Association for Computational Linguistics

EACL 2003

10th Conference of The European Chapter

Proceedings of the Workshop on Evaluation Initiatives in Natural Language Processing: are evaluation methods, metrics and resources reusable?

April 14th 2003

Agro Hotel, Budapest, Hungary

Association for Computational Linguistics

EACL 2003

10th Conference of The European Chapter

Proceedings of the Workshop on Evaluation Initiatives in Natural Language Processing: are evaluation methods, metrics and resources reusable?

April 14th 2003

Agro Hotel, Budapest, Hungary

The conference, the workshops and the tutorials are sponsored by:

Chief Patron of the Conference:
Dr. Ferenc Baja
Political State Secretary
Office of Government Information Technology and Civil Relations
Prime Minister's Office

Linguistic Systems BV
Leo Konst (Managing director)
Postbus 1186, 6501 BD Nijmegen, Nederland
tel: +31 24 322 63 02
fax: +31 24 324 21 16
e-mail: info@euroglot.nl, leokonst@telebyte.nl,
http://www.euroglot.nl

Xerox Research Centre Europe
Irene Maxwell
6 chemin de Maupertuis
38240 Meylan, France
Tel: +33 (0)4.76.61.50.83
Fax: +33 (0)4.76.61.50.99
email: info@xrce.xerox.com
website: www.xrce.xerox.com

ATALA
Jean Veronis
Jean.Veronis@up.univ-mrs.fr
45 rue d'Ulm
75230 Paris Cedex 5, France
http://www.atala.org

ELRA/ELDA
Khalid Choukri
choukri@elda.fr
55-57 rue Brillat Savarin
75013 Paris, France
Tel: (+33 1) 43 13 33 33,
Fax: (+33 1) 43 13 33 30
http://www.elda.fr

Order copies of ACL proceedings from: Priscilla Rasmussen, Association for Computational Linguistics, 3 Landmark Center, East Stroudsburg, PA 18301 USA, Phone +1-570-476-8006, Fax +1-570-476-0860, URL http://www.acl-web.org.

INTRODUCTION

Systems that accomplish different Natural Language Processing (NLP) tasks have different characteristics and therefore, it would seem, different requirements for evaluation. However, are there common features in evaluation methods used in various language technologies? Could the evaluation methods established for one type of systems be ported/adapted to another NLP research area? Could automatic evaluation metrics be ported? For instance, could Papineni's MT evaluation metric be used for the evaluation of generated summaries? Could the extrinsic evaluation method used within SUMMAC be applied to the evaluation of Natural Language Generation systems? What are the reusability obstacles encountered and how could they be overcome? What are the evaluation needs of system types such as dialogue systems, which have been less strenuously evaluated till now, and how could they benefit from current practices in evaluating Language Engineering technologies? What are the evaluation challenges that emerge from systems that integrate a number of different language processing functions (e.g. multimodal dialogue systems such as Smartkom)? Could resources (e.g. corpora) used for a specific NLP task, be reused for the evaluation of an NLP system and if so, what adaptations would this require?

Cross-fertilization of evaluation resources has taken place to some extent: in MUC, the extraction-specific adaptation of the standard Information Retrieval precision metric has been accepted as a standard for the evaluation of Information Extraction systems. In SUMMAC, parts of the TREC collection (documents, relevance assessments and even assessment software) have been reused. Both MTEval and SUMMAC have used conceptually similar approaches to evaluation (i.e. subject-based evaluation by testing reading comprehension). Many U.S. and European funding initiatives have been devoted to the evaluation of specific NLP systems, such as: MUC, SUMMAC, TREC and its follow-up initiative CLEF, MTEval and DUC. ISLE, the European initiative for establishing standards in Language Engineering has a working group on the evaluation of Machine Translation systems and its predecessor, EAGLES, has addressed evaluation issues for Language Engineering in general.

The ELSE project (1998-2000) was concerned with the evaluation infrastructure that could be deployed within the scope of the IST Key Actions of the 5th Framework Program of the European Community and indeed, the funding of evaluation activities has been addressed within the 5th Framework (as reported by Mariani and Paroubek 1999). Transatlantic co-operation for the evaluation of Human Language Technologies has also been stressed, among other issues, within an extensive report that was submitted to both the U.S. National Science Foundation and the European Commission's Language Engineering Office in 1999. This report mentions that evaluation techniques in the different Language Engineering areas grow more similar, a fact that emphasizes the need for co-ordinated and reusable evaluation resources.

The time has come to bring together all the above attempts to address the evaluation of NLP systems as a whole and explore ways for reusing established evaluation methods, metrics and other resources, thus, contributing to a more co-ordinated approach to the evaluation of language technology. This is exactly what this workshop has achieved: to bring together leading researchers from various NLP areas (such as Machine Translation, Information Extraction, Information Retrieval, Automatic Summarization, Question-Answering, Dialogue Systems and Natural Language Generation) in order to discuss this topic.

The papers included in this volume address issues of reuse of evaluation resources within and across NLP research areas. We cordially thank the authors and the members of the Programme Committee whose significant contributions made this workshop possible. We are especially grateful to our invited speakers: Donna Harman and Kevin McTait and ELSNET for its support and endorsement.

Katerina Pastra,
April 2003

SPONSORS:

European Network in Human Language Technologies (ELSNET)
(http://www.elsnet.org)

Institute for Language, Speech and Hearing (ILASH), University of Sheffield, UK
(http://www.dcs.shef.ac.uk/research/ilash/)

WORKSHOP ORGANIZATION COMMITTEE:

Katerina Pastra
Department of Computer Science
University of Sheffield
Sheffield, S1-4DP, UK
Phone: +44 114-222-1945
katerina@dcs.shef.ac.uk
http://www.dcs.shef.ac.uk/~katerina

Yorick Wilks
Department of Computer Science
University of Sheffield
Sheffield, S1-4DP, UK
Phone: +44 114-222-1804
yorick@dcs.shef.ac.uk
http://www.dcs.shef.ac.uk/~yorick

SCIENTIFIC COMMITTEE:

Kalina Bontcheva	(University of Sheffield, UK)
Hamish Cunningham	(University of Sheffield, UK)
Rob Gaizauskas	(University of Sheffield, UK)
Donna Harman	(NIST, USA)
Lynette Hirschman	(MITRE, USA)
Maghi King	(ISSCO, Switzerland)
Steven Krauwer	(Utrecht University, The Netherlands)
Inderjeet Mani	(MITRE, USA)
Joseph Mariani	(LIMSI, France)
Patrick Paroubek	(LIMSI, France)
Katerina Pastra	(University of Sheffield, UK)
Martin Rajman	(EPFL - Switzerland)
Karen Spärck-Jones	(University of Cambridge, UK)
Horacio Saggion	(University of Sheffield, UK)
Beth Sundheim	(SPAWAR Systems Center, USA)
Simone Teufel	(University of Cambridge, UK)
Yorick Wilks	(University of Sheffield, UK)

WORKSHOP WEBSITE:

http://www.dcs.shef.ac.uk/~katerina/EACL03-eval/index.html

WORKSHOP PROGRAM

Monday, April 14 2003

8:45-9:00 Welcome

9:00-9:30 *Issues in Reuse of Evaluation Data and Metrics*
Donna Harman (Invited Speaker)

9:30-10:00 *Reuse and Challenges in Evaluating Language Generation Systems: Position Paper*
Kalina Bontcheva

10:00-10:30 *The PEACE SLDS understanding evaluation paradigm of the French MEDIA campaign*
Laurence Devillers, Hélène Maynard, Patrick Paroubek and Sophie Rosset

10:30-11:00 *Coffee Break*

11:00-11:30 *Some statistical methods for evaluating information extraction systems*
Will Lowe and Gary King

11:30-12:00 *A Quantitative Method for Machine Translation Evaluation*
Jesús Tomás, Josep Àngel Mas and Francisco Casacuberta

12:00-12:30 *Colouring Summaries BLEU*
Katerina Pastra and Horacio Saggion

12:30-14:00 *Lunch Break*

14:00-14:30 *Intrinsic versus Extrinsic Evaluations of Parsing Systems*
Diego Mollá and Ben Hutchinson

14:30-15:00 *Adaptation of the F-measure to Cluster Based Lexicon Quality Evaluation*
Angelo Dalli

15:00-15:30 *No-bureaucracy evaluation*
Adam Kilgarriff

15:30-16:00 *Coffee Break*

16:00-16:30 *Living up to standards*
Margaret King

16:30-17:00 *Setting up an Evaluation Infrastructure for Human Language Technologies in Europe*
Kevin McTait and Khalid Choukri (Invited Speakers)

17:00-18:00 *Panel Discussion*

Table of Contents

Issues in Reuse of Evaluation Data and Metrics
Donna Harman 1

Reuse and Challenges in Evaluating Language Generation Systems: Position Paper
Kalina Bontcheva 3

The PEACE SLDS understanding evaluation paradigm of the French MEDIA campaign
Laurence Devillers, Hélène Maynard, Patrick Paroubek and Sophie Rosset 11

Some statistical methods for evaluating information extraction systems
Will Lowe and Gary King 19

A Quantitative Method for Machine Translation Evaluation
Jesús Tomás, Josep Àngel Mas and Francisco Casacuberta 27

Colouring Summaries BLEU
Katerina Pastra and Horacio Saggion 35

Intrinsic versus Extrinsic Evaluations of Parsing Systems
Diego Mollá and Ben Hutchinson 43

Adaptation of the F-measure to Cluster Based Lexicon Quality Evaluation
Angelo Dalli 51

No-bureaucracy evaluation
Adam Kilgarriff 57

Living up to standards
Margaret King 65

Setting up an Evaluation Infrastructure for Human Language Technologies in Europe
Kevin McTait and Khalid Choukri 73

Issues in Reuse of Evaluation Data and Metrics

Donna Harman
National Institute of Standards and Technology
donna.harman@nist.gov

There has been a major explosion in the number of large-scale evaluations in NLP, along with an increasing interest in these evaluations by an expanding number of research groups. An example of this is the growth of the question-answering evaluation in TREC, starting with 20 participating groups in 1999 and now attracting more than 35 groups. Many of these groups are newcomers to these types of evaluations and sometimes also newcomers to research in a given area.

Each of these evaluations leaves behind large sets of evaluation data, which were expensive to create. They also leave behind sets of metrics and testing methodology that represent a huge investment in time by many of the participating research groups. Usually these metrics and methodologies were jointly created based on many discussions among the researchers and the evaluation organizers; additionally they are usually complex and require time for sufficient understanding by all who are involved.

The ability to reuse the data, testing methodology and/or metrics, either for the original testing environment or for a new type of environment is very attractive. In many cases reuse within a given community has been planned for from the beginning and the test sets and metrics become a valuable community resource. Examples of these types of resources are the data and metrics from the MUC evaluations and the TREC ad hoc evaluations.

This talk will examine the issues of reusability both within a given community and across communities. Whereas the benefits of reuse are obvious, there are many subtle problems that might not be clear, especially to people new to research in a given area.

All evaluations are based on many assumptions. For example they may be modelling a specific type of application, molding the data and the metrics to insure that they accurately mirror the intended use of the technology. So using material or metrics from these evaluations for testing in a different application may provide misleading results. Again this problem is often very subtle. Here an example would be using speech transcription data created based on broadcast news to predict how well a given system would perform in a live transcription application (different vocabularies, speaker characteristics, and acoustic devices).

Another danger involves the assumptions built into the scale of the evaluation and the metrics that allow for statistical significance of the results to be tested. Each metric has certain statistical properties that were (hopefully) carefully examined in the original evaluation, and these properties may make a given metric inappropriate in a different evaluation.

Despite all these dangers, it is critical that we get the maximum use of these expensive resources. Workshops such as this one should be encouraged as a way to both share resource information and carefully examine the assumptions built into these resources.

Reuse and Challenges in Evaluating Language Generation Systems: Position Paper

Kalina Bontcheva
University of Sheffield
Regent Court, 211 P ortobello Street
Sheffield S1 4DP, UK
kalina@dcs.shef.ac.uk

Abstract

Although there is an increasing shift towards evaluating Natural Language Generation (NLG) systems, there are still many NLG-specific open issues that hinder effective comparative and quantitative evaluation in this field. The paper starts off by describing a *task-based*, i.e., black-box evaluation of a hypertext NLG system. Then we examine the problem of *glass-box*, i.e., module specific, evaluation in language generation, with focus on evaluating machine learning methods for text planning.

1 Introduction

Although there is an increasing shift towards evaluating Natural Language Generation (NLG) systems, there are still many NLG-specific open issues that hinder effective comparative and quantitative evaluation in this field. As discussed in (Dale and Mellish, 1998), because of the differences between language understanding and generation, most NLU evaluation techniques[1] cannot be applied to generation. The main problems come from the *lack of well-defined input and output* for NLG systems (see also (Wilks, 1992)). Different systems assume different kinds of input, depending on their domains, tasks and target media, which makes comparative evaluation particularly difficult.[2] It is also very hard to obtain a quantitative, objective, measure of the quality of output texts, especially across different domains and genres. Therefore, NLG systems are normally evaluated with respect to their usefulness for a particular (set of) task(s), which is established by measuring user performance on these tasks, i.e., *extrinsic* evaluation. This is often also referred to as *black-box* evaluation, because it does not focus on any specific module, but evaluates the system's performance as a whole. This paper presents one such evaluation experiment with focus on the issue of reusing resources such as questionnaires, and task and experiment designs. It then examines the problem of *glass-box*, i.e., module specific, evaluation in language generation, with focus on the problem of evaluating machine learning methods for text planning.

2 The System in Brief

HYLITE+ (Bontcheva and Wilks, 2001; Bontcheva, 2001b) is a dynamic hypertext system[3] that generates encyclopaedia-style explanations of terms in two specialised domains: chemistry and computers. The user interacts with the system in a Web browser by specifying a term she wants to look up. The system generates a

[1] For a comprehensive review see (Sparck Jones and Galliers, 1996).

[2] The same is not true for understanding tasks since they all operate on the same input, i.e., existing texts. So for example, two part-of-speech taggers or information extraction systems can be compared by running them on the same test corpus and measuring their relative performance.

[3] In *dynamic hypertext* page content and links are created on demand and are often adapted to the user and the previous interaction.

hypertext explanation of the term; further information can be obtained by following hypertext links or specifying another query. The system is based on applied NLG techniques, a re-usable user modelling component (VIEWGEN), and a flexible architecture with module feedback. The adaptivity is implemented on the basis of a user and a discourse models which are used to determine, for example, which concepts are unknown, so clarifying information can be included for them. The user model is updated dynamically, based on the user's interaction with the system. When a user registers with the system for the first time, her model is initialised from a set of stereotypes. The system determines which stereotypes apply on the basis of information provided by the user herself. If no such information is provided, the system assumes a novice user.

3 Extrinsic Evaluation of HYLITE+

Due to the fact that HYLITE+ generates hypertext which content and links are adapted to the user, it can be evaluated following strategies from two fields: NLG and adaptive hypertext. After reviewing the approaches, used for evaluation of the NLG and adaptive hypertext systems most similar to ours,e.g., (Cox et al., 1999), (Reiter et al., 1995), (Höök, 1998), we discovered that they were all evaluated extrinsically by measuring human performance on a set of tasks, given different versions of the system. The experiments were typically followed by an informal interview and/or questionnaire, used to gather some qualitative data, e.g., on the quality of the generated text.

Setting up and conducting such task-based experiments is costly and time-consuming, therefore we looked at opportunities for reusing materials and methodologies from previous evaluation experiments of similar systems from the two fields. This resulted in a substantial reduction of the time and effort needed to prepare the experiments. We also used the findings of some of these experiments in order to improve the design of our own evaluation. For example, (Cox et al., 1999) used pre-generated static pages as a baseline and the study reported that the difference in the two systems' response times might have influenced some of the results. Therefore, we chose instead to have both the baseline non-adaptive and the adaptive systems to generate the pages in real time, which eliminated the possible influence of the different response times.

3.1 Choosing the Main Goals of the Evaluation

The first issue that needs to be addressed when designing the extrinsic, or black-box, evaluation is to determine what are the goals of the experiment. Hypermedia applications are evaluated along three aspects: *interface look and feel*, *representation of the information structure*, and *application-specific information* (Wills et al., 1999). The information structure is concerned with the hypertext network (nodes and links) and navigation aids (e.g., site maps, links to related material, index). The application-specific information concerns the hypermedia content – text, images, audio and video. For our system there is no need to evaluate the interface, since HYLITE+ uses simple HTML and existing Web browsers (e.g. Netscape, Internet Explorer) as rendering tools. Therefore, the evaluation efforts were concentrated on the information content and navigational structure of the generated hypertext.

Information content was measured on the basis of:

- average *time to complete* each task;
- average number of *pages visited* per task;
- average number of *distinct pages* visited per task;
- percent of *correctly answered questions* per task;
- questionnaire results about *content* and *comprehension* of the generated pages;
- *user preference* for any of the systems.

The **navigational structure** was measured by the following metrics:

- average *time per page visited*;
- average *number of pages visited*;
- *total number of pages visited*;

- number of *links followed*;
- usage of the browser Back button;
- usage of the *system's topic list* to find information;
- observation and subjective *opinion on orientation*;
- subjective *opinion on navigation* and ease of finding information.

3.2 Choosing the Methodology

The experiment has a *repeated measures*, *task-based* design (also called within-subjects design), i.e., the same users interacted with the two versions of the system, in order to complete a given set of tasks. Prior to the experiment, the participants were asked to provide some *background information* (e.g., computing experience, familiarity with Web browsers, and electronic encyclopaedia) and fill in a *multiple choice pre-test*, that diagnosed their domain knowledge.

The design of the tasks follows the design used in the evaluation of two other adaptive hypermedia applications – PUSH (Höök, 1998) and (Wills et al., 1999). Each of the participants was first given a set of three tasks – each set contained one browsing, one problem-solving, and one information location task. The order was not randomised, because the browsing task was also intended as a task that would allow users to familiarise themselves with the system and the available information; it was not used for deriving the quantitative measures discussed above.

The participants performed the first set of tasks with the non-adaptive/adaptive system and then swapped systems for the second set of three tasks. The types of tasks – browsing, problem-solving, and information location – were chosen to reflect the different uses of hypermedia information.

Qualitative data and feedback were obtained using a *questionnaire* and *semi-structured interviews*, where the subjects could discuss their experience with the two systems. There were two main types of questions and statements: those related to the usability of the adaptive and baseline systems, e.g., statements like "I found the adaptive system difficult to use"; and those related to hypertext and navigation, e.g., links, text length, structure.

3.3 Results

Due to the small number of participants and the differences in their prior domain knowledge and browsing styles, the results obtained could not be used to derive a statistically reliable comparison between the measures obtained for the adaptive and the non-adaptive versions, but the quantitative results and user feedback are sufficiently encouraging to suggest that HYLITE+ adaptivity is of benefit to the user.

