MT Evaluation

Margaret King
ETI/T1M (ex-ISSCO)
University of Geneva

Eduard Hovy
USC Information Sciences Institute,
USA

Benjamin K. T'sou
City University of Hong Kong
Hong Kong

John White
Litton PRC
USA

Yusoff Zaharin
Universiti Sains Malaysia
Malaysia

Abstract

This panel deals with the general topic of evaluation of machine translation systems. The first contribution sets out some recent work on creating standards for the design of evaluations. The second, by Eduard Hovy. takes up the particular issue of how metrics can be differentiated and systematized. Benjamin K. T'sou suggests that whilst men may evaluate machines, machines may also evaluate men. John S. White focuses on the question of the role of the user in evaluation design, and Yusoff Zaharin points out that circumstances and settings may have a major influence on evaluation design.

1 Setting standards for evaluation: Margaret King

Evaluation became a critical issue in the world of machine translation at a very early date, with the publication of the ALPAC report in the mid-60's. Highly controversial already at the time of its publication, the report is still capable of rousing strong feelings today. And of course, there is much about it that might be called into question, from the partiality or impartiality of the Committee responsible for the report through to aspects of the evaluation design itself, such as sample size and the validity or invalidity of the metrics chosen.

But at least two good things must be said about the ALPAC report. First, it heightened, almost painfully, awareness of the importance of evaluation. Secondly. it did contain a serious attempt to produce a well designed evaluation.

Since then, there have been any number of MT evaluations. Indeed, it was once said that probably more had been spent on attempts to evaluate MT than on research and development in the area. Many of these attempts have been laudable contributions to the discipline of evaluation design, and are mentioned as such in the contributions of the panelists. Nonetheless it remains true that there is no single, generally accepted methodology for designing the evaluation of an MT system.

In the rest of my remarks, I want to prepare a basis for debate by presenting some recent work, carried out be the Evaluation Working Group of the European EAGLES Initiative (EAGLES. 1999). whose main focus is to propose a framework for designing evaluations, not just of MT systems but of human language technology systems in general. The work is closely related to two ISO standards. ISO/IEC 9126 and ISO/IEC 14598. which are both concerned with the even more general question of evaluation of software product.

The basic thrust behind both the ISO work and the EAGLES specialization of it to HLT systems is not to produce evaluations for specific classes of systems worked out to the last detail, but to provide standards and guidelines for constructing specific evaluations. The value of doing so is manifold: the standards and guidelines provide the evaluator with a readily available source of accumulated wisdom, which, if used as a sort of checklist, can help to ensure that nothing is forgotten and that the evaluation is sound; designing specific evaluations following standard guidelines helps to make evaluations more easily comparable: the standards and guidelines provide a framework for thinking critically about past evaluations and about evaluations coming from other sources: and last, but by no means least, sharing a common framework makes possible cooperative work on developing commonly accepted metrics for evaluation and ways of applying them.

1.1 The EAGLES/ISO Evaluation Framework.

The framework can be presented in the form of a seven step recipe for designing an evaluation. After some general remarks, each step is listed and briefly discussed below. An informal example of how the recipe might be applied will follow in the next section.

The overall process of evaluation is the same whether comparing different systems or trying to evaluate a single candidate system or even a system component. The ultimate question is whether the object being evaluated fits with what the customer of the evaluation wants or needs. Notice here that we are talking about the customer *of the evaluation,* who is not necessarily an end user in the conventional sense: in fact, it is probably the normal case that the customer of the evaluation and the end user do *not* coincide. In practice, the initial requirements thus defined may not be set in stone; carrying out the evaluation may cause the needs to be reconsidered, especially in the case where no available system meets all requirements, or where a system provides functionalities of which the evaluator was not aware until after the evaluation started. It is also quite normal for needs to evolve over time. Despite all this, a best effort is made to define requirements before the evaluation begins.

Given those requirements, we must find some way of judging whether a candidate system meets them. General requirements on the system are broken down into requirements on individual system attributes, which in their turn may be further broken down into component attributes. For each terminal attribute, a metric is defined and validated. Each of these attributes is then measured and the results compared with the original requirements to evaluate how well the system fulfills them.

This process can be described as a series of seven steps:

Step One: Why is the evaluation being done?

In this step the evaluator will try to clarify what the purpose of the evaluation is, and to ensure that all parties have the same understanding of the purpose. Furthermore, he will define what exactly it is that is being evaluated. The range of possibilities here is quite large: the evaluation may concern a system as a whole or a component of a system. It may concern a system considered in isolation or a system in a specific context of use. It is important too to define the boundaries of the system. To illustrate the sort of questions being asked here, let us take the example of a machine translation component which is being used as part of a multilingual information retrieval system. Is it just the machine translation component which is being evaluated, or the whole information retrieval system? Are we to take into account that the information retrieval system is destined to be used by administrative staff in the local hospital, or are we more generally concerned with its behaviour over a range of different contexts and subject matters? Does the system to be evaluated include pretty user interfaces or are we mainly concerned with results rather than their presentation? Does the intended user of the system fall within the boundary of the system to be evaluated, or shall be content with, say, using students to stand in for users? All these questions are in their way obvious, but it is all too easy to neglect them, and finding clear answers to them may be time consuming.

Step 2: Elaborate a task model.

Even with the purest of pure research projects. someone is going to use the system and use it for some purpose, if only to find out why the system does not give the results that were hoped for. This step involves defining more exactly what task the system will be used to achieve (what it is *for),* identifying all relevant roles and agents, finding out who will use the system to do what and what kind of people they are.

