger field of text mining which is still an emerging technology with little previous experience in user-centered evaluation – see King and Underwood (2006) for a discussion of the issues involved. Our aim is to define a framework which will allow all stakeholders in the evaluation of text mining to share their knowledge and experience.

Acknowledgments

This research is supported by the Swiss National Science Foundation with the grant n. 200021-103318.

References

Azuma M. (2001). SQuaRE: The Next Generation of the ISO/IEC 9126 and 14598 International Standards Series on Software Product Quality. *Proceedings of Escom 2001 (12th European Software Control and Metrics Conference)*, London, UK, pp. 337-346.

Church K. W. and Hovy E. H. (1993). Good Applications for Crummy MT. *Machine Translation*, vol. 8, pp. 239-258.

EAGLES Evaluation Working Group (1996). *EAGLES Evaluation of Natural* Language *Processing Systems*. Center for Sprogteknologi, EAG-EWG-PR.2.

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

Hovy E. H., King M. and Popescu-Belis A. (2002). Principles of Context-Based Machine Translation Evaluation. *Machine Translation*, vol. 17(1), pp. 1-33.

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

ISO/IEC (1999a). *ISO/IEC 14598-1:1999 (E) -- Information Technology -- Software Product Evaluation -- Part 1: General Overview*, Geneva, International Organization for Standardization.

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

ISO/IEC (2001). *ISO/IEC 9126-1:2001 (E) -- Software Engineering -- Product Quality -- Part 1: Quality Model*, Geneva, International Organization for Standardization.

King M. and Maegaard B. (1998). Issues in Natural Language Systems Evaluation. *Proceedings of LREC 1998*, Granada, Spain, vol. 1/2, pp. 225-230.

King M. and Underwood N. (2006). Evaluating Symbiotic Systems: the Challenge. *Proceedings of LREC 2006*, Genoa, Italy, in press.

Mason J. and Rinsche A. (1995). *Translation Technology Products*. Report OVUM Ltd.

Möller S. (2004). A New ITU-T Recommendation on the Evaluation of Telephone-Based Spoken Dialogue Systems. *Proceedings of LREC 2004*, Lisbon, Portugal, vol. V, pp. 1607-1610.

Nomura H. and Isahara H. (1992a). The JEIDA Report on Machine Translation. *Proceedings of AMTA Workshop on MT Evaluation: Basis for Future Directions*, San Diego, CA, USA.

Nomura H. and Isahara H. (1992b). JEIDA's Criteria on Machine Translation Evaluation. *IPSJ SIGNotes Natural Language*, Tokyo, Japan, Information Processing Society of Japan, pp. 107-114.

Sparck Jones K. (2001). Automatic language and information processing: rethinking evaluation. *Natural Language Engineering*, vol. 7(1), pp. 29-46.

Suryn W., Abran A. and April A. (2003). ISO/IEC SQuaRE: the second generation of standards for software product quality. *Proceedings of IASTED 2003*, Marina del Rey, CA, USA.

TEMAA (1996). *TEMAA Final Report*. Center fo Sprogteknologi, Copenhagen, Danemark, LRE-62-070.

Walker M. A., Litman D. J., Kamm C. A. and Abella A. (1997). PARADISE: A Framework for Evaluating Spoken Dialogue Agents. *Proceedings of ACL / EACL'97*, Madrid, Spain, pp. 271-280.