The most important outcome of this small-scale evaluation was that it showed the need to control not just for user's prior knowledge (e.g., novice, advanced), but also for hypertext reading style. Although previous studies of people browsing hypertext (e.g., (Nielsen, 2000)) have distinguished two types: *skimmers* and *readers*, in this experiment we did not control for that, because the tasks from which we derived the quantitative measures were concerned with locating information and problem solving, not browsing. Still, our results showed the need to control for this variable, regardless of the task type, because reading style influences some of the quantitative measures (e.g., task performance, mean time per task, number of visited pages, use of browser navigation buttons). Due to space limitations no further details can be provided in this paper, but see (Bontcheva, 2001a) for a detailed discussion.

3.4 Discussion

The methodology used for HYLITE's black-box evaluation was based on experience not only in the field of language generation, but also in the field of hypermedia, which motivated us to evaluate also the usability of the system and elicit the users' attitudes towards the intelligent behaviour of our generation system. This emphasis on usability, which comes from human-computer interaction, allowed us to obtain results which ultimately had implications for the architecture of our generation system (see (Bontcheva and Wilks, 2001) for further details) and which we would have not obtained otherwise. This leads us to believe that reuse of evaluation resources and methodologies from different, but related fields, can be beneficial for NLP systems in general.

On the other hand, even though evaluating the NLG system in a task-based fashion has had positive impact, there is still a need for glass-box evaluation on a module by module basis, especially using quantitative evaluation metrics, in order to be able to detect specific problems in the generation modules. This is the evaluation challenge that we discuss in the rest of the paper.

4 The Challenge: Automatic Quantitative Evaluation of Content Planners

Content planning, also called deep language generation, is the stage where the system needs to decide *what to say*, i.e., select some predicates encoding the semantics of the text to be generated, and then decide *when to say* them, i.e., choose an ordering of these predicates that will result in the generation of coherent discourse. Typically content plans are created manually by NLG experts in collaboration with domain specialists, using a corpus of target texts. However, this is a time consuming process, so recently researchers have started experimenting with using machine learning for content planning. This is the research area which we will investigate as part of building an NLG system for the e-science Grid project MIAKT[4]. The surface realisation module will be reused from HYLITE+, while the HYLITE+ content planner will be used as a baseline.

An integral part of the development of machine learning approaches to NLP tasks is the ability to perform automatic quantitative evaluation in order to measure differences between different configurations of the module and also allow comparative evaluation with other approaches. For example, the MUC corpora and the associated scoring tool are frequently used by researchers working on machine learning for Information Extraction both as part of the development process and also as means for comparison of the performance of different systems (see e.g., (Marsh and Perzanowski, 1998)). Similarly, automatic quantitative evaluation of content planners needs:

- an annotated corpus;
- an evaluation metric and a scoring tool, implementing this metric.

Below we will discuss each of these components and highlight the outstanding problems and challenges.

4.1 Evaluation Corpora for Content Planning

Research on content planning comes from two fields: document summarisation which uses some NLG techniques to generate the summaries; and natural language generation where the systems generate from some semantic representation, e.g., a domain knowledge base or numeric weather data. Here we review some work from these fields that has addressed the issue of evaluation corpora.

4.1.1 Previous Work

(Kan and Mckeown, 2002) have developed a corpus-trained summarisation system for indicative summaries. As part of this work they annotated manually 100 bibliography entries with indicative summaries and then used a decision tree learner to annotate automatically another 1900 entries with 24 predicates like `Audience`, `Topic`, and `Content`. For example, some annotations for the `Audience` predicate are: `For adult readers; This books is intended for adult readers`. The annotated texts are then used to learn the kinds of predicates present in the summaries, their ordering using bigram statistics, and surface realisation patterns.

(Barzilay et al., 2002) have taken the problem of learning sentence ordering for summarisation one step further by considering multi-document summarisation of news articles. Their experiments show that ordering is significant for text comprehension and there is no *one* ideal ordering, rather there is a set of acceptable orderings. Therefore, an annotated corpus which provides only one of the acceptable orderings is not sufficient to enable

[4] The MIAKT project is sponsored by the UK Engineering and Physical Sciences Research Council (grant GR/R85150/01) and involves the University of Southampton, University of Sheffield, the Open University, University of Oxford, and King's College London.

the system to differentiate between the many good orderings and the bad ones. To solve this problem they developed a corpus of multiple versions of the same content, each version providing an acceptable ordering. This corpus[5] consists of ten sets of news articles, two to three articles per event. Sentences were extracted manually from these sets and human subjects were asked to order them so that they form a readable text. In this way 100 orderings were acquired, 10 orderings per set. However, since this procedure involved a lot of human input, the construction of such a corpus on a larger scale is quite expensive.

The difference between the techniques used for summarisation and those used for generation is that the summarisation ones typically do not use very detailed semantic representations, unlike the full NLG systems. Consequently this means that a corpus annotated for summarisation purposes is likely to contain isufficient information for a full NLG application, while corpus with detailed semantic NLG annotation will most likely be useful for a summarisation content planner. Since the experience from building annotated corpora for learning ordering for summarisation has shown that they are expensive to build, then the creation of semantically annotated corpora for NLG is going to be even more expensive. Therefore, reuse and some automation are paramount.

So far, only very small semantically annotated corpora for NLG have been created. For example, (Duboue and McKeown, 2001) have collected an annotated corpus of 24 transcripts of medical briefings. They use 29 categories to classify the 200 tags used in their tagset. Each transcript had an average of 33 tags with some tags being much more frequent than others. Since the tags need to convey the semantics of the text units, they are highly domain specific, which means that any other NLG system or learning approach that would want to use this corpus for evaluation will have to be retargetted to this domain.

4.1.2 The Proposed Approach for MIAKT

As evident from this discussion, there are still a number of problems that need to be solved so that a semantically annotated corpus of a useful size can be created, thus enabling the comparative evaluation of different learning strategies and content planning components. Previous work has typically started from already existing texts/transcripts and then used humans to annotate them with semantic predicates, which is an expensive operation. In addition, the experience from the Information Extraction evaluations in MUC and ACE has shown that even humans find it difficult to annotate texts with deeper semantic information. For example, the interannotator variability on the scenario template task in MUC-7 was between 85.15 and 96.64 on the f-measures (Marsh and Perzanowski, 1998).

In the MIAKT project we will experiment with a different approach to creating an annotated corpus of orderings, which is similar to the approach taken by (Barzilay et al., 2002), where humans were given sentences and asked to order them in an acceptable way. Since MIAKT is a full NLG system we cannot use already existing sentences, as it was possible in their summarisation systems. Instead, we will use the HYLITE+ surface realiser to generate sentences for each of the semantic predicates and then provide users with a graphical editor, where they can re-arrange the ordering of these sentences by using drag and drop. In this way, there will be no need for the users to annotate with semantic information, because the system will have the corresponding predicates from which the sentences were generated. This idea is similar to the way in which language generation is used to support users with entering knowledge base content (Power et al., 1998). The proposed technique is called “What You See Is What You Meant” (WYSIWYM) and allows a domain expert to edit a NLG knowledge base reliably by interacting with a text, generated by the system, which presents both the knowledge already defined and the options for extending it. In MIAKT we will use instead the generator to produce the sentences, so the user only needs to enter their order. We will not need to use WYSIWYM editing for knowledge entry, because the knowledge base will already exist.

The difference between using generated sentences and sentences from human-written texts is that the human-written ones tend to be more com-

[5] Available at http://www.cs.columbia.edu/ noemie/ordering/.

plex and aggregate the content of similar predicates. This co-occurence information may be important, because, in a sense, it conveys stronger restrictions on ordering than those between two sentences. Therefore we would like to experiment with taking an already annotated corpus of human-authored texts, e.g., MUC-7 and compare the results achieved by using this corpus and a corpus of multiple orderings created by humans from the automatically generated sentences. In general, the question here is whether or not it is possible to reuse a corpus annotated for information extraction for the training of a content planning NLG component.

4.2 Evaluation Metrics

Previous work on learning order constraints has used human subjects for evaluation. For example, (Barzilay et al., 2002) asked humans to grade the summaries, while (Duboue and McKeown, 2001) manually analysed the derived constraints by comparing them to an existing text planner. However, this is not sufficient if different planners or versions of the same planner are to be compared in a quantitative fashion. In contrast, quantitative metrics for automatic evaluation of surface realisers have been developed (Bangalore et al., 2000) and they have been shown to correlate well with human judgement for quality and understandability.

These metrics are two kinds: using string edit distance and using tree-based metrics. The string edit distance ones measure the insertion, deletion, and substitution errors between the reference sentences in the corpus and the generated ones. Two different measures were evaluated and the one that treats deletions in one place and insertion in the other as a single movement error was found to be more appropriate. In the context of content planning we intend use the string edit distance metrics by comparing the proposition sequence generated by the planner against the "ideal" proposition sequence from the corpus.

The tree-based metrics were developed to reflect the intuition that not all moves are equally bad in surface realisation. Therefore these metrics use the dependency tree as a basis of calculating the string edit distances. However, it is not very clear whether this type of metrics will be applicable to the content planning problem given that we do not intend to use a planner that produces a tree-like structure of the text (as do for example RST-based planners, e.g., (Moore, 1995)).

If the reuse experiments in MIAKT are successful, we will make our evaluation tool publically available, together with the annotated corpus and the knowledge base of predicates, which we hope will encourage other researchers to use them for development and/or comparative evaluation of content planners.

5 Conclusion

In this paper we discussed the reuse of existing resouces and methodologies for extrinsic evaluation of language generation systems. We also showed that a number of challenges still exist in evaluation of NLG systems and, more specifically, evaluation of content planners. While other fields like machine translation and text summarisation already have some evaluation metrics and resources available for reuse, language generation has so far lagged behind and no comparative system evaluation has ever been done on a larger scale, e.g., text summarisation systems are compared in the DUC evaluation exercise. As a step towards comparative evaluation for NLG, we intend to make available the annotated corpus, evaluation metric(s) and tools to be developed as part of the recently started MIAKT project.

6 Acknowledgments

The work on MIAKT described here is being supported by the UK Engineering and Physical Sciences Research Council (grant GR/R85150/01). The work on HYLITE+ was supported by a PhD fellowship by the University of Sheffield and an Overseas Research Students Award. I also wish to thank Yorick Wilks and Hamish Cunningham for their comments on this work, the anonymous reviewers who helped me improve the paper, and the human evaluators who participated in the HYLITE+ experiments.

References

Srinivas Bangalore, Owen Rambow, and Steve Whittaker. 2000. Evaluation metrics for generation.

In *International Conference on Natural Language Generation (INLG 2000)*, Mitzpe Ramon, Israel.

Regina Barzilay, Noemie Elhadad, and Kathleen R. McKeown. 2002. Inferring strategies for sentence ordering in multidocument news summarization. *Artificial Intelligence Research*, 17:35–55.

Kalina Bontcheva and Yorick Wilks. 2001. Dealing with dependencies between content planning and surface realisation in a pipeline generation architecture. In *Proceedings of the International Joint Conference in Artificial Intelligence (IJCAI'2001)*, Seattle, USA, August.

Kalina Bontcheva. 2001a. *Generating Adaptive Hypertext Explanations*. Ph.D. thesis, University of Sheffield.

Kalina Bontcheva. 2001b. Tailoring the content of dynamically generated explanations. In M. Bauer, P. Gmytrasiewicz, and J. Vassileva, editors, *User Modelling 2001*, volume 2109 of *Lecture Notes in Artificial Intelligence*. Springer Verlag, Berling Heidelberg.

Richard Cox, Mick O'Donnell, and Jon Oberlander. 1999. Dynamic versus static hypermedia in museum education: an evaluation of ILEX, the intelligent labelling explorer. In Susanne P. Lajoie and Martial Vivet, editors, *Artificial Intelligence in Education: Open Learning Environment: New Computational Technologies to Support Learning, Exploration and Collaboration*, pages 181 – 188. IOS Press, Amsterdam ; Oxford. Papers from the 9th International Conference on Artificial Intelligence in Education (AI-ED 99).

Robert Dale and Chris Mellish. 1998. Towards evaluation in natural language generation. In *Proceedings of First International Conference on Language Resources and Evaluation*, pages 555 – 562, Granada, Spain, 28-30 May.

Pablo A. Duboue and Kathleen R. McKeown. 2001. Empirically estimanting order constraints for content planning in generation. In *Proceedings of ACL-EACL 2001*, Toulouse, France, July.

Kristina Höök. 1998. Evaluating the utility and usability of an adaptive hypermedia system. *Knowledge-Based Systems*, 10:311—319.

Min-Yen Kan and Kathleen R. Mckeown. 2002. Corpus-trained text generation for summarization. In *Proceedings of the Second International Conference on Natural Language Generation (INLG 2002)*.

Elaine Marsh and Dennis Perzanowski. 1998. MUC-7 evaluation of IE technology: Overview of results. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*. `http://www.itl.nist.gov/iaui/894.02/-related_projects/muc/index.html`.

Johanna D. Moore. 1995. *Participating in Explanatory Dialogues*. MIT Press, Cambridge, MA.

Jakob Nielsen. 2000. *Designing Web Usability: The Practice of Simplicity*. New Riders Publishing.

Richard Power, Donia Scott, and Richard Evans. 1998. What you see is what you meant: direct knowledge editings with natural language feedback. In *13th European Conference on Artificial Intelligence (ECAI'98)*, pages 677–681. John Wiley and Sons.

Ehud Reiter, Chris Mellish, and Jon Levine. 1995. Automatic generation of technical documentation. *Journal of Applied Artificial Intelligence*, 9(3):259–287.

Karen Sparck Jones and Julia R. Galliers. 1996. *Evaluating Natural Language Processing Systems: An Analysis and Review*. Number 1083 in Lecture Notes in Artificial Intelligence. Springer Verlag, Berlin, Heidelberg.

Yorick A. Wilks. 1992. Where am I coming from: The reversibility of analysis and generation in natural language processing. In Martin Puetz, editor, *Thirty Years of Linguistic Evolution*. John Benjamins.

G. B. Wills, I. Heath, R.M. Crowder, and W. Hall. 1999. User evaluation of an industrial hypermedia application. Technical report, M99/2, University of Southampton. `http://www.bib.ecs.soton.ac.uk/data/1444/html/html/`.

The PEACE SLDS understanding evaluation paradigm of the French MEDIA campaign

Laurence Devillers, Hélène Maynard, Patrick Paroubek, Sophie Rosset
LIMSI-CNRS
Bt 508 University of Paris XI - BP 133 F-91403 ORSAY Cedex, France
{devil,hbm,pap,rosset}@limsi.fr

Abstract

This paper presents a paradigm for evaluating the context-sensitive understanding capability of any spoken language dialog system: PEACE (French acronym for *Paradigme d'Evaluation Automatique de la Compréhension hors et En-contexte*). This paradigm will be the basis of the French Technolangue MEDIA project, in which dialog systems from various academic and industrial sites will be tested in an evaluation campaign coordinated by ELRA/ELDA (over the next two years). Despite previous efforts such as EAGLES, DISC, AUPELF ARCB2 or the ongoing American DARPA COMMUNICATOR project, the spoken dialog community still lacks common reference tasks and widely agreed upon methods for comparing and diagnosing systems and techniques. Automatic solutions are nowadays being sought both to make possible the comparison of different approaches by means of reliable indicators with generic evaluation methodologies and also to reduce system development costs. However achieving independence from both the dialog system and the task performed seems to be more and more a utopia. Most of the evaluations have up to now either tackled the system as a whole, or based the measurements on dialog-context-free information. The PEACE proposal aims at bypassing some of these shortcomings by extracting, from real dialog corpora, test sets that synthesize contextual information.

1 Introduction

Generally speaking common reference tasks (Whittaker et al., 2002) and methods to compare and diagnose spoken language dialog systems (SLDS) and spoken dialog techniques are lacking despite previous efforts futher discussed in the next section such as EAGLES, DISC, AUPELF ARCB2 or the ongoing American project DARPA COMMUNICATOR. Without an objective assessment of dialog systems, it is difficult to reuse previous work and to advance theories. The assessment of a dialog system is complex in part to the high integration factor and tight coupling between the various modules present in any SLDS, for which unfortunately today, no common accepted reference architecture exists. Nevertheless, a major problem remains the dynamic nature of dialog. Consequently to these shortcomings, researchers are often unable to provide principled design and system capabilities for technology transfer. In other research areas, such as speech recognition and information retrieval, common reference tasks have been highly effective in sharing research costs and efforts. A similar development is highly needed in the dialog community.

In this contribution which addresses only a part of the SLDS evaluation problem, a paradigm for evaluating the context-sensitive understanding capability of any spoken language dialog system is proposed. PEACE (Devillers et al., 2002a) described in section 3, is based on test sets extracted from real corpora, and has three main aspects: it is generic, contextual and it offers diagnostic capabilities. Here genericity is envisaged in a context of information dialogs access. The diagnostic aspect is important in order to determine the different qualities of the systems under test. The contextual aspect of evaluation is a crucial point since dialog is dynamic by nature. We propose to simulate/synthesize the contextual information. The PEACE paradigm will be tested in the French Technolangue MEDIA project and will serve as basis in the comparison and diagnostic evaluation of systems presented by various academic and industrial sites (section 4). ELRA/ELDA is the coordinator of the larger scope evaluation campaign EVALDA, which includes the MEDIA campaign that began in January 2003.

2 Overview of SLDS evaluation

Without an attempt to be exhaustive, we overview some recent efforts for evaluation of SLDS.

The objective of the European DISC project was to write the best-practice guidelines for SLDS development and evaluation of its time. DISC has collected a systematic list of bottom-up evaluation criteria, each corresponding to a partially ordered list of properties likely to be encountered in any SLDS. This properties are positioned on a grid defining an SLDS abstract architecture and relate to various phases of the generic DISC SLDS development life-cycle (Dybkjær and al., 1998). They are complemented by a standard evaluation pattern made of 10 generic questions (e.g. “ Which symptoms need to be observed?”) which has been instantiated for all the evaluation criteria. If the DISC results are quite extensive and presented in an homogeneous way, they do not provide a direct answer to the question of SLDS evaluation. Its contribution lies more at the specification level. Although the approach and the goals of the European EAGLES project were different, one could forward the same remark about the results of the speech evaluation work group (D. Gibbon, 1997). In (Fraser, 1998), one find a set of evaluation criteria for voice oriented products and services, organized in four broad categories.: 1) voice command, 2) document generation, 3) phone services 4) other.

To the best of our knowledge, the MADCOW (Multi Site Data COllection Working group) coordination group set up in the USA by ARPA in the context of the ATIS (*Air Travel Information Services*) task to collect corpora, was the first to propose a common infrastructure for SLDS automatic evaluation (MADCOW, 1992), which also addressed the problem of language understanding evaluation, based on system answer comparison. Unfortunately no direct diagnostic information can be produced, since understanding is appreciated by gauging the distance from the answer to a pair of minimal and a maximal reference answers. In ATIS, the protocol was only been applied to context free sentences. Up to now it has been one of the most used by the community since it is relatively objective and generic because it relies on counts of explicit information and allows for a certain variation in the answers. On the other hand, the method displays a bias toward silence and does not give the means to appreciate error severity.