To exemplify using the multilingual information retrieval system again, we are asking here questions like whether the system will be used by trained librarians who are experts in finding information but who are monolingual, or by students doing research projects, or by some other kind of agent performing some other kind of task.

Step 3: Define top level quality characteristics.

Here we begin to work out what features of the system need to be evaluated. The EAGLES framework makes heavy use of a list of quality characteristics of software product given in the ISO/IEC 9126 standard. The list includes characteristics like functionality, portability, reliability etc., and even though in the latest versions of the standard these top level characteristics are broken down into finer grained sub-characteristics, they still remain at a fairly high level of generality. The main purpose of using the list is in order to have a check list which provides a structure for thinking about what the relevant attributes of the object to be evaluated are.

Not all attributes are of equal importance. For example, in a context of use where time is critical (information retrieval in the operating theatre?) the time behaviour sub-attribute of efficiency may be more important than anything else. In other contexts of use, reliability may be the most important, or some of the functionality sub-attributes.

Step 4: Produce detailed requirements for the object under evaluation, using the information gained in the previous steps as a basis.

It is with this step that the real nitty gritty work of designing an evaluation begins. Just naming an attribute of quality is not enough to define an evaluation. For each attribute identified, a valid and reliable way of measuring how a system fares with respect to that attribute must be found. If no way can be found, then the attribute has to be reconsidered and broken down into attributes which are measurable. It may be the case that finding attributes which are measurable requires the elaboration of a hierarchy of attributes and sub-attributes, which may in theory be indefinitely deep.

An important point to note here is the underlying assumption that there is rarely, if ever, just one attribute which determines whether a system is satisfactory or not.

It is worth noticing too that it is this step which historically has proved the most difficult in MT evaluation. A notorious example is an attribute which might be dubbed "quality of the translation produced". At various times and in various evaluations, this has been broken down into different sub-attributes in an attempt to find something which is objectively measurable – accuracy, fluency, intelligibility, fidelity, information preservation have all served as sub-attributes at one time or another, with differing degrees of satisfaction.

Step 5: Define the metrics to be applied to the system for the requirements produced under 4.

For each attribute defined, a corresponding metric must be defined. That work was begun in step 4, of course, but here becomes more concrete, with both measure and method for obtaining the measure being defined in detail. It is here that questions of experimental design come into play, as do questions of validation of metrics. It may well be that work done in this step causes rethinking of the decisions taken in Step 4.

This step, not surprisingly, is also a traditional sticking point for machine translation evaluation. To illustrate, consider the intelligibility attribute used as a sub-attribute of quality in the ALPAC evaluations. Intelligibility is a very plausible candidate as a sub-attribute of quality. But the metric used involved asking humans to rate translations on a scale where each point was defined through an English description, and the descriptions were sometimes inherently circular, as in “perfectly clear and intelligible” or “hopelessly unintelligible”. As I have argued in King (1996), this could only be a valid metric if there was substantial agreement across a representative community of machine translation users on what counted as intelligible and what as unintelligible. This, sadly, does not seem to be the case.

Once a metric and a method for applying it has been defined, the next issue is to define what counts as a good score, a satisfactory score or an unsatisfactory score given the task model worked out in Step 2, and, especially, to define where are the cut off points.

It is perhaps worth noting that not all metrics necessarily involve applying some sort of test. One attribute likely to be of a very great importance in many cases is the price of the system. This can be discovered simply by consulting a catalogue, or asking for a price quote from the vendor. But the price offers a nice example of deciding on satisfactory scores and cut off points. There is no point whatever in going to all the effort of executing the rest of the evaluation if the maximum price that can be paid is 500 Swiss francs, and the cheapest system costs 20'000.

Step 6. Design the execution of the evaluation.

This step involves developing any test materials needed to support testing, and defining the actual circumstances of the evaluation, such as who will carry out the different measurements when and in what circumstances. It also involves defining what form the results will take.

Step 7: Execute the evaluation.

This final step is the step which is often thought of as being “the evaluation”. It is the point at which measurements are taken, other pertinent data acquired. the results compared to the previously determined satisfaction ratings and summarized in the form of an evaluation report.

1.2 An informal example

This section tries to make the seven step recipe given above a little more concrete by presenting an oversimplified informal example of a fictitious evaluation for a case where a translation agency is considering acquiring a terminology management tool, in order to gain better efficiency and consistency in the terminology which they translate. Only the first five steps are taken into consideration, since the evaluation is fictitious. In real life, of course, the situation would be much more complex and the requirements much more detailed than those presented here.

1. **Why is the evaluation being done?**

- What is the purpose of the evaluation?

To choose the most suitable terminology management tool, for use by both translators and terminologists. Whilst the manager is looking for efficiency and cost savings, the individual translators and terminologists are hoping for a way to make their work more satisfying.

- What exactly is being evaluated?

Terminology management tools which can be accessed via a network.

2. **Elaborate a task model.**

- What is the system going to be used for?

Looking up terms during a translation, storing newly translated terms and ensuring consistency within and across translations.

- Who will use it? What will they do with it? What are these people like?

Technical translators with an average of seven years in translating technical texts from English to French, Spanish and Japanese will use it during translation to look up terms and their translations. The in-house terminologist will use it to build up and organize terminology and validate the accuracy and consistency of the terminology available to translators.

3. **Define top level quality characteristics.**

- What features of the system need to be evaluated? Are they all equally important?

Languages: the tool must be able to support all the relevant languages, otherwise it will be of no use.

Access: how many people can access the tool at one time? What can they do with it?

Size: How many terms and their translations can be stored?

Consistency: Does the tool have facilities for ensuring that for each term only one translation per target language is entered?
Speed: How fast is terminology look-up and up-dating? Whilst look-up and updating should not take an unreasonable amount of time, this characteristic may be not so important as the preceding ones.

4. Produce detailed requirements

Languages: The tool must be able to support all of English. French, Spanish and Japanese writing systems and character sets.
Access: The tool must allow for at least three translators to look up terms at one time. It must not allow different translators to automatically update and thus overwrite translations of existing terms with translations which have not been approved by the terminologist. The tool should allow for different types of access by different users.
Size: The agency wants to be able to store and access up to a million terms in the next five years.
Consistency: The tool should have facilities for ensuring that for each term only one translation per target language is stored. The tool should allow for completely new terms to be added during translation and marked as such to allow the terminologist to approve of them later.
Speed: Terminology look-up and up-dating must be quicker than the current procedure using index cards. However, there could be a trade-off here; if the improvement in consistency is very great (thus reducing the average revision time required) then speed of look-up and up-dating may be less important. This is one of the attributes which needs to be split up into measurable sub-attributes (see below).

5. Devise metrics to be applied to the system

Some metrics (measures and methods) will involve simple inspection of the documentation accompanying the tool, for example, the languages supported or the maximum size of a term base. The acceptable values for the language and the size measures are already determined in the detailed requirements.
In other cases it is advisable not to rely overmuch on the manufacturer's own description. So, for example, to check how many people can access the tool at once and what they are allowed to do requires experimentation with the tool itself. A good score for the number of people who can efficiently work with the data base at any one time would be 8 (since this is the total number of translators employed). A score of less than 3 would be unacceptable.
Other characteristics such as speed must be split up into measurable sub-attributes, and involve a number of different factors which should be taken into consideration: for example, the time it takes to retrieve a term may be affected by the size of the data base, and/or the number of other users working with the system at the same time. Thus we get a set of different measures such as:

a) average time to retrieve a term from a 100'000 term database (single user)
b) average time to retrieve a term from a l00'000 term data base (3 users)
e) average time to retrieve a term from a 100'000 term data base (5 users)

and so on for each of the aspects of system behaviour which interests us.

1.3 Conclusion and acknowledgements.

The previous sections have tried to set out the skeleton of a framework for thinking about how to design an evaluation. Space and time limitations mean that much has been omitted, but it is hoped that the reader will catch enough of the flavour to want to go further.

It would be quite wrong of me to close without thanking my EAGLES colleagues, especially Sandra Manzi from TIM and Nancy Underwood and Bente Maegaard from CST Copenhagen who have contributed immensely to the work on which this contribution is based.

2 Differentiating and Systematizing Evaluation Metrics: Eduard Hovy

2.1 Introduction

In her panel statement, Maghi King outlines a seven-step recipe for carrying out a successful evaluation of language technology (in particular, MT):

1. Specify why the evaluation is being done
2. Elaborate a task model
3. Define the top level quality characteristics
4. Produce detailed requirements for the system under evaluation, on the basis of 2 and 3
5. Devise the metrics to be applied to the system for the requirements produced under 4
6. Design the execution of the evaluation
7. Execute the evaluation

It is noticeable that the first two of these steps require actions that many evaluation studies of the past have not considered necessary. The so-called 'core technology evaluation approach' espoused by DARPA, the funders of MT research in the US during 1990-94, explicitly tried to decontextualize the MT systems under consideration. (The most elegant statements of this view, propounded by George Doddington, were never published, unfortunately, but see [White et al. 94] for the actual measures used.) Their goal was to pinpoint and measure just the 'essence' of MT systems, arguing that interfaces, user assistance techniques, task and environment tuning. etc., were all beside the main point.

One can appreciate both points of view. King's approach is that of the MT system builder, someone who wants her system not to be a theoretical construct, but a

piece of software actually used. DARPA's approach is that of the research funder, someone who wants to maximize the effect of research investment on the aspects of MT that are not also relevant for other software enterprises such as expert systems, database access systems, etc.

I believe that we should design an evaluation framework large enough to accommodate both approaches. We should identify the important facets of the full situated-and-actually-used MT process and group them, as clearly as possible, into (semi-)'independent' sets of focus aspects. We can then associate with each aspect its relative importance, its evaluation measure(s), and other information.

This belief rests on the assumption that one can in fact decompose the whole process. This is clearly not the case. But one can taxonomize the important aspects into increasingly specific and narrowly defined subaspects. In [Hovy 99] I taxonomized them into (at least) two 'independent' areas of concern: **purpose** and **process**. By *purpose* I mean all the aspects of the situated MT process that are affected when one changes the task, the user, the situation, and the input genre. By *process* I mean all the aspects of MT that are involved after one activates the MT system and waits for it to do something. King is concerned with both aspects, and in fact seems to place more weight on purpose, while DARPA is concerned almost exclusively with process, and preferred to evaluate only fully automated MT.