In ARISE (*Automatic Railway Information Systems for Europe*) (Lamel, 1998), a corpus of roughly 10,000 calls has been used in conjunction with user debriefing questionnaire analysis to diagnose different versions of a phone information server. The hand-tagging objective measures of the corpus include understanding error counts (glass box methodology). Although it provides fine grained diagnostic information, this procedure cannot be easily generalized since it requires hand-annotated corpus and access to the internal representation of the system.

Two metrics have been developped at MIT (Glass et al., 2000): the *Query Density* (QD) and the *Concept Efficiency* (CE), which measure respectively over the course of a dialogue: the mean number of new concepts introduced per user query, and the number of turns necessary for each concept to be understood by the system. Concepts are generated automatically for each utterance with a parsable orthographic transcription as a series of keyword-value pairs. The higher the

QD, the more effectively a user is able to communicate information to the system. The CE is an indicator of recognition or understanding errors; the higher it is, the fewer times a user needs to repeat himself. These metrics were evaluated on single systems (JUPITER and and MERCURY); to compare different systems of the same type, one would need a common ontology. In (Glass et al., 2000), the authors believe that CE should be related to user frustation, but to show it they would need to use the PARADISE framework.

PARADISE (Walker et al., 1998) can be seen as a sort of meta-paradigm which correlates objective and subjective measurements. Its grounding hypothesis states that the goal of any SLDS is to achieve user-satisfaction, which in turn can be predicted through task success and various interaction costs. With the help of the kappa coefficient (Carletta, 1996) proposes to represent the dialog success independently from the task intrinsic complexity, thus opening the way to task generic comparative evaluation. PARADISE has been tested in the COMMUNICATOR project (Walker et al., 2001) with 9 systems working on the same task over different databases. With four basic measures (e.g. task completion) the protocol has been able to predict 37% of user satisfaction variation, and 42% with the help of a few extra measurements on dialog acts and subtasks. One critic, one can make about PARADISE concern its cost (real user tests are costly) and the use of subjective assessment.

The adaption of the DQR text understanding evaluation methodology (Sabatier et al., 2000) to speech resulted in a generic and qualitative procedure. Each element of its test set holds three parts, the *Declaration* to define the context, a *Question* which bears on point present in the context and the *Response*. The test set is organized through seven levels of test, from basic explicit understanding to semantic interpretation and reply pertinence assessment. This protocol is task and system generic but test set construction is not straightforward and the bias introduced by the wording of the question is difficult to assess.

Recently the GDR-13 work group of CNRS on spoken dialog understanding, has proposed an evaluation methodology for literal understanding. According to (Antoine and al., 2002), DEFI tries to remedy two important weaknesses of the MADCOW methodology, namely the lack of genericity and the lack of diagnostic information, by crafting system specific test sets from a primary set of enunciations representative of the task (provided by the developers). Secondary enunciations are then derived from the primary ones in order to exhibit particular language phenomena. Afterwards, the systems are evaluated by their developers using specific test set and their own metrics. The various results can be mapped over a generic abstract architecture for comparison (although this mapping is still unspecified at the time of writing). DEFI has already been used in one evaluation campaign, with 5 systems presented by 4 laboratories. (Antoine and al., 2002) has reported the following weaknesses of the protocol: how to control the bias introduced by the derivation of enunciations, how to guaranty that derived enunciation will remain in the task scope (this prevented some system from being evaluated over the complete test set) and finally how to restrict and organize the language phenomena used in the test set.

3 The PEACE paradigm

We first describe the paradigm and relate preliminary experiments with PEACE. This paradigm which is as basement for the MEDIA project will be refined by all the partners and use for an evaluation campaign between seven systems of industrial and academic sites.

3.1 Description

The PEACE paradigm relies on the idea that for database querying tasks, it is possible to define a common semantic representation, onto which all the systems are able to convert their own representation (Moore, 1994). The paradigm based on data extracted from real corpus, includes both literal and contextual understanding test sets. More precisely, it provides:

- the definition of a semantic representation (see 3.1.1),
- the definition of a model for dialogic contexts (see 3.1.2),

- the definition and typology of linguistic phenomena and dialogic functions used to selectively diagnoze the system language capabilities (anaphora resolution, constraints relaxation, etc.) (see 3.1.3),
- a data structuring method. The format of the annotated data will be adapted to language resource standard annotations implemented (see 3.1.4),
- and evaluation metrics with the corresponding evaluation tool (see 3.1.5).

3.1.1 Generic semantic representation

The difficulty of choosing a semantic representation lies in finding a complete and simple representation of a user utterance meaning in a unified format. A frame Attribute Value Representation (AVR) has been chosen, allowing a fast and reliable annotation. The values are either numeric units, proper names, or semantic classes, that group together lexical units which are synonyms for the task. The order of the (attribute, value) pairs in the semantic representation matches their respective position in the utterance. A modal information (positive (+) and negative(-)) is also assigned to each (attribute, value) pair. The semantic representation of an utterance consists then in a list of triplets of the form (mode, attribute, normalized value). An example is given in figure 1. In order to take into account for long-time dependencies or to allow multiple referenced objects, the semantic representation may be enriched by adding a *reference* value to each triplet for the representation of links between 2 attributes of the utterance.

Attributes can grouped into different classes:

- the **database attributes** (the most frequent) correspond to the attributes of the database tables (e.g. `category` for an hotel);
- the **modifier attributes** are associated to the database concepts. Their values are used to modify the database concept interpretation values (e.g. the attribute `category-modifier` with possible values: >; <, =, `Max`, `Min`);
- the **discursive attributes** are introduced to handle various aspects of dialogic interaction (e.g. `command` with values *cancelation*, *correction*, *error specification...*, or *response* with values *yes* or *no*);
- the **argument attribute** which represents the topic at the focus of the utterance.

User Query	*c'est pas Paris c'est Passy* **it is not Paris it is Passy**
(LU) AVR	(-, place, Paris) (+, place, Passy)

Figure 1: Example of a semantic representation of an utterance with positive and negative information for the ARISE task. `Place` is an database attribute,`Paris` and `Passy` are values and +/- modal markers.

When dealing with information retrieval applications, defining the database and modifier attributes and the appropriate values can be done in a rather straightforward way. Most of those attributes are derived directly from the information stored in the database. Furthermore, most of the discursive attributes are domain-independent. Some database attributes remain unchanged across many tasks, such as those dealing with dates or prices.

This semantic representation has been used at LIMSI for PARIS-SITI TASK (touristic information) and ARISE TASK (traintable information) both with triplet representation. More recently in the context of the AMITIES project, quadruplets were used.

3.1.2 Contextual understanding modeling

Contextual understanding evaluation provides information about the capability of the system to take into account the dialog history in order to properly interpret the user query. Contextual understanding evaluation is rarely performed because of the dynamic nature of the dialog make the dialog context depend on the system's dialog strategy.

Nevertheless PEACE proposes a system-independent way to evaluate local contextual interpretation. Given $U_1...U_t$ the user interactions, and $S_1...S_t$ the answers of the agent or system, the context a time t is a function $f(U_1,S_1,U_2,S_2,...U_t,S_t)$. In the PEACE paradigm, a *paraphrase of the context* is derived from the semantic representation (Bonneau-Maynard et al., 2000).

The dialog contexts are extracted from real dialogs in three steps. First, the internal semantic frames representing the dialog contexts are automatically extracted from the log files of the session recordings. Secondly, the semantic frames are converted into AVR format and then hand-corrected to faithfully represent the dialog history. The last step consists in the writing of a sentence for each context (the context paraphrase), which results in the same AVR representation as the one of the dialog context.

Two possibilities may be investigated for building the paraphrase from the internal semantic representation of the dialog context. A rule-based or template-based natural language generation module can be used to automatically produce the paraphrase. The paraphrase can also be obtained by concatenating the sentences preceding the extracted dialog state. In both cases, a manual verification is needed.

3.1.3 A typology of linguistic phenomena and dialogic functions

For dialog system evaluation, it is essential to build test sets randomly extracted from real corpus. For dialog system diagnosis, it is also crucial to build test sets labeled with the linguistic phenomena and dialogic functions. Thus, the capabilities of system's contextual understanding can be assessed for the main linguistic and dialogic difficulties such as, for instance, anaphora or ellipsis resolution.

3.1.4 A data structuring method

Two types of units, one for literal understanding (LU), the other for contextual understanding (CU) are defined. The format of the annotated data will be adapted to language resource standard annotations implemented in XML, e.g. (Geoffrois et al., 2000), (Ide and Romary, 2002).

Each unit is extracted from a real dialog corpus. LU units are composed of the user query, the corresponding audio signal, an automatic transcription obtained with a recognition system, and finally the literal semantic representation of the utterance (see Figure 1). CU units are composed of the dialog context (given by the paraphrase), the user query and the resulting AVR of the user query in the given context (see Figure 2). Those units are also labeled with linguistic and dialogic phenomena.

Context paraphrase	*je voudrais un hôtel 4* *étoiles dans le neuvième* **I would like a 4 category hotel in the ninth**

`(LU) AVR`	`(+, argument, hotel)`
	`(+, district, 9)`
	`(+, category, 4)`

User query	*la même catégorie dans* *un autre arrondissement* **the same category in another district**

`(LU) AVR`	`(+, other, district)`
	`(+, same, category)`

(CU) AVR	**(+, argument, hotel)** **(-, district, 9)** **(+, category, 4)**

Figure 2: Example of a contextual understanding unit composed of a context paraphrase, a user query and the resulting AVR. AVR of context paraphrase and user query are given in TYPEWRITING MODE. Ellipsis (*"in the ninth"*) and anaphora (*"same category", "another district"*) may be observed.

3.1.5 Evaluation metrics and scoring tool

Common evaluation metrics are essential for analyzing the system capabilities. The scoring tool for AVR comparison is able to compare between two AVR frame representation sets. For evaluation, system outputs translated in AVR format composed one set, the other one contains the AVR references which are manually annotated. Both frame sets have the form of a list of AVRs (fixed length records). Each record is composed of three or four fields (mode, attribute, value, reference). The comparison consists in applying a set of predefined operators each assigned with a cost value. The comparison process looks for operator lists to be applied to the test frame in order to obtain the reference frame that minimizes the final cost value. For a global evaluation, the classical operators from speech evaluation (DELetion, INSertion and SUBstitution) may be used (as used for first two values of `Accuracy` percentage in Table 1).

With our scoring tool the definition of new operators is quite easy. It is then also possible to distinguish between different types of errors by defining specific operators (as used to estimate Topic identification in Table 1), or by using different cost values (for example a substitution is often considered more costly for dialog management).

3.2 Example use of PEACE

In order to validate the evaluation paradigm, a set of approximatively 1,700 literal units and a set of 100 contextual units has been used for the PARIS-SITI task (Bonneau-Maynard and Devillers, 2000). Results for both literal and contextual understanding test sets are given in Table 1. In order to observe the ability of the systems to deal with recognition errors, each literal understanding unit also contains the ASR transcription of the original user utterance. The various measures of understanding accuracy are computed as the ratio between the sum of the number of deleted, inserted and substituted attributes, and the total number of AVR attributes in the test set. The possibility of an automatic evaluation of the LU accuracy and the ability of the scoring tool to point out the errors allowed us to easily improve the literal understanding accuracy from 89.0% to 93.5%. Due to a 26.5% ASR error rate, the LU accuracy goes down from 93.5% to 72% after ASR transcription. The contextual understanding accuracy on the 100 test units is 82.6% on exact transcription. For instance, anaphoric references are relatively well solved, with 80.4% accuracy on the 50 units containing at least one anaphoric reference. For each example, the anaphoric referenced object is generally correctly identified and remaining errors are often due to a bad history constraint management.

3.3 Discussing the PEACE paradigm

The PEACE paradigm enables automatic evaluation of literal and contextual dialog understanding. The evaluation paradigm makes the distinction between different types of errors, allowing a qualitative and diagnostic analysis of the performances of a speech understanding module. Very few evaluation paradigms propose automatic diagnosis of contextual interpretation (Glass et al., 2000). The proposed methodology is based on semi-automatically built reference test sets, and therefore is much more time effective than manual evaluation. Furthermore, it provides reproducible tests.

	#Units	#Attr.	%Acc.	Prec.
LU exact	1 681	3 991	93.5%	0.7
LU ASR .	1 681	3 991	72.0%	1.4
Topic id.	680	833	94.3%	1.6
Modifier id.	323	445	95.7%	1.9
CU exact	100	430	86.8%	3.2
Anaphoric resolution	50	245	84.4%	4.5
Ellipsis resolution	25	106	85.3%	6.7

Table 1: Literal understanding (LU) accuracy on both exact and ASR transcription, and contextual understanding (CU) accuracy. Second column indicates the number of units included in the test set (i.e # of user utterances), third column gives the total number of attributes in the correct AVR test sets. Details, using specific operators, are given for `argument` (topic) and `modifier` identification for LU on exact transcription, and for anaphoric reference and ellipsis resolution for CU. Last column gives the 95% precision of the accuracy estimation (Montacié and Chollet, 1997)

.

Although the semantic representation is task dependent, the example described above shows the feasibility of the paradigm for any dialog system interfacing to a database. Robustness to many linguistic phenomena such as repetitions, hesitations or auto-corrections may be evaluated with this method. XML coding will facilitate the genericity and the reusability of the test sets, by allowing the selection of the dialogic contexts to be studied.

The representation of the dialog context with a single paraphrase, derived from a “flat” structured AVR, may have some limitations in case of long-time dialog dependencies. It does not allow for memorizing all the steps of the dialog. For example, if the speaker says first “*I would like a 2 star hotel*”, then “*no I prefer 3 stars*” and finally says “*give me again my first choice*”, the CU unit cannot take into account this succession of queries. However, this kind of interaction is rarely observed in dialogue corpora: the user usually repeats the constraint value (“*give me again a 2 star hotel*”). To represent more precisely the dialog state, the representation of the dialog context should incorporate some meta-information inspired for example from the DAMSL annotation standard [1] (Devillers et al., 2002b).

Another point is the representativity of the test sets. This may be considered as a limitation as far as PEACE paradigm is built on the idea that the test units are extracted from real dialogs. Obviously, the larger the test sets are, the better. A diagnostic evaluation may need a very large test corpora to validate system performance against the wide range of phenomena present in spontaneous dialog.

The ability to automatically diagnose the performances of contextual understanding modules on local difficulties such as ellipsis, negations, anaphoric reference or constraint relaxation is one of the major advantages of the PEACE paradigm, which has not been investigated by other methodologies. This is why it has been chosen for the MEDIA project described in the next section.

4 The MEDIA project

The MEDIA project proposes a paradigm based on a reference task and on test sets extracted from real corpora for evaluating literal and contextual understanding in dialog systems. The PEACE paradigm will serve as basis for the MEDIA project. The consortium is composed of IRIT, LIA, LIMSI, LORIA, VALORIA for the French academic sites and France Telecom R&D and TELIP for the industrial sites. The scientific committee contains representatives of AT&T (USA), Tilburg University (Netherlands), IBM, IMAG, LIUM and VECSYS (France).

The project has four main parts. First, the selection of reference task such as for example a task of web-based travel agency. The reference task has to correspond to a real-life application allowing real user tests. Secondly, multi-level representation such as the semantic representation, the typology of linguistic phenomena and dialogic functions, the dialog context model... will be commonly refined and adapted to the reference task. The third part deals with the recording and labeling of a dialog corpus which will be used for both system adaptation and test set selection. The last part is the organisation of the evaluation campaigns by ELRA/ELDA for the participating sites.

ELRA/ELDA is the coordinator of a larger scope project: EVALDA which includes among others, the MEDIA project. ELDA with VECSYS will provide transcribed and annotated corpora and evaluation tools according to consortium specifications. The recording of 1200 French dialogs (240 speakers, 5 dialogs each, 15k user queries) is planned. Three sets of LU and CU units will be built from this corpus. A large size adaptation set will be used by the participants to adapt their system to the task and the semantic representation. The development set (around 1K LU (resp. CU) units) will be used to validate the evaluation protocole. The size of the test set is planned to be around 3K LU (resp. CU) units. Various approaches are currently used at the participating sites; stochastic or syntactic and semantic rule-based modeling. The project started in January 2003 and will last two years.

5 Conclusion

Assessing the dialog system understanding capabilities requires to evaluate the transition between successive states of the dialog. At least, we must be able to test a sequence of two states at any point in the dialog. The dynamic and interactive nature of the dialog makes construction and reuse of test sets difficult. Furthermore, to evaluate one particular dialog transition, the system has to be put in a particular state corresponding to the original dialog context. The variable describing the dialog state can be composed of complex information such as the current semantic frame (list of triplets (mode,attribute,value) or quadruples (mode, attribute, value, reference)), the dialog history semantic frame and potentially other information like recognition scores, dialog acts, etc.

The PEACE paradigm allows the evaluation of two successive simplified dialog states. It has been successfully tested with test samples focusing on linguistic difficulties of literal and contextual understanding. For these tests, the dialog state is the dialog history semantic frame. The contextual understanding modeling in PEACE is *system independent* since the context is given by a paraphrase

[1] http://www.cs.rochester.edu/research/trains/annotation

of queries. PEACE allows a *diagnostic evaluation* of specific semantic attributes and *particular linguistic phenomena*.

In our opinion, it is crucial for the dialog community to agree on a *common reference task* and reference test sets in order to be able to compare and diagnose dialog systems. Both evaluation with real users and artificial simulation of successive dialog states using test sets extracted from real corpora have to be carried out in parallel. The use of test sets reduces the global cost of dialog system evaluation, moreover such tests are reproducible.

The PEACE protocol will be used as basis for the French Technolangue MEDIA project in a two year evaluation campaign where dialog systems from both academia and industry will be evaluated. In other domains, it could be related with (Hirschman, 2000) propositions for Question Answering evaluation.

References

J.Y. Antoine and al. 2002. Predictive and objective evaluation of speech understanding: the "challenge" evaluation campaign of the i3 speech workgroup of th french cnrs. In *LREC2002*, Spain, May. ELRA.

H. Bonneau-Maynard and L. Devillers. 2000. A framework for evaluating contextual understanding. In *ICSLP*.

H. Bonneau-Maynard, L. Devillers, and S. Rosset. 2000. Predictive performance of dialog systems. In *LREC2000*, volume 1, pages 177–181, Athens, Greece, May. ELRA.

J. Carletta. 1996. Assessing agreement on classification tasks: the kappa statistics. *Computational Linguistics*, 2(22):249–254.

R. Winski D. Gibbon, R. Moore. 1997. *Handbook of Standards and Ressources for Spoken Language Ressources*. Mouton de Gruyter, New York.

L. Devillers, H. Maynard, and P. Paroubek. 2002a. Méthodologies d'évaluation des systèmes de dialogue parlé : réflexions et expériences autour de la compréhension. In *Traitement Automatique des Langues*, volume 43, pages 155–184.

L. Devillers, S. Rosset, H. Bonneau-Maynard, and L. Lamel. 2002b. Annotations for dynamic diagnosis of the dialog state. In *LREC2002*, Spain, May. ELRA.