No study to date spans the whole space of focus aspects. The most elaborate scheme I have seen is a study by the EAGLES Working Group on Evaluation of Natural Language Processing Systems [EAGLES 96], which outlines an evaluation method for MT systems that used as point of departure the ISO standard 9126 [ISO 91]. In fact, King, as co-chair of the study group, draws her panel statement from that report to some degree. While the EAGLES report goes further toward identifying many focus aspects, it does not present them with measures in as simple a way as could be done. A person eager to acquire an MT system will read the EAGLES study with pleasure, but will find it hard to put to practical use because of two reasons. First, no direct link is drawn between the user's task and any evaluation measure. As a result, the user does not know which of the measures to apply, and how much importance to give each one. Second, the sheer number of evaluation measures is rather overwhelming. As with the OVUM report [Ovum 95], the user is almost required to become an expert in MT evaluation before being able to make a decision!

Two other studies of multidimensional approaches to MT evaluation that may be noted for their similarity to the ideas proposed. The excellent 1992 JEIDA Survey [Nomura and Isahara 92] plots the user's situation and needs on two radar plots, and then compares the plots to the characteristics of given MT systems. In a plea for making evaluation sensitive to the task and situation. [Church and Hovy 93] argue that even crummy MT can be useful in the right circumstances.

2.2 A Taxonomy of features and Tests

In this paper I outline a step on the path toward a systematized view of MT evaluation as a whole Analogously to the JEIDA and EAGLES reports, this approach provides a range of evaluation aspects and associated measures. However, it organizes the aspects into a taxonomy of increasing specificity, which acts as a kind of sliding scale of complexity. The user is presented with layers of evaluation measures, increasingly finely differentiated: the more he or she cares about some aspect, the deeper and more delicately he or she can characterize his or her wants, and the more specific (and complex) the corresponding evaluation measures become.

The idea is nothing new; it maps out for MT in explicit terms what we all do when we buy a car or any other complex thing. If I care about car interiors, I will ask the salesperson many questions about the upholstery, the dashboard, the seat adjustment levers, etc., and not so many about the engine capacity, number of valves, and fuel consumption. In fact, I may just ask "is the engine good?", and leave my evaluation of that part of the car at that crude level. It is my choice, after all. Someone else may instead ask detailed questions about the car's suspension, if they plan to drive in rough terrain, and give the interior just a passing glance.

With each point in the taxonomy is associated one or more evaluation measures, useful for determining a system's behavior for that aspect at that level of delicacy. To compute a system's score for some aspect, one applies the measures associated directly with that aspect, and optionally (for more detail) proceeds down one level, to apply the measures associated with each of the aspect's children nodes. If desired, one can propagate the average of their scores back up to their immediate ancestor.

In effect, the sliding-scale feature taxonomization allows the user to create his or her own evaluations at arbitrarily detailed levels. The simple procedure is:

1 characterize the translation goals and the operational process(es);

2 starting at the tops of the two taxonomies, proceed downward to locate appropriately detailed points; stop when the distinctions between children points become too detailed to be of interest;

3 at each point chosen on the downward journey, select the evaluation metrics listed there;

4 apply them to the candidate system and record the results.

5 When done, either stop, and evaluate the next system, or (to achieve a simple, less differentiated score) propagate the scores upward, one level at a time, until the desired level is reached.

6 Record the scores at the level reached, and compare the system to others at this level.

Steps 5 and 6 are generally not useful when one is comparing the suitability of several systems for a particular use: one then needs to evaluate them at the same levels of delicacy. When practical reasons make this impossible, steps 5 and 6 can be applied. Steps 5 and 6 are also useful if one wishes to present a somewhat simplified overall impression of the evaluation result. retaining the ability to furnish more details on request.

To give concreteness to the idea of taxonomizing MT evaluations at increasing levels of delicacy, I provide in this section the topmost regions of the two taxonomies of purpose and process. More details, and definitions of the various taxonomy nodes and their associated measures, are provided in [Hovy 99]. Naturally, subsequent studies will suggest extensions and alterations.

1. Purpose
- 1.1. **Assimilation**. Measures: 1. *Domain coverage.* *2. Genre coverage.* 3. *Speed.* 4. *Automation.* *5. Reliability.*
  - 1.1.1. **Doc routing/sorting**. Measures: 1. *Terminology precision. 2. Extensibility.* 3. *Customizability.*
  - 1.1.2. **Extraction / summarization**. Measures: 1. *Semantic fidelity.* *2. Customizability.*
- 1.2. **Dissemination**. Measures: 1. *Syntactic quality.* *2. Semantic fidelity.* 3. *Extensibility.*
  - 1.2.1. **Internal/house dissemination**. Measures: 1. *Speed.*
    - 1.2.1.1. **Routine.** Measures: none.
    - 1.2.1.2. **Experimental.** Measures: 1. *Adaptability.*
  - 1.2.2. **External dissemination/export.** Measures: 1. *Stylistic quality.* 2. *Reliability.*
    - 1.2.2.1. **Single client**. Measures: none.
    - 1.2.2.2. **Multi-client.** Measures: 1. *Cross-document consistency.*
- 1.3. **Conversation**. Measures: 1. *Intelligibility/comprehensibility:* 2. *Dialogue.* 3. *Extensibility:* 4. *Non-textual pragmatic content.* 5. *Reliability:*
  - 1.3.1. **Interactive conversation**. Measures: 1. *Speed.*
  - 1.3.2. **Delayed conversation**. Measures: none.