L. Dybkjær and al. 1998. The disc approach to spoken language systems development and evaluation. In *LREC1998)*, volume 1, pages 185–189, Spain, May. ELRA.

N. Fraser. 1998. *Spoken Language System Assessment*, volume 3. Mouton de Gruyter, New York.

E. Geoffrois, C. Barras, S. Bird, and Z. Wu. 2000. Transcribing with annotation graphs. In *LREC2000*, volume 2, pages 1517–1521, Greece, May. ELRA.

J. Glass, J. Polifroni, S. Seneff, and V. Zue. 2000. *Data collection and performance evaluation of spoken dialogue systems: the MIT experience*.

Lynette Hirschman. 2000. Reading comprehension and question answering new evaluation paradigms for human language technology. In *LREC2000 Workshop "Using Evaluation within HLT Programs: Results and Trends"*, pages 54–59, Greece, May. ELRA.

N. Ide and L. Romary. 2002. Towards multimodal content representation. In *LREC 2002*.

L. Lamel. 1998. Spoken language dialog system development and evaluation at limsi. In *Actes de l'International Symposium on Spoken Dialogue*, Sydney, Australia, November.

MADCOW. 1992. Multi-site data collection for a spoken language corpus. In *DARPA Speech and Natural Language Workshop*.

C. Montacié and G. Chollet. 1997. Systèmes de référence pour l'évaluation d'applications et la caractérisation de bases de données en reconnaissance de la parole. In *16ème JEP*.

R.C. Moore. 1994. Semantic evaluation for spoken-language systems. In *DARPA Speech and Natural Language Workshop*.

P. Sabatier, Ph. Blache, J. Guizol, F. Lévy, A. Nazarenko, and S. N'Guema. 2000. évaluer des systèmes de compréhension de textes. In *Ressources et Evaluation en Ingénierie Linguistique*, pages 265–275. Chibout K. *et al.* (Eds) Duculot.

M. Walker, D. Litman, C. Kamm, and A. Abella. 1998. Evaluating spoken dialogue agents with paradise: 2 cases studies. *Computer Speech and Language*, 3(12):317–347.

M. Walker, R. Passonneau, and J.E. Boland. 2001. Quantitative and qualitative evaluation of darpa communicatorspoken dialog systems. In *Actes du 39^{me} ACL*, pages 515–522, Toulouse, France, July. ACL.

S. Whittaker, L. Terveen, and B. Nardi. 2002. Reference task agenda for HCI. In *ISLE workshop 2002*.

Some statistical methods for evaluating information extraction systems

Will Lowe
Computer Science Department
Bath University
wlowe@latte.harvard.edu

Gary King
Center for Basic Research
in the Social Sciences
Harvard University
king@harvard.edu

Abstract

We present new statistical methods for evaluating information extraction systems. The methods were developed to evaluate a system used by political scientists to extract event information from news leads about international politics. The nature of this data presents two problems for evaluators: 1) the frequency distribution of event types in international event data is strongly skewed, so a random sample of newsleads will typically fail to contain any low frequency events. 2) Manual information extraction necessary to create evaluation sets is costly, and most effort is wasted coding high frequency categories .

We present an evaluation scheme that overcomes these problems with considerably less manual effort than traditional methods, and also allows us to interpret an information extraction system as an estimator (in the statistical sense) and to estimate its bias.

1 Introduction

This paper introduces a statistical approach we developed to evaluate information extraction systems used to study international relations. Event extraction is a form of categorization, but the highly skewed frequency profile of international event categories in real data generates severe problems for evaluators. We discuss these problems in section 3, show how to circumvent using a novel sampling scheme in section 4, and briefly describe our application. Finally we discuss the advantages and disadvantages of the methods, and their relations to standard evaluation procedure. We start with a brief review of information extraction in international relations.

2 Event Analysis in International Relations

Researchers in quantitative international relations have been performing manual information extraction since the mid-1970s (McClelland, 1978; Azar, 1982). The information extracted has remained fairly simple; a researcher fills a 'who did what to whom' template, usually from historical documents, a list of countries and international organizations to describe the actors, and a more or less articulated ontology of international events to describe what occurred (McClelland, 1978). In the early 1990s automated information extraction tools mostly replaced manual coding efforts (Schrodt et al., 1994). Information extraction systems in international relations perform a similar task to those competing in early Message Understanding Competitions (Sundheim, 1991, 1992). With machine extracted events data it is now possible to do near real-time conflict forecasting with data based on newswire leads, and detailed political analysis afterwards.

3 Event Category Distributions

We wanted to evaluate an information extraction system from Virtual Research Associates[1]. This system bundles extraction and visualization software with a custom event ontology containing, at last count, about 200 categories of international event.

We found two problems with the nature of international events data. First, the frequency distribution over the system's ontology, or indeed several other ontologies we considered, is heavily skewed. A handful of mostly diplomatic event types predominate, and the frequency of other event types falls of very sharply: we ran the system over all the newsleads in Reuters' coverage of the Bosnia conflict, and of the approximately 45,000 events it extracted, 10,605 were in the category of 'neutral comment', 4 of 'apology' and 35 of 'threat of force'. Thus the relative frequencies of event categories in this data can be 2,500 to 1.

Also, as these figures suggest, the more interesting and politically relevant events tend to be of low frequency. This problem is quite general in categorization systems with reasonably articulated category systems, and not specific to international relations. But any dataset with these properties causes an immediate problem for evaluation.

Ideally we would choose a random subset of leads whose events are known with certainty (because we have coded them manually beforehand), run the system over them, and then compute various sample statistics such as precision and recall[2]. However, a small randomly chosen subset is very unlikely contain instances of most interesting events, and so the system's performance will not be evaluated on them. Given the possible frequency ratios above, the size of subset necessary to ensure reasonable coverage of lower frequency event categories is enormous. Put more concretely, to construct a test set of news leads the evaluator will on average have to code around 2,500 comments to reach a single apology and about 300 comments to find a single threat of force.

3.1 Standard Evalution Methods

The standard evaluation methods developed over the course of the Message Understanding Competitions consist mainly in sample statistics to compute over the evaluation materials e.g. precision and recall, but do not give any guidance for choosing the materials themselves (Cowie and Lehnert, 1996; Grishman, 1997). This is just done by hand by the judges. Perhaps because the selection question is neglected, it is seldom clear what larger population the test materials are from (save that it is the same one as the training examples), and as a consequence it is unclear what the implications for generalization are when a system obtains a particular set of scores for precision and recall (Lehnert and Sundheim, 1991).

Since this literature did not help us generate a suitable evaluation sample, we approached the problem from scratch, and developed a statistical framework specific to our needs.

4 Method

One reasonable-sounding but *wrong* way to address the problem of creating a test set without having to code tens of thousands of irrelevant stories is the following:

1. Use the extraction system itself to perform an initial coding,
2. Take a sample of the output that covers all the event types in reasonable quantities,
3. Examine each coding to see whether the system assigned the correct event code.

This looks like it can guarantee a good sample of low frequency events at much lower cost to the manual coder; we can just pick a fixed number of events from each category and evaluate them. However, this method exhibits *selection bias*. To see this, let M and T be variables indicating which event category the $\underline{\text{M}}$achine (that is, the information extraction system) codes an event into, and the $\underline{\text{T}}$rue category to which the event actually belongs. Statistically, the quantity of interest to us is the probability that the machine is *correct*:

$$\mathrm{P}(M = i \mid T = i) \tag{1}$$

[1] `http://www.vranet.com`

[2] This paper only evaluates extraction performance on event types, though there would seem to be no reason why a similar approach would not work for actors etc.

This is the probability that the machine classifies an event into category i given that the true event coding is indeed i. A full characterization of the success of the machine requires knowing $\mathrm{P}(M = i \mid T = i)$ for $i = 0, \ldots, J$, which includes all J event categories and where $i = 0$ denotes the situation where the machine is unable to classify an event into any category. In short, the quantity of interest is the full probability density $\mathrm{P}(M \mid T)$.

In statistical terms, this distribution is a *likelihood function* for the information extraction system. This observation allows us to treat the system like any other statistical estimator and offers the interesting possibility of analyzing generalization via its sampling properties, e.g. its bias, variance, mean squared error, or risk.

Unfortunately, the problem with the reasonable-sounding approach described above is that it does not in fact allow us to estimate $\mathrm{P}(M \mid T)$ because it is implicitly conditioning on M, not T. In particular, the proportion of events that are actually in category i among those the machine *put* in category i gives us instead an estimate of

$$\mathrm{P}(T \mid M) \tag{2}$$

which is not the quantity of interest. (2) is the probability of the truth being in some event category rather than the machine's response whereas in fact the true event category is fixed and it is the machine's response that is uncertain[3]. Worse, $\mathrm{P}(T \mid M)$ is a systematically biased estimate of $\mathrm{P}(M \mid T)$ because these two quantities are related by Bayes theorem:

$$\mathrm{P}(M \mid T) = \frac{\mathrm{P}(M,T)}{\mathrm{P}(T)} = \frac{\mathrm{P}(T \mid M)\mathrm{P}(M)}{\mathrm{P}(T)}, \tag{3}$$

and the only circumstances under which they would be equal is when $\mathrm{P}(M)$ is uniform. But the figures in section 3 suggest that $\mathrm{P}(M)$ is highly skewed.

However this last observation suggests a better method for unbiased estimation of (1).

1. Estimate $\mathrm{P}(T \mid M)$ as described above
2. Compute $\mathrm{P}(M)$ by running the system over the *entire* data set and normalizing the frequency histogram of event categories
3. Estimate $\mathrm{P}(M \mid T)$ by correcting $\mathrm{P}(T \mid M)$ with $\mathrm{P}(M)$ using Bayes theorem

Our implementation of this scheme was to first run the system over 45,000 leads about the Bosnia conflict, and normalize the frequency histogram of events extracted to create $\mathrm{P}(M)$. Then, randomly choose 5 leads assigned to each event category, and manually determine which event type the instantiate. Then normalize to estimate $\mathrm{P}(T \mid M)$. And finally, use (3) to create $\mathrm{P}(M \mid T)$. We chose four times as many uncategorized leads as from each true category in addition. A larger sample here is advisable to see what sort of categories the system misses. These sample sizes are fixed, but it may also be possible to use active learning techniques to tune them (as in e.g. Argamon-Engelson and Dagan, 1999) for even more efficient sampling.

The advantage of this roundabout route to (1) is that it requires many fewer events to be manually coded. We ran the system over 45,000 leads but only manually coded a handful of events for each category. This guaranteed us even coverage of the lowest frequency event categories whilst not biasing the end result – for an ontology with about 200 categories this is a substantial decrease in evaluator effort.

This method works by making use of the extraction system itself to produce one important marginal: $\mathrm{P}(M)$. If we assume that the aim is to evaluate the system on the Bosnia conflict, $\mathrm{P}(M)$ is not estimated, but is rather an exact population marginal[4]. Then we can guarantee that our estimate of $\mathrm{P}(M \mid T)$ is unbiased because the method for estimating $\mathrm{P}(T \mid M)$ is clearly unbiased, and $\mathrm{P}(M)$ adds no error.

4.1 Summary Measures

$\mathrm{P}(M \mid T)$ allows the computation of a number of useful summary measures[5]. For example, we

[3] This is due to changes in the journalist's choice of vocabulary and syntactic construction that are uncorrelated with the identity of the event being described.

[4] We might consider the Bosnian conflict to be a sample point from the larger population of all wars, but that population – if it exists at all – is certainly difficult to quantify.

[5] Detailed discussion of several summary measures for the system we evaluated can be found in King and Lowe (2002).

can easily compute $\mathrm{P}(M,T)$ from quantities already available, so $\sum^J \mathrm{P}(M=i,T=i)$ is the proportion of time the system extracts the correct category. Alternatively, if it is more important to extract some categories than others, then various weighted measures can be constructed e.g. $\sum^J \mathrm{P}(M=i \mid T=i)w_i$ where ws are non-negative and sum to 1, representing the relative importance of extracting each category. Some more graphical methods of evaluation using $\mathrm{P}(M \mid T)$ are presented below.

4.2 Estimator Properties

Given a likelihood function for the extraction system we can investigate its properties as an estimator. It is particularly useful to know the *bias* of an estimator, defined in this case as the difference between the expected category response from the system when the true event category is i, and i itself, where the expectation is taken of repeated information extraction tasks that instantiate the same event categories. We do not examine the corresponding variance here, and a more complete evaluation might also address the question of consistency.

4.2.1 Conflict and Cooperation

The machines response and the true category is best seen as a set of multinomial probabilities (with a unit vector with the value 1 at the index of the system's extracted category or the true category respectively. Estimator properties are cumbersome to represent in this format, so here we map the system's response to a single real value corresponding to the level of conflict or cooperation of the event category. This re-representation is usual in international relations and allows standard econometric time series methods to be applied (Schrodt and Gerner, 1994; Goldstein and Freeman, 1990; Goldstein and Pevehouse, 1997).

For our purposes it also allows the straightforward graphical presentation of the main ideas. We define the level of conflict or cooperation level of an event category i as G_i, a real number between -10 (most conflictual) to 10 (most cooperative) (see Goldstein, 1992, for the full mapping). For example, according to this scheme, when i denotes the event category 'extending economic aid', $G_i = 7.4$, 'policy endorsement' maps to 3.6, 'halt negotiations' maps to -3.8, and a 'military engagement' maps to -10, the maximally conflictual event. The mapping allows univariate, and politically relevant comparison between the true conflict level and that of the event categories the system extracts.

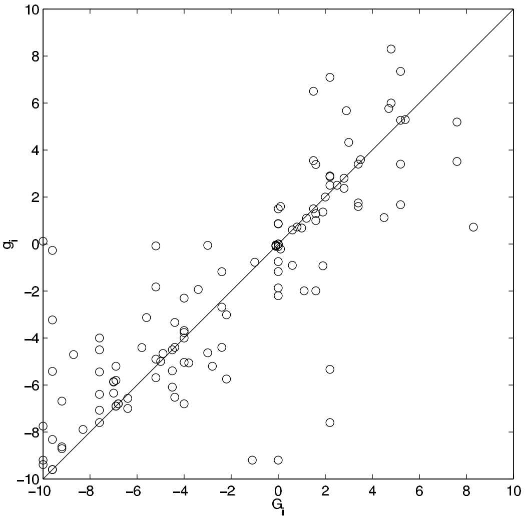


Figure 1: Expected (g_i) versus true (G_i) conflict-cooperation level for each event category.

The expected system response when the true category has conflict/cooperation level G_i is:

$$g_i = \sum^J G_j \mathrm{P}(M=j \mid T=i, M \neq 0) \qquad (4)$$

where

$$\mathrm{P}(M=j \mid T=i, M \neq 0) = \frac{\mathrm{P}(M \mid T)\mathbf{1}(M \neq 0)}{\mathrm{P}(M \neq 0 \mid T)}.$$

and $\mathbf{1}(M \neq 0)$ is an indicator function equaling 1 if $M \neq 0$ and 0 otherwise.

A plot of G_i against g_i for each event category is shown in Figure 1. An unbiased estimator would show expected values on the main diagonal. Estimator bias for event category i is simply $g_i - G_i$. Estimator variance is simply the spread around the diagonal.

4.3 Comparison

We also compared the system's performance to 3 undergraduate coders (U1-3) working on the same data set. To examine undergraduate performance requires first $\mathrm{P}(U,T)$, from which we can get $P(U \mid T)$. However, we cannot simply count the proportion of times each undergraduate assigns a lead to category i when it is in fact in category i because this ignores the fact that we have sampled the leads themselves using the system, and must therefore condition on M. On the other hand we do have access to the relevant conditional distribution $P(U,T \mid M = i)$. This is the distribution of undergraduate and true categories, conditioned on the fact the the system assigns an event to category i. The desired $P(U,T)$ is a weighted average of these distributions:

$$P(U,T) = \sum_i P(U,T \mid M = i)P(M = i).$$

$P(U \mid T)$ is then obtained by marginalization[6]. Clearly these calculations can also be used to compare other systems with the same ontology using the same materials.

Summary statistics similar to those described above can be easily computed (King and Lowe, 2002). Here we provide graphical results: Figure 2 plots the bias of the system and that of the undergraduates over the category set (with smoothed estimates superimposed). In the figure, the bias $G_i - g_i$ is plotted against G_i, so deflections from the horizontal are systematic bias. In almost all cases we find that more conflictual (negative valued) categories are mistaken for more cooperative ones, with some suggestion of a similar effect at the cooperative end too. Of most interest is the basic similarity in performance between undergraduates and the information extraction system.

It would be helpful if the bias that appears in these plots were systematically related to the expected system response. If this was the case, in future use we could simply adjust the system's response up or down by some coefficient determined in the evaluation process and remove the bias. However, figure 3 shows that there is no systematic relation between the expected reponses and the level of bias, so no such coefficient canbe computed. This is a rather pessimistic result for this system, suggesting a level of bias that cannot be straightforwardly removed. On the other

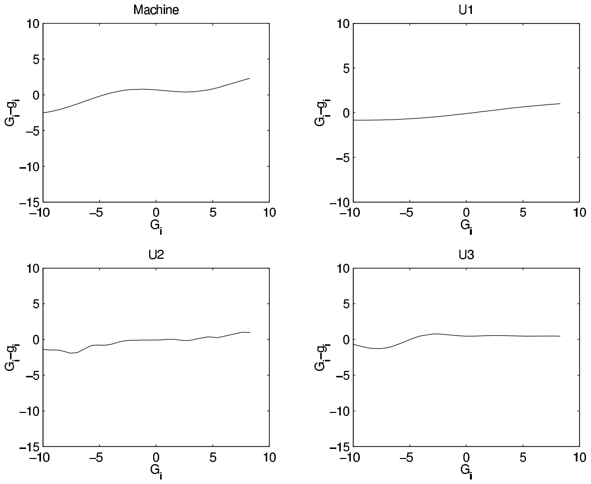


Figure 2: System (M) versus undergraduate coder (U1-3) bias. Connected lines are generated by smoothing $G_i - g_i$.

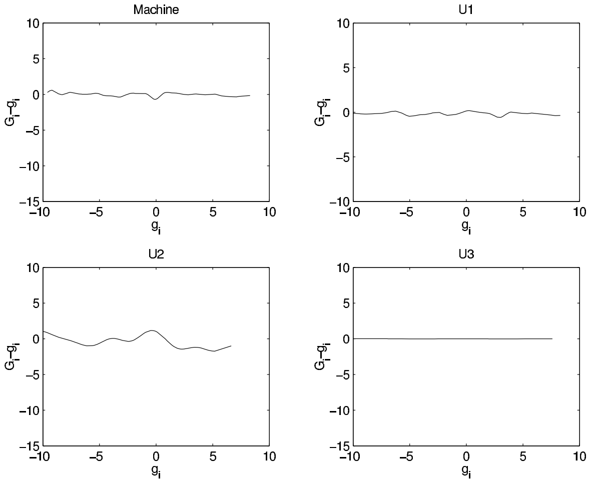


Figure 3: Bias plotted against expected system and undergraduate response. Deviations from the horizontal suggest the possibility of a post-output correction to correct for bias in subsequent application.