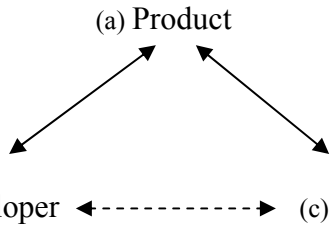
2. **Process**
- 2.1. **Ease of use**. Measures: 1. *Startup effort.* 2. *Normal running effort. 3. Learnability.*
  - 2.1.1. **Portability**. Measures: 1. *Hardware portability:* 2. *Software portability:*
  - 2.1.2. **Maintenance.** Measures: 1. *Maintenance effort.* *2. Vendor support.*
- *2.2* **Interaction.** Measures: 1. *Interface. 2. Tools* 3. *Thesauri and other resources.*
  - 2.2.1. **Inline system assistance**. Measures *1. Extensibility. 2 Documentation. 3. Tools*
  - 2.2.2. **Editing**. Measures: 1. *Text editor.* 2. *Resources.* 3. *House style.*
    - 2.2.2.1. **Pre-editing**. Measures: *1. Style sheets. 2. Writer training.*
    - 2.2.2.2. **Post-editing** Measures: 1. *Problem ident. 2. Learning.* 3. *Editor training. 4. Style sheets.*
    - 2.2.2.3. **No editing**. Measures: none.
- 2.3. **Extensibility**. Measures: *I. Extensibility:* *2. Internal access.* 3. *Documentation.*
  - 2.3.1. **Translator memory.** Measures: 1. *Adaptability: 2. Self-adaptability:*
  - 2.3.2. **Example base**. Measures: 1. *Adaptability: 2. Self-adaptability:*
  - 2.3.3. **Lexicon.** Measures: 1. *Coverage/size* *2. Linguistic expertise.*
  - 2.3.4. **Grammar**. Measures: 1. *Coverage/size* *2. Linguistic expertise.*
  - 2.3.5. **Discourse**. Measures: 1. *Coverage/size* *2. Linguistic expertise.*
  - 2.3.6. **Semantics**. Measures: 1. *Semantics.* *2.Coverage/size. 3.Linguistic expertise*
- 2.4. **System software and platform.** Measures**:** 1. *Cost. 2. Computer resources.* 3. *Software environment.*
  - 2.4.1. **Client-server**. Measures: 1. *Capacity* *2. Network requirements.*
  - 2.4.2. **Standalone**. Measures: 1. *Knowledge sharing.*

3 Man Machine Mutual Evaluation: Benjamin K. T'sou.

3.1 Introduction

Language technology systems or products may be said to be machines which embody the collective creativity of their human developers and could functionally replace human efforts with an anticipated high degree of success.

In terms of the three major components relevant to the evaluation of such systems: (a) PRODUCT. *(b)* DEVELOPER and (c) USER, the most intimate links are between (a) & (b). and (a) and (c). Most developers would claim that they very much have had users in mind. and might have undertaken elaborate pilot testing and even post sale user feedback.



This direct link between the developer and user is in fact part of in-house evaluation, which may be different from user evaluation and assessment.

3.2 Scope of Evaluation

A product could be given an overall simplistic assessment, e.g. good, bad or acceptable (perhaps with qualifications), or one or more specific components in the system could be purposefully evaluated. E.g. the coverage of its bilingual terminological bank, or the speed of processing (King, 1999). Of the five qualitative characteristics: language, access, size, consistency and speed, the first two are readily defined and evaluated by the user. But the remaining three characteristics are variables such that benchmarks can be reset or upgraded after actual usage by the user. The successful developer tends to anticipate such changes.

3.3 Man Evaluating Machine

Evaluation methodology traditionally involved human input, be it Recall and Precision in IR, which have been popular measures, or other forms of statistical measures. For example, comparisons have been typically made between the output of an MT system and translations or judgement by a selected group of human translators, or the output of an automatic summarization system could be compared with abstracts produced by a group of human agents, or judged by them.

One critical concern surrounds what constitutes TYPICAL human behavior (or user) the system should be compared to (or replaced). Very often it hinges on questions relating to numerical saliency in determining what could be TYPICAL or IDEAL.

3.4 Machine Evaluating Man

It may be worthwhile to consider whether there are domains in which a system could be utilized in the evaluation of related human performance. While this approach already exists at low level factory QC work, it is apparently also found in more sophisticated situations where consistency in performance is critically important. It should be noted that this is the case even in the absence of perfect systems. The examples of spell checkers and grammar checkers are good cases in point. They serve to check against human inconsistency in applying what they already know and are capable of doing in the first place but might fail to be consistent in their performance. The domain of language and law is another good example. There is in legal translation or bilingual legal drafting, a critical need to check for regular consistency in fidelity of the bilingual expressions used by different legal draftsmen or translators of legal documents. Any failure could turn out to be costly mistakes in the perversion of justice or in monetary terms. Take another example involving efforts in summarization as shown in Figure 1 below.

There are two graphs , representing different patterns in the selection of salient propositions from a typical Chinese editorial by: (i) a large number of human subjects from the Chinese Mainland, (ii) another large group of human subjects in Taiwan. In fact, an automatic system called CIFAS for summarization being developed by the author and his associates also shows similar but not identical patterns (T'sou *et al.,* 1996). It is clear that while there is reasonable intra-group and inter-group consistency, there are also clear differences. These variations are quite consistent and may be accounted for by cultural differences. What then should be the basis for evaluating the automatic system? When viewed from a different perceptive, this predication can be turned into positive advantage. This is because it is possible to fine tune such a system according to the preferences of either group, and the resultant system can be utilized to check or train human agents in the production of appropriate summaries according to different preferential requirements.

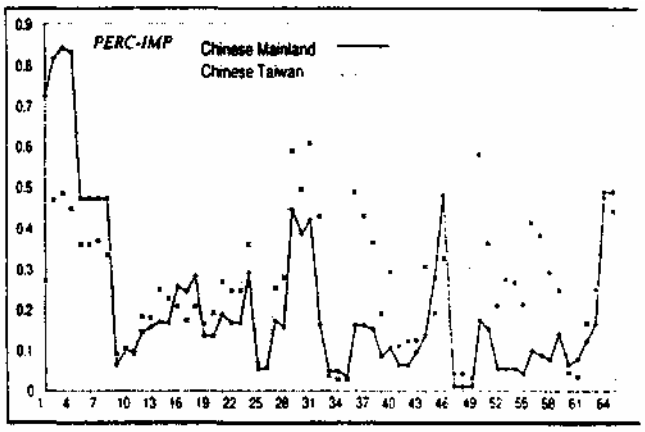


Figure 1: Perceived importance of an editorial judged by Chinese Mainland and Chinese Taiwan groups

For another example, take the large scale and regular verbatim transcription work relating to court proceedings. The current State of Art has not reached 100% accuracy. In fact, in Hong Kong for example, because of the sheer volume of work and the non-readiness of the system [current accuracy about 90%], much manual work is required. In such a case, the QC requirement of less than 5% error rate poses a difficult requirement in terms of certainty about the evaluation. Clearly a comprehensive check is preferred to evaluation by sampling. But the cost will be prohibitively high and impractical. However, human errors from a large range of diverse sources can be profitably and comprehensively checked by a 90% accurate system to provide a cost-effective higher yield. much as is the case with imperfect spell check and grammar check systems.

3.5 Machine Evaluating Machine

Given the existence of complementary systems, such as an MT system which translates from L1 to L2 and another which translates from L2 to L1, it would he entirely

possible to evaluate both systems concurrently by applying them in sequence. In the ideal world, both systems could be judged to be perfect if on completion of the loop, the original text is generated. Examples such as this have been raised in jest, but it may not be that far off for actual cases to take place in future.

4 MT Evaluation and the User: John S. White.

I am grateful for the opportunity to participate in this panel on MT evaluation. I want to respond to one of the challenge propositions, that MT evaluation is meaningless without taking users into account. I will take the position against, though probably only in the letter of the proposition and not in the spirit. We will see that users won't be who they used to be, and that the diversity and complexity of MT evaluation today has much to do with the current state of the art rather than the natural order of things.

I wonder what it is like to evaluate an English spelling checker. Certainly there is some history of this somewhere, and it likely has a legacy of metrics, objectives and controversy. But I am certain that evaluation of spelling checkers is easier than MT evaluation, and I can think of three reasons for that. There are many, maybe most, people standing around who can tell you what the right answer is. There is in fact a right answer, and, finally, spelling correction is a mostly solved problem, for English at least. None of these are true for MT: in MT you need relatively rare expertise to tell you the correspondence between a source and target expression. There are many possible correct correspondences, and we all would concede that MT is not mostly solved for any language pair.

We all know by now that there are many methods of evaluating MT, some good for what they measure, some bad. Some methods measure a particular phenomenon and then have the results extrapolated to overall quality in the advertising. Other methods are more comparable across systems but require lots of time and controlled human judgments. In the midst of all this emerges a consistent view of MT evaluation, namely, that the methods measure different things that different people need to know about MT (Arnold et al. 1993, van Slype 1979, and Vasconcellos 1994). Let's review a few of these.

Feasibility. At the moment when I have decided that some linguistic theory or computational algorithm might have great potential for rendering the strings of one language into the strings of another. I need to be able to do a *feasibility* test of my brilliant new approach. This will involve testing fundamental contrastive phenomena of the source and target, along with test patterns that may demonstrate that whatever works does so because of my theoretical underpinning and not in spite of it.

Internal, Declarative, Comparison. Somewhat later, I need to perform iterative *internal* testing on my new system to make sure that it keeps covering the things I can already cover and begins to cover some new phenomena. If all is successful, a funding agency will want to do some sort of *declarative* testing to see if the coverage I have achieved internally predicts any extensibility to the minute cases in the textual universe. My sponsors may want to *compare* my results with other systems to see if my wonderful idea is really the one that holds the best promise for the future.

Operational. The point here is that all of these types of evaluation are necessary in the life of a development project, and none of them yet involve measuring usability. Clearly, as we move on, we get closer to the direct needs of human consumers. The sort of evaluation that is done when deciding to start using automated translation tools, often called *operational* testing, is almost, but not quite, oriented toward the actual user. It is more a compromise between the desiderata of the end user and the realities of context: cost, expertise, compatibility with the software and hardware dominion already in place, solvency of the vendor, etc.

Usability. Finally, when all the other boxes are checked, it becomes imperative to do *usability* evaluations. These typically have to do with performance issues like rational sequence of steps to a solution, or response time within the level of expectation. It is interesting here that the actual ability of a system to translate has only an indirect effect on usability. A system capable of phenomenal intelligibility and fidelity, but which takes days to translate or requires the invocation of arcane commands will never by used by anyone Meanwhile, a less stellar translation capability that works about as fast as all the usual applications, and which is navigated in the usual way, will be used not only by translators but by everyone (the Systran/Altavista offering an obvious case in point).