[6] We would normally expect to use $P(U \mid T, U \neq 0)$, but the undergraduates never failed to assign categories.

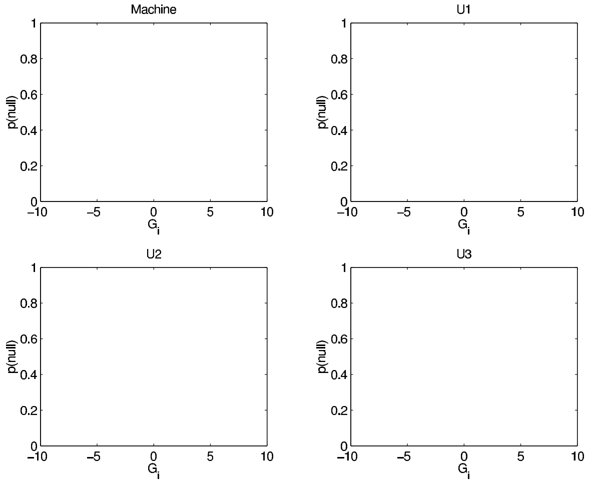


Figure 4: The probability that the system, or undergraduate fails to assign an event to a category, plotted against the level of conflict/cooperation of that category.

hand, one of the advantages of the methods presented here is that this bias is now *estimated*, and, since bias estimates are available on a category-by-category basis, redesigning effort can be directed in a way that maximizes generalization performance.

Finally, figure 4 plots the probability that the machine failed to assign an event category, $\mathrm{P}(M = 0 \mid T = i)$ (denoted p(null) in the figure), as a function of that category's conflict/cooperation value, G_i. Our interest in G_i reflects the use this data is typically put to, since we are most concerned with errors that make the world look systematically more (or less) cooperative than it really is. But we might equally have plotted $\mathrm{P}(M = 0 \mid T = i)$ against i itself, or any other property of events that might be suspected to generate difficult to categorize event descriptions.

Like the previous figures, plotting $\mathrm{P}(M = 0 \mid T = i)$ against other quantities is a useful diagnostic, indicating where future work should best be applied. In this case there appears to be no systematic relationship between the true level of conflict/cooperation and the probability that either the system or the undergraduates will fail to assign the event to a category.

5 Conclusion

We have presented a set of statistical methods for evaluating an information extraction system without unreasonable manual labour when the distribution of categories to be extracted is heavily skewed. The scheme uses a form of biased sampling and subsequent correction to estimate a probability distribution of system responses for each true category in the data. This distribution costitutes a likelihood function for the system. We then show how functions of this distribution can be used for evaluation, and estimate the system's statistical bias.

The two main ideas: using estimates of $\mathrm{P}(M \mid T)$ as the basis for evaluation, and using a non-standard sampling scheme for the estimation, are separate. Emphasis on using $\mathrm{P}(M \mid T)$ comes from standard statistical theory, and if correct, suggests how evaluation in information extraction might be integrated in to that body of theory. When a sample of leads is randomly chosen and can be expected to be reasonably representative, then the sampling machinery described above, the computation of $\mathrm{P}(M)$, and the application of Bayes theorem will not be necessary. But when the distribution of categories to be extracted is so highly skewed then our method is the only one that will make it feasible to evaluate a system on *all* of its categories in an unbiased way.

The principle difference between these and standard evaluation methods is in our explicitly statistical framework, and our consideration of how to sample in a representative way, and methods to get around cases where we cannot. The exact relationship to precision, recall etc. is the topic of current research. In the meantime we hope that the methods presented might advance understanding of effective evaluation methods in computational linguistics.

Acknowledgments

We thank Doug Bond, Craig Jenkins, Dylan Balch-Lindsay, Phil Schrodt, and two anonymous reviewers for helpful comments, and the National Science Foundation (IIS-9874747), the National Institutes of Aging (P01 AG17625-01), the Weatherhead Center for International Relations, and the

World Health Organization for research support.

References

Argamon-Engelson, S. and Dagan, I. (1999). Committee-based sample selection for probabilistic classifiers. *Journal of Artificial Intelligence Research*, 11:335–360.

Azar, E. E. (1982). *Codebook of the Conflict and Peace Databank.* Center for International Development, University of Maryland.

Cowie, J. and Lehnert, W. (1996). Information extraction. *Communications of the ACM*, 39(1):80–91.

Goldstein, J. S. (1992). A conflict-cooperation scale for WEIS events data. *Journal of Conflict Resolution*, 36(2).

Goldstein, J. S. and Freeman, J. R. (1990). *Three-Way Street: Strategic Reciprocity in World Politics.* Chicago University Press.

Goldstein, J. S. and Pevehouse, J. C. (1997). Reciprocity, bullying and international conflict: Time-series analysis of the Bosnia conflict. *American Political Science Review*, 91(3):515–529.

Grishman, R. (1997). Information extraction: Techniques and challenges. In Pazienza, M. T., editor, *Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology*, volume 1299 of *Lecture Notes in Artificial Intelligence*, chapter 2, pages 10–27. Springer Verlag.

King, G. and Lowe, W. (2002). An automated information extraction tool for international conflict data with performance as good as human coders: A rare events evaluation design. http://gking.harvard.edu/infoex.pdf.

Lehnert, W. and Sundheim, B. (1991). A performance evaluation of text-analysis technologies. *AI Magazine*, pages 81–95.

McClelland, C. (1978). *World Event / Interaction Survey (WEIS) 1966-1978.* Inter-University Consortium for Political and Social Research, University of Southern California.

Schrodt, P. A., Davis, S. G., and Weddle, J. L. (1994). Political science: KEDS — a program for the machine coding of event data. *Social Science Computer Review*, 12.

Schrodt, P. A. and Gerner, D. J. (1994). Validity assessment of a machine-coded event data set for the Middle East, 1982-92. *American Journal of Political Science*, 38(3).

Sundheim, B. (1992). Overview of the fourth message understanding evaluation and conference. In *Proceedings of the Fourth Message Understanding Conference*, pages 3–22.

Sundheim, S., editor (1991). *Proceedings of the Third Message Understanding Conference*, San Mateo, CA. Morgan Kaufmann.

A Quantitative Method for Machine Translation Evaluation

Jesús Tomás
Escola Politècnica Superior de Gandia
Universitat Politècnica de València
jtomas@upv.es

Josep Àngel Mas
Departament d'Idiomes
Universitat Politècnica de València
jamas@idm.upv.es

Francisco Casacuberta
Institut Tecnològic d'Informàtica
Universitat Politècnica de València
fcn@iti.upv.es

Abstract

Accurate evaluation of machine translation (MT) is an open problem. A brief survey of the current approach to tackle this problem is presented and a new proposal is introduced. This proposal attempts to measure the percentage of words, which should be modified at the output of an automatic translator in order to obtain a correct translation. To show the feasibility of the method we have assessed the most important Spanish-Catalan translators in comparing the results obtained by the various methods.

1 Introduction

Research in automatic translation lacks an appropriate, consistent and easy to use criterion for evaluating the results (White et al., 1994; Niessen et al., 2000). However, it turns out to be indispensable to have some tool that may allow us to compare two translation systems or to elicit how any variation of our system may affect the quality of the translations. This is important in the field of research as well as when a user has to choose between two or more translators.

The evaluation of a translation system shows a number of inherent difficulties. First of all we are dealing with a subjective process, which is even difficult to define.

This paper is circumscribed to the project SISHITRA (*SIStemas HÍbridos para la TRAducción valenciano-castellano* supported by the Spanish Government), whose aim is the construction of an automatic translator between Spanish and Catalan texts using hybrid methods (both deductive and inductive).

In the following section we discuss some of the most important translation quality metrics. After that, we introduce a semiautomatic methodology for MT evaluation and we show a tool to facilitate this kind of evaluation. Finally, we present the results obtained on the evaluation of several Spanish-Catalan translators.

2 Metrics in MT Evaluation

2.1 Automatic Evaluation Criteria

Within the scope of inductive translation, the use of objective metrics, which can be evaluated automatically, is quite frequent. These metrics take as their starting point a possible reference translation for each of the sentences we want to translate. This reference will be compared with the proposed sentences by the translation system. The most important metric systems are:

Word Error Rate (WER):
WER is the percentage of words, which are to be inserted, deleted or replaced in the translation in order to obtain the sentence of reference (Vidal, 1997; Tillmann et al., 1997). WER can be obtained automatically by using the editing distance between both sentences. This metric is computed efficiently and is reproducible (successive applications to the same data produce the same results). However, the main drawback is its dependency on the sentences of reference. There is an almost unlimited number of correct translations for one and the same sentence and, however, this metric considers only one to be correct.

Sentence Error Rate (SER):
SER indicates the percentage of sentences, whose translations have not matched in an exact manner those of reference. It shows similar advantages and shortcomings as WER.

Some variations on WER have been defined, which can also be obtained automatically:

Multi reference WER (mWER):
Identical approach to WER, but it considers several references for each sentence to be translated, i.e., for each sentence the editing distance will be calculated with regard to the various references and the smallest one is chosen (Niessen et al., 2000). It presents the drawback of requiring a great human effort before actually being able to use it. However, the effort is worthwhile, if it can be later used for hundreds of evaluations.

BLEU Score:
BLEU is an automatic metric designed by IBM, which uses several references (Papineni et al., 2002). The main problem of mWER is that all possible reference translations cannot be introduced. The BLEU score try to solve this problem by combining the available references. In a simplified manner we could say that it measures how many word sequences in the sentence under evaluation match the word sequences of some reference sentence. The BLEU score also includes a penalty for translations whose length differs significantly from that of the reference translation.

2.2 Subjective Evaluation Criteria

Other kinds of metrics have been developed, which require human intervention in order to obtain an evaluation. Among the most widely used we could stand out:

Subjective Sentence Error Rate (SSER)
Each sentence is scored from 0 to 10, according to its translation quality (Niessen et al., 2000).
An example of these categories is:

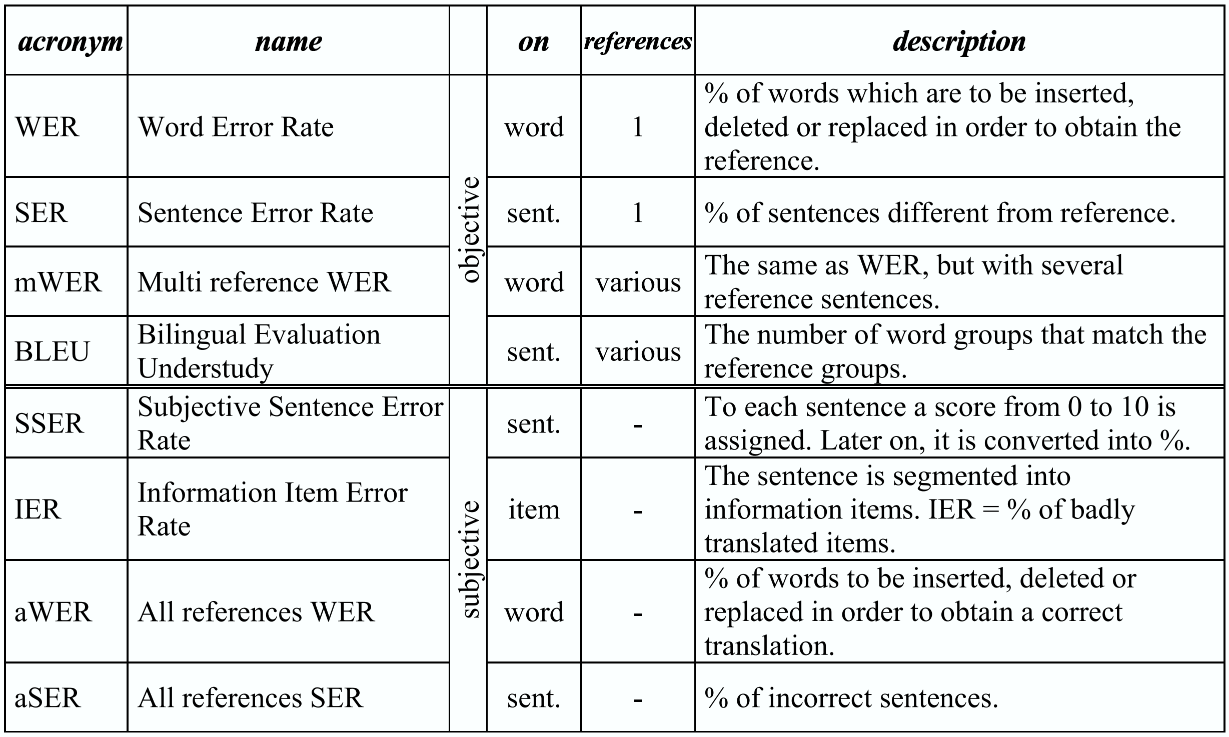
0 – nonsensical...
1 – some aspects of the content are conveyed
...
5 – comprehensible, but with important syntactic errors
...
9 – OK. Only slight style errors.
10 – perfect.

The biggest problem shown by this technique is its subjective nature. Two people who may evaluate the same experiment could obtain quite different results. To solve this problem several evaluations can be performed. Another drawback is that the different sentence lengths have not been taken into account. The score of a 100 word-long sentence has the same impact on the total score as that of a word-long sentence.

Information Item Error Rate (IER)
An unclear question is how to evaluate long sentences consisting of correct and wrong parts. IER attempts to find a solution to this question. In order to solve the problem the concept of “information items” is introduced. The sentences are divided into word segments. Each item of the sentence is marked with “OK”, “error”, “syntax”, “meaning” or “others”, as shown in the translation. The metric IER (Information Item Error Rate) can then be calculated as the percentage of badly translated items (not marked as “OK”) (Niessen et al., 2000).

2.3 New Evaluation Criteria

Automatic metrics are especially useful, since their cost is practically null. However, they are very dependent on the used references. In some cases they can yield misleading results, for instance, if we want to compare an inductive translation system with some deductive one which, in principle, should produce translations of a similar quality. If we extract the references from the same source as the training material of the inductive translator, the inductive translator will have an advantage over the deductive translator, since it has learned to translate by using a vocabulary and structures that are similar to those appearing in the references.



acronym	*name*		*on*	*references*	*description*
WER	Word Error Rate	objective	word	1	% of words which are to be inserted, deleted or replaced in order to obtain the reference.
SER	Sentence Error Rate		sent.	1	% of sentences different from reference.
mWER	Multi reference WER		word	various	The same as WER, but with several reference sentences.
BLEU	Bilingual Evaluation Understudy		sent.	various	The number of word groups that match the reference groups.
SSER	Subjective Sentence Error Rate	subjective	sent.	-	To each sentence a score from 0 to 10 is assigned. Later on, it is converted into %.
IER	Information Item Error Rate		item	-	The sentence is segmented into information items. IER = % of badly translated items.
aWER	All references WER		word	-	% of words to be inserted, deleted or replaced in order to obtain a correct translation.
aSER	All references SER		sent.	-	% of incorrect sentences.

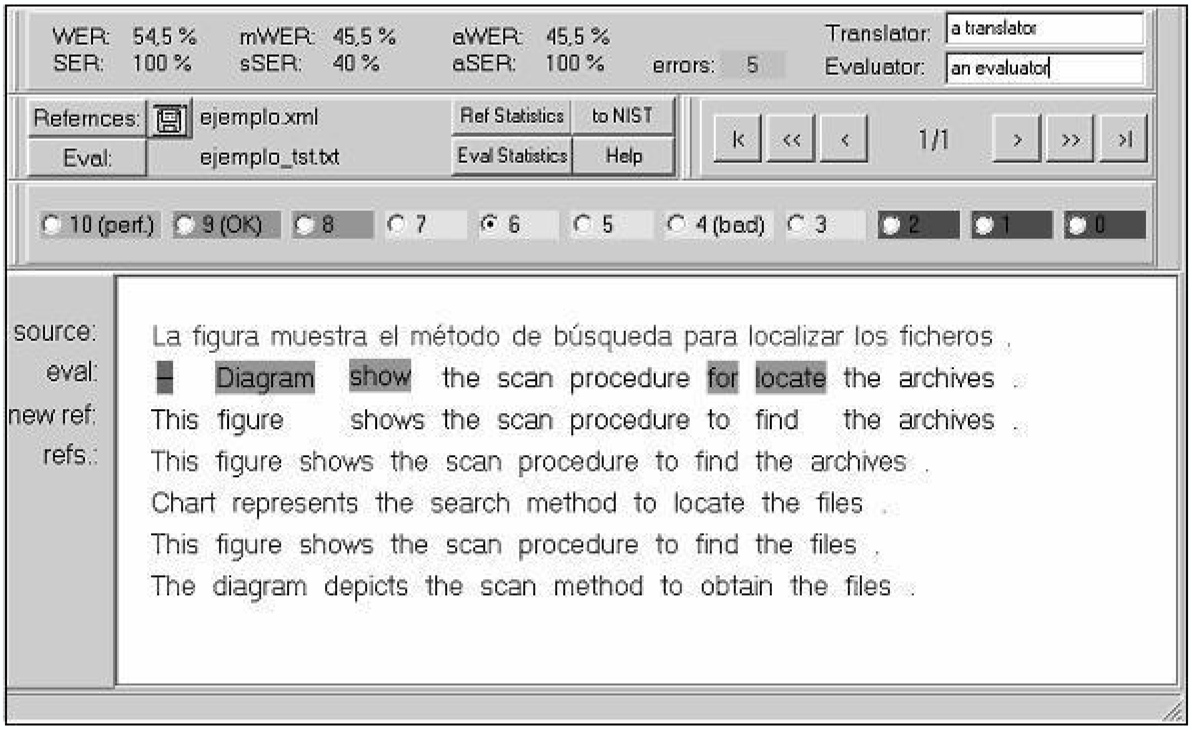
Table 1. Some metrics in MT evaluation

The non-automatic evaluation metrics described above presents various constraints: When an SSER is used, it may be very difficult to decide the score to be assigned to one sentence. For example, if in one sentence a small syntactic error appears, we can assign an 8. If in the following sentence two similar errors appear, what score should we assign? The same or half the score? To solve these kinds of matters, IER introduces the concept of "information item". This proposal has the drawback of being quite costly, both during the initial stage of deciding the word segments which form each item as well as when classifying the correction for each item. After having seen the previous drawbacks the following metric has been introduced:

All references WER (aWER):
It measures the number of words, which are to be inserted, deleted or replaced in the sentence under evaluation in order to obtain a correct translation. It can also be seen as a particular case of the mWER, but taking for granted that all the possible references are at our disposal. Since it is impossible to have *a priori* all possible references, the evaluator will be able to propose new references, if needed. The evaluation process can be carried out very quickly, if one takes as the starting point the result obtained by the WER or the mWER. The idea consists of visualising the incorrect words detected by one of these methods (editing operations). The evaluator just needs to indicate whether each of the marked items is an actual error or whether it can rather be considered as an alternative translation

This metric resembles very much the one proposed in (Brown et al, 1990). That work suggested for measuring the translation quality counting the number of times an evaluator would have to press the keyboard keys in order to make the proposed sentence correct.

All references Sentence Error Rate (aSER):
The SER metric presents the drawback of working with only one reference. Therefore, it does not really measure the number of wrong sentences, but rather those that do not match exactly the reference. For this reason we thought it would be interesting to introduce a metric that could indicate the percentage of sentences whose



WER: 54,5 % mWER: 45,5 % aWER: 45,5 %
SER: 100 % sSER: 40 % aSER: 100 % errors: 5

Translator: a translator
Evaluator: an evaluator

Refemces: ejemplo.xml
Eval: ejemplo_tst.txt

○ 10 (perf.) ○ 9 (OK) ○ 8 ○ 7 ● 6 ○ 5 ○ 4 (bad) ○ 3 ○ 2 ○ 1 ○ 0

source: La figura muestra el método de búsqueda para localizar los ficheros .
eval: - Diagram show the scan procedure for locate the archives .
new ref: This figure shows the scan procedure to find the archives .
refs.: This figure shows the scan procedure to find the archives .
Chart represents the search method to locate the files .
This figure shows the scan procedure to find the files .
The diagram depicts the scan method to obtain the files .

Figure 1. The Graphic User Interface. The system highlights the non-matching words between the evaluation sentence and the nearest reference.

translations are incorrect. This metric can be obtained as a by-product of the aWER.

3 Evaluation Tool for MT

In order to facilitate the evaluation of automatic translators a graphic user interface has been implemented. The metrics provided by the program are: WER, mWER, aWER, SER, SSER and aSER. Figure 1 shows how it is displayed.

Next, the way the program works is described:

On the editing window from top to bottom the following items are displayed: the source sentence, the sentence to be evaluated, the new sentences proposed by the user, the four most similar references to the sentence under evaluation (according to editing distance). The new sentence proposed by the user will be in principle the same as that of the most similar reference. In the sentence being evaluated using different colours, depending on whether they are considered insertions, replacements or deletions, the words that may be wrong are highlighted.

The user can click with the mouse on those words that may be considered correct. As a result, this action will modify the new reference. In the example (figure 1), if the user clicks on the highlighted words *"-"*, *"Diagram"* and *"locate"*, he will obtain the new reference *"Diagram shows the scan procedure to locate the archives."*. This new reference reduces the editing distance from 5 to 2. The user will also be able to click directly on some word of new reference to modify it. The aim of this is to allow the evaluator the introduction of any new reference which may be a correct translation of the source sentence and which, furthermore, may resemble most closely the sentence being evaluated.

This tool can be obtained for free on (http://ttt.gan.upv.es/~jtomas/eval), both in the Linux version as wells as in Windows.

3.1 Evaluation Database Format

A format in XML has been defined to store the reference files. For each evaluation sentence we store: the source sentence, the target reference sentences and the target sentences proposed by the different MT with their subjective evaluations. Should during an aWER evaluation a new reference be proposed, this one is also stored. An example of a file with a sentence under evaluation is shown as follows:

```
<evalTrans>
<sentence>
  <source>
    La figura muestra el método.
  </source>
  <eval translator="first reference">
    <target>
      This figure shows the procedure.
    </target>
  </eval>
  <eval translator="multi reference">
    <target>
      This figure shows the method.
    </target>
  </eval>
  <eval translator="Statistical"
   evaluator="JM" sser="8" awer="1/5">
    <target>
      Chart represent the method.
    </target>
    <newRef>
      Chart represents the method.
    </newRef>
  </eval>
</sentence>
...
</evalTrans>
```

4 Example of Evaluation

4.1 Spanish-Catalan Translators

The tool described in the previous section has been applied to the most important Spanish-Catalan translators.

The Catalan language receives more or less intense institutional support in all territories of the Spanish state, where it is co-official with Spanish (Balearic Islands, Catalonia and Valencian Community). This makes it compulsory from an administrative standpoint to publish a bilingual edition of all official documents. For that purpose the use of a Machine Translator becomes almost indispensable.

But the official scope is not the only one where we can find the need to write bilingual documents in a short period of time. The most obvious example can be the bilingual edition of some newspapers, such as *El País* or *El Periódico de Catalunya*, both in their editions for the autonomous community of Catalonia.

In the following section there is a brief description of each of the programs we have reviewed:

Salt: an automatic translation program of the Valencian local government, which also includes a text corrector. It can be downloaded for free from http://www.cultgva.es. It has an interactive option for solving doubts (subjective ambiguity resolution) and is executed with the OS Microsoft Windows.

Incyta: the translation business web-site Incyta (http://www.incyta.com) was adding at the time of this evaluation example review a free on-line automatic translator for short texts.

Internostrum: an on-line automatic translation program, available at http://www.torsimany.ua.es, designed by the Language and Computational Systems Department of the University of Alicante. It marks the doubtful words or segments as a review helping aid. It uses finite-state technology (Canals et al., 2001).

Statistical: An experimental translator developed at the Computer Technology Institute of the Polytechnic University of Valencia. All components have been inferred automatically from training pairs using statistical methods (Tomás & Casacuberta, 2001). It is accessible at http://ttt.gan.upv.es/~jtomas/trad.

4.2 Setting up the evaluation experiment

In order to carry out our evaluation, we have translated 120 sentences (2456 words) with the different MT. These sentences have been taken from different media: a newspaper, a technical manual, legal text... The references used by the WER were also taken from the Catalan version of the same documents. In mWER and in BLEU we used three additional references. These new references have been introduced by a human translator modifying the initial reference.

Before applying the metrics shown in point 2, a human expert carries out a detailed analysis in order to establish the quality of the translations. The experiment consists of sorting out the four outputs obtained by each translator for each test sentence, according to its quality. If the expert does no find any quality difference between the

Translator	first	second	thrid	fourth
Salt	69%	13%	13%	4%
Incyta	63%	11%	13%	13%
Statistical	60%	13%	7%	20%
Internostrum	48%	12%	20%	20%

Table 2. Comparative classification sentence by sentence.

sentences proposed by two translators, he assigns the same rank to them. Table 2 shows the results obtained. After this sentence by sentence analysis, the expert concludes that *Salt* is the better translator, followed closely by *Incyta*. *Statistical* is in an intermediate position and the worst is *Internostrum*.

4.3 Results

The results of our experiment can be observed in Figure 2. Table 3 shows the evaluation time for the 120 sentences. The first thing we can point out is that the *Salt* translator obtains the best results from all used metrics and *Internostrum* is the worst of all metrics. The other two translators obtain different results depending on the used method. Next we will discuss the results obtained by the different methods:

The **WER** metric shows a strong dependence on the used reference. If the translator employs a similar style or vocabulary with regard to those of the reference, it clearly achieves better results. This fact determines that the obtained results do not show faithfully the quality of the translations. Specifically, for *Incyta* it obtains bad results, although that does not coincide with the conclusions of the expert.

The main advantage of this method is that it is a totally automatic measurement without any evaluation cost. These conclusions can also be extended to the SER.

mWER solves in part the problem posed by the WER. To attempt to introduce *a priori* all possible translations turns out to be impossible, so that it has to choose a subset of these giving thus the method a certain subjective nature. In the case of our evaluation, the references were introduced by using certain dialectal variants. That worked slightly against some automatic translator, which preferred some other dialectal variants.

The **BLEU** metric tries to combine the available references in order to improve the mWER metric. In our experiment the use of several references, in mWER and BLEU, does not solve the deficiency of WER. It continues being most detrimental to *Incyta*.

The use of the mWER and BLEU required a great initial effort, when the references were written, by even choosing only three new references for each translation. However, these methods had a big advantage: each evaluation is done without any additional cost.

When we applied the **SSER**, we faced the following dilemma: Which criteria should we use for applying the scoring scale? We decided that the latter had to be related with the global understanding of the sentence and the number of errors in correspondence with the sentence length. Since this criterion is not made explicit in the method the choice of a different criterion would have produced very diverse results.

Regarding the evaluation effort, it was the most costly method. In order to evaluate each sentence it was necessary to read and understand both the source sentence and the target sentence to try to score at the end the translation.

The **aWER** metric breaks with the dependence on the used references, which displayed the WER, mWER and BLEU. Moreover, it turned out to be much more objective and clearer to apply than the SSER. The metric achieved by this method provides us with clear and intuitive information. If we use the *Salt* translator we will have to correct 3% of the words in order to obtain a correct translation. Interpret the metrics supplied by the other methods it becomes unavoidable to know the conditions under which the evaluation has been carried out (references used, criteria ...).

The evaluation effort for the aWER is significantly less than the mWER and the SSER.

Translator	WER	mWER	aWER	SER	SSER	BLEU	aSER
▩ Salt	9.9 *1*	6.6 *1*	3.0 *1*	68.3 *1*	10.3 *1*	0.866 *1*	40.0 *1*
▨ Incyta	10.9 *2*	7.6 *2*	3.1 *1*	74.2 *2*	11.2 *1*	0.855 *2*	41.7 *1*
▤ Statistical	10.7 *2*	7.8 *2*	3.8 *2*	70.8 *2*	12.8 *2*	0.857 *2*	45.8 *2*
□ Internostrum	11.9 *3*	8.5 *3*	4.9 *3*	80.0 *3*	15.8 *3*	0.837 *3*	58.3 *3*

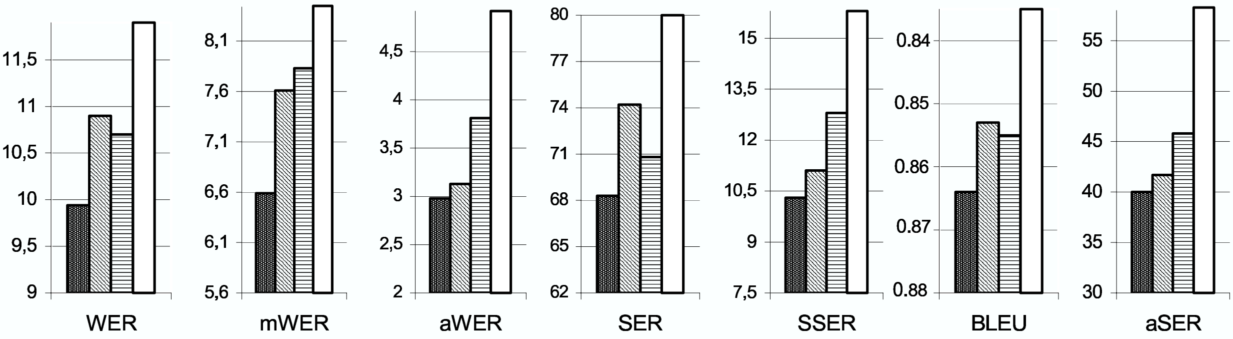


Figure 2. Comparative evaluation results using 7 different metrics for the 4 Spanish-Catalan translators. In order to interpret quickly the results obtained in each metric, we have classified each translator using the following ranking: 1- better 2- intermediate 3- worse.

	mWER / BLEU	SSER	aWER / aSER
Set-up time*	210	0	0
Internostrum	0	70	40
Salt	0	60	25
Incyta	0	55	30
Statistical	0	60	25
Total:	210	245	120

Table 3. Comparative evaluation time (minutes) of the 120 sentences using the different metrics. *Time spent to introduce the proposed references.

The discussion on the aWER method can be extended to the aSER.

Considering the expert evaluation, the subjective metrics reflect better the quality of the evaluated translations than the automatic ones. The *Incyta* translator works quite appropriately, but it proposes translations that deviate from the references. Thus, the automatic measures (WER, mWER and BLEU), based on these references, do not evaluate correctly this translator. On the other hand, the *Statistical* Translator works worse, even though its translations are more similar to the references. It is an example-based translator, and the training and test sentences have been obtained from the same sources. This can benefit the evaluation of the *Statistical* translator using automatic measures.

5 Conclusions

In this paper we present a criterion (aWER) for the evaluation of translation systems. The evaluation of the translations can be carried out quickly thanks to the use of a computer tool developed for this purpose.

We have compared this criterion with other criteria (WER, mWER, SER, BLEU and SSER) using the translations obtained by several Spanish-Catalan translators. It is our understanding that automatic measures (WER, mWER and BLEU) do not evaluate correctly the translators (specifically, they affect *Incyta* negatively).

The scores produced by human experts (SSER and aWER) are the metrics that best capture the translation quality among the different systems. As its most important aWER feature we would stand out that, in spite of being a subjective method which requires the intervention of a human evaluator, the latter will not have to take too subjective decisions.

We believe that the aWER tool could be used in another domain, for the evaluation of other natural language processing systems, e.g. summarizing systems.

In a future our aim is to add to this comparative study other score methods, in addition to comparing the variability introduced by different human evaluators in each of the methods.

Acknowledgement

This work was partially funded by the Spanish CICYT under grant TIC2000-1599-C02 and the IST Programme of the European Union under grant IST-2001-32091. The authors wish to thank the anonymous reviewers for their criticism and suggestions.

References

Brown, P. F., J. Cocke, S. Della Pietra, V. Della Pietra, F. Jelinek, R. Mercer, & P. Roossin. 1990. A Statistical Approach to Machine Translation. *Computational Linguistics* 16(2).

Canals-Marote, R., A. Esteve-Guillén, A. Garrido-Alenda, M.I. Guardiola-Savall, A. Iturraspe-Bellver, S. Montserrat-Buendia, S. Ortiz-Rojas, H. Pastor-Pina, P.M. Pérez-Antón, M.L. Forcada. 2001. The Spanish-Catalan machine translation system interNOSTRUM. In *Proceedings of the Machine Translation Summit VIII.* Santiago de Compostela, Spain.

Niessen, S., F.J. Och, G. Leusch, and H. Ney. 2000. An Evaluation Tool for Machine Translation: Fast Evaluation for MT Research. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation*, Athens, Greece.

Papineni, K.A., S. Roukos, T. Ward, W.J. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the* 40th *Annual Meeting of the Association for Computational Linguistics* (ACL), Philadelphia.

Tomás, J., F. Casacuberta. 2001. Monotone Statistical Translation using Word Groups. In *Proceedings of the Machine Translation Summit VIII.* Santiago de Compostela, Spain.

Tillmann, C., S. Vogel, H. Ney, H. Sawaf, and A. Zubiaga. 1997. Accelerated DP based Search for Statistical Translation. In *Proceedings of the 5th European Conference on Speech Communication and Technology,* Rhodes, Greece.

Vidal, E. 1997. Finite-State Speech-to-Speech Translation. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Munich, Germany.

White, J., T. O'Connell, F. O'Mara. 1994. The DARPA Machine Translation Evaluation Methodologies: Evolution, Lessons and Future Approaches. In *Proceedings of the first Conference of the Association for Machine Translation in the Americas.* Columbia, USA.

Colouring Summaries BLEU

Katerina Pastra
Department of Computer Science
University of Sheffield
katerina@dcs.shef.ac.uk

Horacio Saggion
Department of Computing Science
University of Sheffield
saggion@dcs.shef.ac.uk

Abstract

In this paper we attempt to apply the IBM algorithm, BLEU, to the output of four different summarizers in order to perform an intrinsic evaluation of their output. The objective of this experiment is to explore whether a metric, originally developed for the evaluation of machine translation output, could be used for assessing another type of output reliably. Changing the type of text to be evaluated by BLEU into automatically generated extracts and setting the conditions and parameters of the evaluation experiment according to the idiosyncrasies of the task, we put the feasibility of porting BLEU in different Natural Language Processing research areas under test. Furthermore, some important conclusions relevant to the resources needed for evaluating summaries have come up as a side-effect of running the whole experiment.

1 Introduction

Machine Translation and Automatic Summarization are two very different Natural Language Processing (NLP) tasks with -among others- different implementation needs and goals. They both aim at generating text; however, the properties and characteristics of these target texts vary considerably. Simply put, in Machine Translation, the generated document should be an accurate and fluent translation of the original document, in the target language. In Summarization, the generated text should be an informative, reduced version of the original document (single-document summary), or sets of documents (multi-document summary) in the form of an abstract, or an extract. Abstracts present an overview of the main points expressed in the original document, while extracts consist of a number of informative sentences taken directly from the source document. The fact that, by their very nature, automatically generated extracts carry the single sentence qualities of the source documents[1], may lead one to the conclusion that evaluating this type of text is trivial, as compared to the evaluation of abstracts or even machine translation, since in the latter, one needs to be able to evaluate the content of the generated translation in terms of grammaticality, semantic equivalence to the source document and other quality characteristics (Hovy et al., 2002).

Though the evaluation of generated extracts is not as demanding as the evaluation of Machine Translation, it does have two critical idiosyncratic aspects that render the evaluation task difficult:

- the compression level (word or sentence level) and the compression rate of the source document must be determined for the selection of the contents of the extract ; the values of these variables may greatly affect the whole evaluation setup and the results obtained

[1] Even if coherence issues may arise beyond the sentence boundaries i.e. at the text level

- the very low agreement among human evaluators on what is considered to be "important information" for inclusion in the extract, reaching sometimes the point of total disagreement on the focus of the extract (Mani, 2001; Mani et al., 2001). The nature of this disagreement on the adequacy of the extracts is such that - by definition - cannot manifest itself in Machine Translation; this is because it refers to the adequacy of the contents chosen to form the extract, rather than what constitutes an adequate way of expressing all the contents of the source document in a target language.

The difference on the parameters to be taken into consideration when performing evaluation within these two NLP tasks presents a challenge for porting evaluation metrics from the one research area to the other. Given the relatively recent success in achieving high correlations with human judgement for Machine Translation evaluation, using the IBM content-based evaluation metric, BLEU (Papineni et al., 2001), we attempt to run this same metric on system generated extracts; this way we explore whether BLEU can be used reliably in this research area and if so, which testing parameters need to be taken into consideration. First, we refer briefly to BLEU and its use across different NLP areas, then we locate our experiments relatively to this related work and we describe the resources we used, the tools we developed and the parameters we set for running the experiments. The description of these experiments and the interpretation of the results follows. The paper concludes with some preliminary observations we make as a result of this restricted, first experimentation.

2 Using BLEU in NLP

Being an intrinsic evaluation measure (Sparck Jones and Galliers, 1995), BLEU compares the content of a machine translation against an "ideal" translation. It is based on a "weighted average of similar length phrase matches" (n-grams), it is sensitive to longer n-grams (the baseline being the use of up to 4-grams) and it also includes a brevity penalty factor for penalising shorter than the "gold standard" translations (Papineni et al., 2001; Doddington, 2002). The metric has been found to highly correlate with human judgement, being at the same time reliable even when run on different documents and against different number of model references. Experiments run by NIST (Doddington, 2002), checking the metric for consistency and sensitivity, verified these findings and showed that the metric distinguishes, indeed, between quite similar systems. A slightly different version of BLEU has been suggested by the same people, which still needs to be put into comparative testing with BLEU before any claims for its performance are made.