Usability is therefore a profoundly important measurement, and so I likely agree with the proposition in spirit. But the current state of MT also requires all those other types of evaluation, with their own objectives and metrics.

Now the connection between this current situation in MT evaluation and my near analogy to spelling checkers has two points. First, I think that if any of the three reasons why MT evaluation is hard actually improved, a smaller set of methods would be sufficient for more stakeholders. Specifically, if MT output ever became so good that it was indistinguishable from a document written in the target language, the internal and declarative types of evaluation could use the same metrics, and the comparative and operational metrics could be much simpler. Less expertise would be required to interpret the measurements to determine suitability for particular stakeholders in the MT process.

The other point of connection to the spell checker analogy has to do with the way we use spell checkers. 1 can remember calling a spelling application that opened a file and dumped a list of unfound words into another file Nowadays, of course, a little red line appears under the word, and I fix it with a click. Present day commercial MT is somewhere in between these two extremes. But we

may be thinking that it is currently closer to the more primitive extreme than it really is. We may also think that that is about where it will stay -- an application run more or less standalone that people approach by stopping whatever they were doing and starting up the translation engine. That day has already passed, of course. As noted, users of various stripes are already presuming to use MT within the common environments and interfaces.

But the integration of MT applications into the information processing stream will be much deeper than this. Whenever possible, MT will happen in the course of collecting information, possibly far in the stream away from the actual user. The current investigations onto cross-lingual information retrieval has just this sort of vision in mind, in which the user cares about the content of the information rather than what form the information was originally in, or any of the processes that made the information available. In this scenario, the user will not interact with MT at all, but with some back-end process like summarization or gisting which took the output of some MT along with many other sources to produce some information the user can then consume.

Here again, the users as we currently conceive of them are not going to be the essential target of MT evaluation in these new contexts. But it turns out that users are the primary providers of evaluative information anyway. In a current effort to develop a new methodology for evaluation based on the suitability of MT output for particular downstream tasks, we depend on the judgments of users who currently perform those tasks in a manual or semi-automatic way. Such task-based evaluation does not measure intelligibility or fidelity directly, nor does it measure usability. It measures the downstream tasks like topic detection, text extraction, and gisting with the metrics pertinent to those tasks, which, when appropriately controlled and benchmarked, lets us predict something about the MT output (Doyon et al., 1999). Ultimately this will allow us to quickly rate MT systems by which downstream tasks they can facilitate (and which they cannot). In the meantime, users who do those downstream tasks will provide us with the both the measures and the diagnostics.

To bring these threads together, I concede that the user is critical to the success of MT both now and in the future. However, there remain several different types of MT evaluation, most of which are only indirectly related to usability. Two things may happen that will change the evaluation picture. MT quality may get better, like spell checkers have, which will make fewer types of evaluation suitable for more of MT's stakeholders. And MT will surely become deeply integrated in end-to-end processes of distilling essential information, meaning that the meaningful evaluations will have to do with whether the user is able to do the task, not whether the MT was faithful, intelligible or usable in its own right. In a way, this is the same as the usability question today, except that the users will be everyone.

5. Evaluation of MT Systems: a Malaysian Experience: Zaharin Yussof

5.1 Introduction

In evaluating Machine Translation (MT) systems, one has to first determine why the evaluation is being done. In ideal situations, the research and development (R&D) content should be evaluated in view of the advancement of the domain. Here, the criteria that come to mind would include the quality of linguistic content, engineering design, software reliability, etc. Such a level of evaluation would best fit fully-automated (FAMT) systems where a substantial amount of R&D is being put in. Nonetheless, the overall usefulness of the system should also be evaluated, where speed, output quality and impact would be measured.

However, since very few FAMT systems are being used for full-scale translation, attention has to be given also to machine-aided human translation (MAHT) systems. In such systems, the evaluation may be more based on pragmatism. R&D content may still be measured, but it would be more towards the system's linguistic data in terms of coverage and accuracy. Evaluation based on engineering design would be more biased towards reliability and man-machine interfacing. An overall level of ease of use would be required to ensure speed and accuracy of results, but perhaps a very important factor to be considered is the operational costs.

This discussion will raise some issues in evaluation of MT systems, in particular within the Malaysian context. The general context is first given, which would explain why certain evaluation criteria have always been chosen despite the many studies and proposals available in the literature worldwide. Then, the various criteria that have been used in various evaluation exercises will be mentioned as well as what improvements the exercises had actually led to. Needless to say, like in many other exercises, the Malaysian experience also falls very far short of solutions.

5.2 The Demand Factor

Malaysia's experience in MT began in 1978 with an FAMT system under a Franco-Malaysian cooperation. Since then, two other FAMT systems had been developed, one fully home-grown and the other under a Malaysian-Japanese cooperation. It has always been accepted that FAMT systems would have to deal with sublanguages, and that it would work best on industrial manuals which not only use restricted languages but are also revised on a regular basis. Unfortunately, in this country the manuals are read in the original language (mainly English) while the major demand is in the translation of text books. The FAMT experiments thus had to be conducted on text books, namely in the domains of chemistry and later computer science. Worse, expectations have always been and have remained very

look-up, thesaurus, affixation, reduplication, spell-checker, etc.) As quality of the output is assumed to be assured by the translator and as ease of use is considered to be subsumed by speed, the evaluation was totally based on the time it takes for the translator to obtain a camera-ready copy. Here, the translators were asked to translate chosen pages either manually or using the system. The pages were chosen to have similar levels of complexity as well as number of words. The results were tabled, cross-referred, etc., with other factors like the time taken to get used to the system, the text type, etc., being (somehow) taken into account. The results did show a more than 50% reduction in time for those well accustomed to the system, but what was interesting was that some were so used to the system that they knew its linguistic content so well that they didn't require the look up tools, which they felt would slow them down. Other subjective matters also came into play, until at the end of the day the organisers did not really know what exactly to infer from of the whole exercise.