BLEU has been used for evaluating different types of NLP output to a small extent. In (Zajic et al., 2002), the algorithm has been used in a specific Natural Language Generation application: headline generation. The purpose of this work was to use an automated metric for evaluating a system generated headline against a human generated one, in order to draw conclusions on the parameters that affect the performance of a system and improve scoring similarity. In (Lin and Hovy, 2002) BLEU has been applied on summarization. The authors argue on the unstable and unreliable nature of manual evaluation and the low agreement among humans on the contents of a reference summary. Lin and Hovy make the case that automated metrics are necessary and test their own modified recall metric, along with BLEU itself, on single and multi-document summaries and compare the results with human judgement. Modified recall seems to reach very high correlation scores, though direct comparative experimentation is needed for drawing conclusions on its performance in relation to BLEU. The latter, has been shown to achieve 0.66 correlation in single-document summaries at 100 words compression rate and against a single reference summary. The correlation achieved by BLEU climbs up to 0.82 when BLEU is run over and compared against multiply judged document units, that could be thought of as a sort of multiple reference summaries. The correlation scores for multi-document summaries are similar. Therefore, BLEU has been found to correlate quite highly with human judgement for the summarization task when multiple judgement is involved, while -as Lin and Hovy indicate- using a single reference is not adequate for getting reliable results with high correlation with the human evaluators.

It is this conclusion that Lin and Hovy have drawn, that contradicts findings by the IBM and NIST people for the importance of using multiple references when using BLEU in Machine Translation. The use of either multiple references or just a single reference has been proved not to affect the reliability of the results provided by BLEU (Papineni et al., 2001; Doddington, 2002), which seems not to be the case in summarization. This is not a surprise; comparisons of content-based metrics for summarization in (Donaway et al., 2000) have led the authors to the conclusion that such metrics correlate highly with human judgement when the humans do not disagree substantially. The fact that more than one reference summaries are needed because of the low agreement between human evaluators has been repeatedly indicated in automatic summarization evaluation (Mani, 2001).

We attempt to test BLEU's reliability when changing various evaluation parameters such as the source documents, the reference summaries used and even parameters unique to the evaluation of summaries, such as the compression rate of the extract. In doing so, we explore whether the metric is indeed reliable only when using more than a single reference and whether any other testing parameter could compensate for lack of multiple references, if used appropriately.

3 Evaluation Experiment

In this section, we will present a description of the experiments themselves, along with the results obtained and their analysis, preceded by information on the corpus we used for our experiments and the tools we developed for setting their parameters and running them automatically.

3.1 Testing corpus

We make use of part of the language resources (HKNews Corpus) developed during the 2001 Workshop on Automatic Summarization of Multiple (Multilingual) Documents (Saggion et al., 2002).

The documents of each cluster are all relevant to a specific topic-query, so that they form, in fact, thematic clusters. The texts are marked up on the paragraph, sentence and word level. Annotations with linguistic information (Part of speech tags and morphological information), though marked up on the documents have not been used in our experiments at all. Three judges have assessed the sentences in each cluster and have provided a score on a scale from 0 to 10 (i.e. utility judgement), expressing how important the sentence is for the topic of the cluster (Radev et al., 2000). In our experiments, we have used three document clusters, each consisting of ten documents in English.

3.2 Summarizers

It is important to note, that our objective is not to demonstrate how a particular summarization methodology performs, but to analyse an evaluation metric. The summaries used for the evaluation were produced as extracts at different 'sentence' (and not word) compression rates[2]. In order to produce summarizers for our evaluation, we use a robust summarisation system (Saggion, 2002) that makes use of components for semantic tagging and coreference resolution developed within the GATE architecture (Cunningham et al., 2002). The system combines GATE components with well established statistical techniques developed for the purpose of text summarisation research. The system supports "generic" and query-based summarisation addressing the need for user adaptation[3]. For each sentence, the system computes values for a number of 'shallow' summarization features: position of the sentence, term distribution analysis, similarity of the sentence with the document, similarity with the sentence at the leading part of the document, similarity of the sentence with the query, named entity distribution analysis, statistic cohesion, etc. The values of these features are linearly combined to produce the sentence fi-

[2] We have to note that the level of compression i.e sentence or word level, affects probably the evaluation of the summarizers' output. Comparative testing could indicate whether this is a crucial parameter for system evaluation.

[3] The software can be obtained from `http://www.dcs.shef.ac.uk/~saggion`

nal score. Top-ranked sentences are annotated until the target n% compression is achieved (an annotation set is produced for each summary that is generated). Different summarization systems can be deployed by setting-up the weights that participate in the scoring formula. Note that as the summarization components are not aware of the compression parameter, one would expect specific configurations to produce good extracts at different compression rates and across documents.

We have configured four different summarizers, namely, the "query-based system" that computes the similarity of each sentence of the source document with the documents topic-query, in order to decide whether to include a sentence in the generated extract or not. We also have the "Simple 1 system", whose main feature is that it computes the similarity of a sentence with the whole document, the "Simple 2 system" which is a lead based summarizer and the "Simple 3 system" that blindly extracts the last part of the source document.

3.3 Judge-based Summaries

Following the same methodology used in (Saggion et al., 2002), we implemented a judge-based summarization system that given a judge number (1, 2, 3, or all), it scores sentences based on a combination of the utility that the sentence has according to the judge (or the sum of the utilities if 'all') and the position of the sentence (leading sentences are preferred). These 'extracts' represent our gold-standards for evaluation in our experiments. In order to use the documents in a stand-alone way, we have enriched the initial corpus mark-up and added to each document information about cluster number, cluster topic (or query) and all the information about utility judgement (that information was kept in separate files in the original HKNews corpus).

3.4 Evaluation Software

We have developed a number of software components to facilitate the evaluation and we make use of the GATE development environment for testing and processing. The evaluation package allows the user to specify different reference extracts (judge-based summarizers) and summarization systems to be compared.

Co-selection comparison (i.e., precision and recall) is being done with modules obtained from the GATE library (AnnotationDiff components). Content-based comparison by the Bleu algorithm was implemented as a Java class. The exact formula provided by the developers of BLEU has been implemented following the baseline configurations i.e use of 4-grams and uniform weights summing to 1:

$$Bleu(S,R) = K(S,R) * e^{Bleu_1(S,R)}$$

$$Bleu_1(S,R) = \sum_{i=1,2,...n} w_i * \lg(\frac{|(S_i \bigcap R_i)|}{|S_i|})$$

$$K(S,R) = \begin{cases} 1 & \text{if } |S| > |R| \\ e^{(1-\frac{|R|}{|S|})} & \text{otherwise} \end{cases}$$

$$w_i = \frac{i}{\sum_{j=1,2,...n} j} \qquad \text{for } i = 1,2,...,n$$

where S and R are the system and reference sets. S_i and R_i are the "bags" of i-grams for system and reference. n is a parameter of our implementation, but for the purpose of our experiments we have set n to 4.

3.5 Experiments

In our experiments we have treated compression rates and clusters as variables each one being a condition for the other and both dependent to a third variable, the gold standard summary. We ran BLEU in all different combinations in order to see the main effects of each combination and the interactions among them. In particular, we have used three different text clusters, consisting of texts that refer to the same topic: cluster 1197 on "Museum exhibits and hours", cluster 125 which deals with "Narcotics and rehabilitation" and cluster 241 which refers to "Fire safety and building management". For the texts of each cluster we have three different reference summaries (created according to the utility judgement score assigned by human evaluators cf. 3.1 and 3.2). We will refer to these as Reference1, Reference2 and Reference3. The judges behind these references are all the same for the three text clusters with one exception: Reference1 in cluster 241 has not been created by the same human evaluator as the Reference 1 summaries for the other two clusters. Last, we ran the experiments at five different compression rates [4]: 10%, 20%, 30%, 40% and 50%.

We first ran BLEU on the reference summaries in order to check whether BLEU is consistent in the data it produces concerning the agreement among human evaluators. We tried all possible combinations for comparing the reference summaries; using at first Reference 1 as the gold standard, we ran BLEU over References 2 and 3 and we did this for two clusters (since the third's -241- Reference 1 set of summaries had been created by another judge - a fourth one). We did this for all five compression rates separately. We repeated the experiment changing the gold standard and the references to be scored accordingly (i.e Reference 1 and 3 against 2, Reference 1 and 2 against 3). The results we got were consistent neither across clusters, nor within clusters across compression rates; however the latter, did show a general tendency for consistency which allows for some observations to be made. In cluster 1197, References 1 and 2 are generally in higher agreement than with 3, a fact verified regardless the reference chosen as a gold standard. The fact that References 1 and 2 are very close was also evident when both compared against Reference 3; though the latter is generally closer to Reference 2, the scores assigned to Reference 1 and 2 are extremely close. In cluster 125, Reference 1 is consistently closer to 3, while 2 is closer to 1 at some compression rates and closer to 3 at others. These very close scores indicate that all three references are similarly "distant" one from another, and no groupings of agreement can actually be made. Agreement between reference summaries augments as the compression rate also increases, with the higher similarity scores always found at the 50% compression rate and the lower ones consistently found at 10%. Table 1 shows a consistent ranking across compression rates in cluster 1197 and an inconsistent one in cluster 125, using in both cases Reference 2 as the gold standard. From this first experiment, the rankings of the reference summaries seem to depend on the different values of the variables used. If that is the case, then one should use BLEU in summarization only when determining specific values for the evaluation experiment, that will guarantee reliable results; but how could one determine which value(s) should be chosen? To explore things further we decided to proceed with a second experiment set up in a similar way.

In our second experiment we try to compare the system generated extracts (and therefore the performance of the four summarizers) against the different human references. Again, the different rounds of the experiment involve multiple parameters; the generated extracts of all three text clusters are compared against each reference summary, against all reference summaries (integrated summary) and at all five compression rates. Going through the different stages of this experiment we observe that:

- For Reference X within Cluster Y across Compressions, the ranking of the systems is not consistent

One does not get the same system ranking at different compression rates. The similarity of a generated extract to a specific reference summary is the same at some compression rates, similar at others (e.g the order of two of the systems swaps) and totally different at other rates. No patterns arise in the way that rankings are similar at specific compression rates; for example, in table 2, there seems to be a prevailing ranking common in four compression rates; however, the ranking provided at 10% is totally different, and no apparent reason seems to justify this deviation (e.g. very close scores). Furthermore, this agreement among the four highest compression rates does not form a pattern i.e it does not appear as such across clusters or references.

- For Reference X at Compression Y across Clusters, the ranking of the systems is not consistent

In our experiments we were able to observe 15 different realisations of these testing configurations and hardly did a case of consistency at a compression rate across clusters appeared.

[4] In our experiments compression is always performed at the sentence level

Ref 2 - 1197	**10%**	**20%**	**30%**	**40%**	**50%**
Reference 1	0.50 - 1	0.67 - 1	0.73 - 1	0.73 - 1	0.79 - 1
Reference 3	0.34 - 2	0.51 - 2	0.52 - 2	0.63 - 2	0.69 - 2
Ref 2 - 125	**10%**	**20%**	**30%**	**40%**	**50%**
Reference 1	0.36 - 1	0.41 - 1	0.59 - 2	0.67 - 2	0.78 - 1
Reference 3	0.20 - 2	0.46 - 2	0.66 - 1	0.73 - 1	0.73 - 2

Table 1: Reference summary similarity scores and rankings across clusters and compression rates

Reference 3	**10%**	**20%**	**30%**	**40%**	**50%**
Query-based	0.44 - 2	0.50 - 1	0.58 - 1	0.66 - 1	0.71 - 1
Simple 1	0.10 - 3	0.23 - 3	0.48 - 3	0.57 - 3	0.64 - 3
Simple 2	0.52 - 1	0.45 - 2	0.53 - 2	0.62 - 2	0.68 - 2
Simple 3	0.03 - 4	0.07 - 4	0.08 - 4	0.11 - 4	0.11 - 4

Table 2: System scores and rankings for cluster 241, against Reference 3, at different compression rates

- For Reference All across Clusters at multiple Compressions, the ranking of the systems is consistent

Estimating similarity scores against Reference All (use of multiple references cf. 3.2), proves to provide reliable, consistent results across clusters and compression rates. Table 3 presents the scores and corresponding system rankings for two different clusters and at the five different compression rates. The prevailing system ranking is [1324], which is what we would intuitively expect according to the features of the summarizers we compare. Some deviations from this ranking are due to very small differences in the similarity scores assigned to the systems[5], which indicates the need for using a larger testing corpus for the experiments.

So, the need for multiple references is evident; BLEU is a consistent, reliable metric, but when used in summarization, one has to apply it to multiple references in order to get reliable results. This is not just a way to improve correlation with human judgement (Lin and Hovy, 2002); it is a crucial evaluation parameter that affects the quality of the automatic evaluation results. In our case we had a balanced set of reference summaries to work with, i.e none of them was too similar to another. The more reference summaries one has and the larger one's testing corpus, the safer the conclusions drawn will be. However, what happens when there is lack of such resources and especially of multiple reference summaries? Is there a way to use BLEU with a single reference summary and still get reliable results back?

Looking at the results of our experiments, when using each reference summary separately as a gold standard, we realised that estimating the average ranking of each system across multiple compression rates might lead to consistent rankings. Following the average rank aggregation techique (Rajman and Hartley, 2001), we transfered the average scores each system got per text cluster at each compression rate into ranks and computed the average rank of each system across all five compression rates per text cluster and against each reference summary. Table 4, shows the average system rankings we got for each system at clusters 1197 and 125, using Reference 1, 2, and 3 separately. [1324] is the average system ranking that is clearly indicated in the vast majority of cases. The two exceptions to this are due to extremely small differences in average scores at specific compression rates and indicate the need for scaling up our experiment, a fact that has already been indicated by the results of our experiment using multiple references (Reference All).

4 Conclusions and Future Work

BLEU has been developed for measuring content similarity in terms of length and wording between texts. For the evaluation of automatically generated extracts, the metric is expected to capture similarities between sentences not shared by both the generated text and the model sum-

[5] For example, at the 10% compression rate, cluster 1197, systems Simple 1 and Simple 2 swap places in the final ranking with a 0.005 difference in their similarity scores

Ref All - 1197	**10%**	**20%**	**30%**	**40%**	**50%**
Query based	0.55 - 1	0.47 - 1	0.49 - 1	0.62 - 1	0.63 - 2
Simple 1	0.3184 - 2	0.32 - 3	0.40 - 3	0.49 - 3	0.62 - 3
Simple 2	0.3134 - 3	0.39 - 2	0.44 - 2	0.56 - 2	0.67 - 1
Simple 3	0.02 - 4	0.03 - 4	0.07 - 4	0.11 - 4	0.13 - 4
Ref All - 125	**10%**	**20%**	**30%**	**40%**	**50%**
Query based	0.44 - 1	0.43 - 1	0.57 - 1	0.72 - 1	0.7641 - 2
Simple 1	0.18 - 3	0.3684 - 2	0.54 - 2	0.60 - 3	0.68 - 3
Simple 2	0.32 - 2	0.3673 - 3	0.44 - 3	0.66 - 2	0.7691 - 1
Simple 3	0.03 - 4	0.06 - 4	0.07 - 4	0.10 - 4	0.14 - 4

Table 3: Systems' similarity scores and rankings using Reference All as gold standard

	10%	**20%**	**30%**	**40%**	**50%**	**Average Rank**
Ref 1 - 125	1324	1234	2134	1324	1234	**1234**
Ref 2 - 125	1324	1324	1324	1324	2314	**1324**
Ref 3 - 125	2314	2314	1324	1324	2314	**2314**
Ref 1 - 1197	1324	2314	1324	1324	2314	**1324**
Ref 2 - 1197	1324	1324	1324	1324	2314	**1324**
Ref 3 - 1197	1324	1324	1324	1324	2314	**1324**

Table 4: Systems' average rankings resulting from ranks at multiple compression rates in clusters 125 and 1197. (Systems assumed to be listed in alphabetical order: Query-based, Simple1, Simple2, Simple3)

mary. Going through the texts scored in the above experiments, we found cases in which BLEU does not actually capture content similarity to such a granularity that a human would. Sometimes, this is because the order of the words forming n-grams differs slightly but still conveys the same meaning (e.g. "...abusers reported..." vs. "...reported abusers...") and most of the times because there is no way to capture cases of synonymy, paraphrasing (e.g. "downward tendency"/"falling trend"/"decrease") and other deeper semantic equivalence (e.g. "number of X" vs. "9,000 of X"). Such phenomena are -of course- expected from a statistical metric which involves no linguistic knowledge at all. Our aim in this paper was to shed some light on the conditions under which the metric performs reliably within summarization, given the different parameters that affect evaluation in this NLP research area. From the results obtained by our preliminary experiments, we have generally concluded that:

- Running BLEU over system generated summaries using a single reference affects the reliability of the results provided by the metric. The use of multiple references is a *sine qua non* for reliable results
- Running BLEU over system generated summaries at multiple compression rates and estimating the average rank of each system might yield consistent and reliable results even with a single reference summary and therefore compensate for lack of multiple reference summaries

In order to draw more safe conclusions, we need to scale our experiments considerably, and this is already in progress. Many research questions need still to be answered, such as how BLEU scores correlate with results produced by other content-based metrics used in summarization and elsewhere. We hope that this preliminary, experimental work on porting evaluation metrics across different NLP research areas will function as a stimulus for extensive and thorough research in this direction.

References

H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. 2002. GATE: A framework and graphical development environment for robust NLP tools and applications. In *ACL 2002*.

G. Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurence statistics. In *Proceedings of HLT 2002, Human Language Technology Conference, San Diego, CA.*

R. Donaway, K. Drummey, and L. Mather. 2000. A comparison of rankings produced by summarization evaluation measures. In *Proceedings of the ANLP-NAACL 2000 Workshop on Automatic Summarization, Advanced Natural Language Processing - North American of the Association for Computational Linguistics Conference, Seattle, DC.*

E. Hovy, M. King, and A. Popescu-Belis. 2002. An introduction to machine translation evaluation. In *Proceedings of the LREC 2002 Workshop on Machine Translation Evaluation: Human Evaluators Meet Automated Metrics, Language Resources and Evaluation Conference*. European Language Resources Association (ELRA).

Ch. Lin and E. Hovy. 2002. Manual and automatic evaluation of summariess. In *Proceedings of the ACL 2002 Workshop on Automatic Summarization, Association for Computation Linguistics, Philadelphia, PA.*

I. Mani, T. Firmin, and B. Sundheim. 2001. Summac: A text summarization evaluation. *Natural Language Engineering*.

I. Mani. 2001. Summarization evaluation: an overview. In *Proceedings of the NAACL 2001 Workshop on Automatic Summarization, North Chapter of the Association for Computational Linguistics, Pittsburgh, PA.*

K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2001. Bleu: a method for automatic evaluation of machine translation. Technical Report RC22176 (W0 109-022), IBM Research Division.

Dr. Radev, J. Hongyan, and M. Budzikowska. 2000. Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In *ANLP/NAACL Workshop on Summarization*, Seattle, WA, April.