An interesting development that came out of that exercise was rather indirect. In Malaysia, usually domain experts are asked to do the translation while language experts are given the job of editing. In general, domain experts tend to be very busy and also very few are very good writers. As such, it took months or even years to get a translation draft submitted, and even then the drafts are often of such very low quality that retranslation of many sections has to be done. With a few experiments, it was found that by switching roles, where language experts do the translation and domain experts do the editing, the overall production time can be reduced very considerably. Drafts can now be obtained within 2-3 months (instead of years) and the whole process can be completed within a year. The role switching system has however not been adopted widely due to objections from various quarters.

5.6 Concluding Remarks

As mentioned in the introduction, the discussion in this paper is exactly what it has been termed, i.e. a discussion. No concrete methodology nor results have been proposed for MT evaluation and none can be seen to be forthcoming in the near future. Apart from strongly agreeing that there is a need for well thought out methodologies for evaluation, the discussion does bring forth certain issues, one of which is that certain local contexts may dictate (or limit) the kind of evaluation criteria to be used. It supports the thesis that evaluation only makes sense if users are taken into account, where 'users' here include in particular the sponsors. In as much as researchers would want to use evaluation to encourage progress in core technologies, in many cases they still need to comply to various demands by users.

References

ALPAC (1996). "Languages and Machines. Computers in Translation and Linguistics." Publication 1416. Automatic Language Processing Advisory Committee. Division of Behavioral Sciences. National Academy of Science. National research Council. Washington D.C.

Arnold, A., L. Sadler, and R. Humphreys. (l993) "Evaluation: an assessment." Machine Translation 8-1/2: 1-24.

Church, K.C. and E.H. Hovy. (1993).. "Good Applications for Crummy MT". In Machine Translation 8 (239-258).

Doyon,J., K. Taylor, and J. White. 1999. "Task-based evaluation for machine translation". MT Summit VII.

EAGLES (1996). The EAGLES Evaluation Working Group: "EAGLES Evaluation of Natural Language Processing Systems: Final Report". EAGLES Document EAG-EWG-PR.2, ISBN 87-90708-00-8 Center for Sprogteknologi, Copenhagen. EAGLES (1999).

http://www.ilc.pi.cnr.it/EAGLES/home.html

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

ISO/IEC (1999a). ISO/IEC 9126: Information Technology— Software Product Evaluation—Quality Characteristics and Guidelines for Their Use. New version, Final Committee Draft.

ISO/IEC (1999b). ISO/IEC 14598-1: Information Technology—Evaluation of Software Products – Part 1: General Guide., Final Committee Draft.

King, M. (1995). "Les belles infideles: Fidelity as a criterion of good translation". In B.H. Partee and P. Sgall (eds.) "Discourse and Meaning", Papers in Honor of Eva Hajicova, John Benjamins, Amsterdam and Philadelphia.

King, M. (1996). "Validity and Evaluation of MT Systems". In H. Somers (ed.), "Terminology, LSP and Translation", Studies in Language Engineering in honour of Juan C. Sager. John Benjamins, Amsterdam and Philadelphia.

King M. (1998a). "Evaluation design: the EAGLES Framework". In Proceedings of the Konvens '98 in Bonn, Gardez! Verlag, St. Augustin.

King, M. (1998b). "Language Resources and Evaluation". In Proceedings of AI&NLP '98, Moncton, Canada.

King. M and Maegaard, B. (1998). "Issues in Natural Language Systems Evaluation". In Proceedings of the First International Conference on Language Resources and Evaluation (LREC), Granada. Spain.

King. M. (1999). The 7-step recipe. Working Document of the EAGLES Evaluation Group. EELS Conference. Hoevelaken. April 1999.

Nomura, H. and J.Isahara. (1992) "JEIDA Report on Machine Translation." In Proceedings of the AMTA Workshop on MT Evaluation. San Diego. (See also [Nomura 92].)

Mason. J. and A. Rinsche. (1995). "Translation Technology Products". OVUM Ltd., London.

T'sou. B.K., Lin. H.L., Ho. H.C., Lai. T.B.Y., and Chan. T.Y.W. (1996). Automatic Chinese full-text abstraction based on rhetorical structure analysis. *Computer Processing of Oriental Languages,* 10, *2,* 225-238.

White, J. et al. (1992-94). ARPA Workshops on Machine Translation. PRC Inc., McLean, VA.

Van Slype, G. (1979). "Critical Methods for Evaluating the Quality of Machine Translation". Prepared for the European Commission Directorate General Scientific and Technical Information and Information Management. Report BR 19142. Bureau Marcel van Dijk.

Vasconcellos, M. (1994). "Apples, oranges, or kiwis? Criteria for the comparison of MT systems." In Vasconcellos, M. (ed). MT Evaluation: Basis for Future Directions. Proceedings of a workshop sponsored by the National Science Foundation. Washington, D.C.: Association for Machine Translation, pp. 37-49.