M. Rajman and A. Hartley. 2001. Automatically predicting mt system rankings compatible with fluency, adequacy or informativeness scores. In *Proceedings of the MT Summit 2001 Workshop on Machine Translation Evaluation: Who did what to whom, European Association for Machine Translation, Santiago de Compostella, Spain.*

H. Saggion, D. Radev., S. Teufel, L. Wai, and S. Strassel. 2002. Developing infrastructure for the evaluation of single and multi-document summarization systems in a cross-lingual environment. In *3rd International Conference on Language Resources and Evaluation (LREC 2002)*, pages 747–754, Las Palmas, Gran Canaria, Spain.

H. Saggion. 2002. Shallow-based Robust Summarization. In *Automatic Summarization: Solutions and Perspectives*, ATALA, December, 14.

Karen Sparck Jones and Julia R. Galliers. 1995. *Evaluating Natural Language Processing Systems: An Analysis and Review*. Number 1083 in Lecture Notes in Artificial Intelligence. Springer.

D. Zajic, B. Dorr, and R. Schwartz. 2002. Automatic headline generation for newspaper stories. In *Proceedings of the ACL 2002 Workshop on Automatic Summarization, Association for Computation Linguistics, Philadelphia, PA.*

Intrinsic versus Extrinsic Evaluations of Parsing Systems

Diego Mollá
Centre for Language Technology
Department of Computing
Macquarie University
Sydney, NSW 2109, Australia
diego@ics.mq.edu.au

Ben Hutchinson
Division of Informatics
University of Edinburgh
Edinburgh EH8 9LW, United Kingdom
B.Hutchinson@sms.ed.ac.uk

Abstract

A wide range of parser and/or grammar evaluation methods have been reported in the literature. However, in most cases these evaluations take the parsers independently (intrinsic evaluations), and only in a few cases has the effect of different parsers in real applications been measured (extrinsic evaluations). This paper compares two evaluations of the Link Grammar parser and the Conexor Functional Dependency Grammar parser. The parsing systems, despite both being dependency-based, return different types of dependencies, making a direct comparison impossible. In the intrinsic evaluation, the accuracy of the parsers is compared independently by converting the dependencies into grammatical relations and using the methodology of Carroll et al. (1998) for parser comparison. In the extrinsic evaluation, the parsers' impact in a practical application is compared within the context of answer extraction. The differences in the results are significant.

1 Introduction

Parsing is a principal stage in many natural language processing (NLP) systems. A good parser is expected to return an accurate syntactic structure of a sentence. This structure is typically forwarded to other modules so that they can work with unambiguous and well-defined structures representing the sentences. It is to be expected that the performance of an NLP system quickly degrades if the parsing system returns incorrect syntactic structures, and therefore an evaluation of parsing coverage and accuracy is important.

According to Galliers and Sparck Jones (1993), there are two main criteria in performance evaluation: "*Intrinsic* criteria are those relating to a system's objective, *extrinsic* criteria those relating to its function i.e. to its role in relation to its setup's purpose." (Galliers and Sparck Jones, 1993, p22). Thus, an intrinsic evaluation of a parser would analyse the accuracy of the results returned by the parser as a stand-alone system, whereas an extrinsic evaluation would analyse the impact of the parser within the context of a broader NLP application.

There are currently several parsing systems that attempt to achieve a wide coverage of the English language (such as those developed by Collins (1996), Järvinen and Tapanainen (1997), and Sleator and Temperley (1993)). There is also substantial literature on parsing evaluation (see, for example, work by Sutcliffe et al. (1996), Black (1996), Carroll et al. (1998), and Bangalore et al. (1998)). Recently there has been a shift from constituency-based (e.g. counting crossing brackets (Black et al., 1991)) to dependency-based evaluation (Lin, 1995; Carroll et al., 1998). Those evaluation methodologies typically focus on comparisons of stand-alone parsers (intrinsic evaluations). In this paper we report on the comparison between an intrinsic evaluation and an evaluation of the impact of the parser in a real application (an extrinsic evaluation).

We have chosen answer extraction as an example of a practical application within which to test the parsing systems. In particular, the extrinsic evaluation uses ExtrAns, an answer extraction system that operates over Unix manual pages (Mollá et al., 2000). The two grammar systems to compare are Link Grammar (Sleator and Temperley, 1993) and the Conexor Functional Dependency Grammar parser (Tapanainen and Järvinen, 1997) (henceforth referred to as Conexor FDG). These parsing systems were chosen because both include a dependency-based parser and a comprehensive grammar of English. However, the structures returned are so different that a direct comparison between them is not straightforward. In Section 2 we review the main differences between Link Grammar and Conexor FDG. In Section 3 we present the intrinsic comparison of parsers, and in Section 4 we comment on the extrinsic comparison within the context of answer extraction. The results of the evaluations are discussed in Section 5.

2 Link Grammar and Conexor FDG

Link Grammar (Sleator and Temperley, 1993) is a grammar theory that is strongly dependency-based. A freely available parsing system that implements the Link Grammar theory has been developed at Carnegie Mellon University. The parsing system includes an extensive grammar and lexicon and has a wide coverage of the English language. Conexor FDG (Tapanainen and Järvinen, 1997) is a commercial parser and grammar, based on the theory of Functional Dependency Grammar, and was originally developed at the University of Helsinki.

Despite both being dependency-based, there are substantial differences between the structures returned by the two parsers. Figure 1 shows Link Grammar's output for a sample sentence, and Figure 2 shows the dependency structure returned by Conexor FDG for comparison. Table 1 explains the dependency types used in the dependency structures of the figures.

The differences between the dependency structures returned by Link Grammar 2.1 and Conexor FDG 3.6 can be summarised as follows.

Direction of dependency: Link Grammar's 'links', although similar to true dependencies, do not state which participant is the head and which is the dependent. However, Link Grammar uses different link types for head-right links and head-left links, so this information can be recovered. Conexor FDG always indicates the direction of the dependence.

Clausal heads: Link Grammar generally chooses the front-most element to be the head of a clause, rather than the main verb. This is true of both matrix and subordinate clauses, as exemplified by the `Wd` and `R` links in Figure 1. Conexor FDG follows the orthodox convention of choosing the main verb as the head of the clause.

Graph structures: Link Grammar's links combine dependencies at the surface-syntactic and deep-syntactic levels (e.g., the link `Bs`, which links a noun modified by a subject-type relative clause to the relative clause's head verb, in Figure 1 indicates a deep-syntactic dependency). The resulting structures are graphs rather than trees. An example is shown in Figure 1, where the noun *man* modified by a relative clause is linked to both the complementiser and the head verb of the relative clause.

Conjunctions: Our version of Link Grammar analyses a coordinating conjunction as the head of a coordinated phrase (Figure 1). This is a modification of Link Grammar's default behaviour which returns a list of parses, one parse per conjunct. However in Conexor FDG's analyses the head will be either the first or the last conjunct, depending on whether the coordinated phrase's head lies to the left or to the right (Figure 2).

Dependency types: Link Grammar uses a set of about 90 link types and many subtypes, which address very specific syntactic constructions (e.g. the link type `EB` connects adverbs to forms of *be* before a noun phrase or prepositional phrase: *He is APPARENTLY a good programmer*). On the other hand, Conexor FDG uses a set of 32 de-

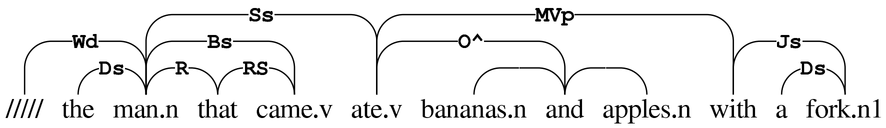

Ss
MVp
Wd
Bs
O^
Js
Ds
R
RS
Ds
///// the man.n that came.v ate.v bananas.n and apples.n with a fork.n1

Figure 1: Output of Link Grammar.

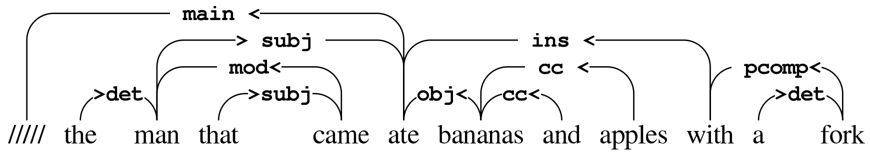


Figure 2: Dependency structure returned by Conexor FDG.

pendency relations, ranging from traditional grammatical functions (e.g. subject, object), to specific types of modifiers (e.g. frequency, duration, location).

Both Conexor FDG and Link Grammar also return non-dependency information. For Link Grammar, this consists of some word class information, shown as suffixes in Figure 1. For Conexor FDG, the base form morphological information of each word is returned, along with a "functional" tag or morpho-syntactic function and a "surface syntactic" tag for each word.[1]

3 Intrinsic Evaluations

Given that both parses are dependency-based, intrinsic evaluations that are based on constituency structures (e.g. (Black et al., 1991)) are hard to perform. Dependency-based evaluations are not easy either: directly comparing dependency graphs (as suggested by Lin (1995), for example) becomes difficult given the differences between the structures returned by the Link Grammar parser and Conexor FDG. We therefore need an approach that is independent from the format of the parser output. Following Carroll et al. (1998) we use *grammatical relations* to compare the accuracy of Link Grammar and Conexor FDG. Carroll et al. (1998) propose a set of twenty parser-independent grammatical relations arranged in a hierarchy representing different degrees of specificity. Four relations from the hierarchy are shown in Table 2. The arguments to each relation specify a head, a dependent, and possibly an initial grammatical relation (in the case of SUBJ in passive sentences, for example) or the 'type', which specifies the word introducing the dependent (in the case of XCOMP).

For example, the grammatical relations of the sentence *the man that came ate bananas and apples with a fork without asking* has the following relations:

```
SUBJ(eat,man,_),
OBJ(eat,banana),
OBJ(eat,apple),
MOD(fork,eat,with),
SUBJ(come,man,_),
MOD(that,man,come),
XCOMP(without,eat,ask)
```

The terms 'head' and 'dependent' used by Carroll et al. (1998) to refer to the arguments of grammatical relations should not be confused with the similar terms in the theory of dependency grammar. Grammatical relations and dependency arcs represent different phenomena. An example should suffice to illustrate the difference; consider *The man that came ate bananas and apples with a fork.* In dependency grammar a unique head is assigned to each word, for example the head of *man* is *ate*. However *man* is the dependent of more than one grammatical relation, namely SUBJ(eat,man,_) and SUBJ(come,man,_). Furthermore, in dependency grammar a word can have at most one dependent of each argument type, and so *ate* can have at most one object, for example. But

[1] See (Järvinen and Tapanainen, 1997) for more information on the output from Conexor FDG.

Link Grammar		**Conexor FDG**	
Name	*Description*	*Name*	*Description*
Bs	Singular external object of relative clause	cc	Coordination
Ds	Singular determiner	det	Determiner
Js	Singular object of a preposition	ins	<*not documented*>
MVp	Verb-modifying preposition	main	Main element
O^	Object	mod	General post-modifier
R	Relative clause	obj	Object
RS	Part of subject-type relative clause	pcomp	Prepositional complement
Ss	Singular subject	subj	Subject
Wd	Declarative sentence		

Table 1: Some of the dependency types used by Link Grammar and Conexor FDG.

Relation	*Description*
SUBJ(head, dependent, initial_gr)	Subject
OBJ(head, dependent)	Object
XCOMP(type, head, dependent)	Clausal complement without an overt subject
MOD(type, head, dependent)	Modifier

Table 2: Grammatical relations used in the intrinsic evaluation.

the same is not true for grammatical relations, and we get both OBJ(eat,banana) and OBJ(eat,apple).

3.1 Accuracy

Our intrinsic evaluation began on the assumption that grammatical relations could be deduced from the dependency structures returned by the parsers. In practise, however, this deduction process is not always straightforward; for example complexity arises when arguments are shared across clauses. In addition, Link Grammar's analysis of the frontmost elements as clausal heads complicates the grammatical relation deduction when there are modifying clauses.

An existing corpus of 500 sentences/10,000 words annotated with grammatical relations was used for the evaluation (Carroll et al., 1999). We restricted the evaluation to just the four relations shown in Table 2. This decision had two motivations. Firstly, since the dependency parsers' output did not recognise some distinctions made in the hierarchy of relations, it did not make sense to test these distinctions. Secondly, we wanted the deduction of grammatical relations to be as simple a process as possible, to minimise the chance of introducing errors. This second consideration also led us to purposefully ignore the sharing of arguments induced by control verbs, as this could not always be deduced reliably. Since this was done for both parsers the comparison remains meaningful.

Algorithms for producing grammatical relations from Link Grammar and Conexor FDG output were developed and implemented. The results of parsing the corpus are shown in Table 3. Since Conexor FDG returns one parse per sentence only and Link Grammar returns all parses ranked, the first (i.e. the best) parse returned by Link Grammar was used in the intrinsic evaluation.

The table shows significantly lower values of recall and precision for Link Grammar. This is partly due to the fact that Link Grammar's links often do not connect the head of the clause, as we have seen with the Wd link in Figure 1.

3.2 Speed

Link Grammar took 1,212 seconds to parse the 10,000 word corpus, while Conexor FDG took 20.5 seconds. This difference is due partly to the fact that Link Grammar finds and returns multiple (and often many) alternative parses. For example,

		With Link Grammar	*With Conexor FDG*
Precision	SUBJ	50.3%	73.6%
	OBJ	48.5%	84.8%
	XCOMP	62.2%	76.2%
	MOD	57.2%	63.7%
	Average	***54.6%***	***74.6%***
Recall	SUBJ	39.1%	64.5%
	OBJ	50%	53.4%
	XCOMP	32.1%	64.7%
	MOD	53.7%	56.2%
	Average	***43.7%***	***59.7%***

Table 3: Accuracy of identification of grammatical relations.

Link Grammar found a total of 410,509 parses of the 505 corpus sentences.

4 Extrinsic Evaluations

It is important to know not only the accuracy of a parser but how possible parsing errors affect the success of an NLP application. This is the goal of an extrinsic evaluation, where the system is evaluated in relation to the embedding setup. Using answer extraction as an example of an NLP application, we compared the performance of the Link Grammar system and Conexor FDG.

4.1 Answer Extraction and ExtrAns

The fundamental goal of Answer Extraction (AE) is to locate those exact phrases of unedited text documents that answer a query worded in natural language. AE has received much attention recently, as the increasingly active Question Answering track in TREC demonstrates (Voorhees, 2001b; Voorhees, 2001a).

ExtrAns is an answer extraction system that operates over UNIX manual pages (Mollá et al., 2000). A core process in ExtrAns is the production of semantic information in the shape of logical forms for each sentence of each manual page, as well as the user query. These logical forms are designed so that they can be derived from *any* sentence (using robust approaches to treat very complex or ungrammatical sentences), and they are optimised for NLP tasks that involve the semantic comparison of sentences, such as AE.

ExtrAns' logical forms are called *minimal logical forms* (MLFs) because they encode the minimum information required for effective answer extraction. In particular, only the main dependencies between the verb and arguments are expressed, plus modifier and adjunct relations. Thus, complex quantification, tense and aspect, temporal relations, plurality, and modality are not expressed.

The MLFs use *reification* to achieve flat expressions, very much in the line of Davidson (1967), Hobbs (1985), and Copestake et al. (1997). In the current implementation only reification to objects, eventualities (events or states), and properties is applied. For example, the MLF of the sentence *cp will quickly copy files* is:

```
holds(e4),
object(cp,o1,[x1]),
object(s_command,o2,[x1]),
evt(s_copy,e4,[x1,x6]),
object(s_file,o3,[x6]),
prop(quickly,p3,[e4]).
```

In other words, there is an entity `x1` which represents an object of type `command`;[2] there is an entity `x6` (a file); there is an entity `e4`, which represents a copying event where the first argument is `x1` and the second argument is `x6`; there is an entity `p3` which states that `e4` is done quickly, and the event `e4`, that is, the copying, holds.

ExtrAns finds the answers to the questions by converting the MLFs of the questions into Prolog queries and then running Prolog's default resolution mechanism to find those MLFs that can prove the question.

This default search procedure is called the *synonym mode* since ExtrAns uses a small WordNet-style thesaurus (Fellbaum, 1998) to convert all the synonyms into a synonym representative. ExtrAns also has an *approximate mode* which, besides normalising all synonyms, scores all document sentences on the basis of the maximum number of predicates that unify between the MLFs of the query and the answer candidate (Mollá et al., 2000). If all query predicates can be matched then

[2] ExtrAns uses additional domain knowledge to infer that *cp* is a command.

the approximate mode returns exactly the same answers as the synonym mode.

4.2 The Comparison

Ideally, answer extraction systems should be evaluated according to how successful they are in helping users to complete their tasks. The use of the system will therefore depend on such factors as how many potential answers the user is presented with at a time, the way these potential answers are ranked, how many potential answers the user is prepared to read while searching for an actual answer, and so on. These issues, though important, are beyond the scope of the present evaluation. In this evaluation we focus solely on the relevance of the set of results returned by ExtrAns.

4.2.1 Method

Resources from a previous evaluation of ExtrAns (Mollá et al., 2000) were re-used for this evaluation. These resources were: a) a collection of 500 man pages, and b) a test set of 26 queries and relevant answers found in the 500 manual pages. The careful and labour-intensive construction of the test set gives us confidence that practically all relevant answers to each query are present in the test set. The queries themselves were selected according to the following criteria:

- There must be at least one answer in the manual page collection.
- The query asks how to perform a particular action, or how a particular command works.
- The query is simple, i.e. it asks only one question.

The manual pages were parsed using Conexor FDG and Link Grammar. The latter has a parameter for outputting either all parses found, or just the best parse found, and both parameter settings were used. The queries were then parsed by both parsers and their logical forms were used to search the respective databases. The experiment was repeated using both the synonym and approximate search modes.

Parser	*Precision*[4]	*Recall*	*F-score*
Conexor FDG	55.8%	8.9%	0.074
LG–best	49.7%	11.4%	0.099
LG–all	50.9%	13.1%	0.120

Table 4: Averages per query in synonym mode.

Parser	*Precision*[4]	*Recall*	*F-score*
Conexor FDG	28.3%	21.9%	0.177
LG–best	31.8%	15.8%	0.150
LG–all	40.5%	20.5%	0.183

Table 5: Averages per query in approximate mode.

4.2.2 Results

Precision, Recall and the F-score (with Precision and Recall equally weighted) for each query were calculated.[3] When no results were returned for a query the precision could not be calculated, but the F-score is equal to zero. The results are shown in Tables 4 and 5. The number of times the results for a query contained no relevant answers are shown in Table 6.

The tables show that the approximate mode gives better results than the synonym mode. This is to be expected, since the synonym mode returns exact matches only and therefore some questions may not produce any results. For those questions, recall and F would be zero. In fact, the number of questions without answers in the synonym mode is so large that the comparison between Conexor FDG and Link Grammar becomes unreliable in this mode. In this discussion, therefore, we will focus on the approximate mode.

The results returned by Link Grammar when all parses are considered are significantly better than when only the first (i.e. the best) parse is consid-

[3] F was calculated using the expression

$$F = 2 \times \frac{|\text{returned and relevant}|}{|\text{returned}| + |\text{relevant}|}$$

which is equivalent to the usual formulation (with $\beta = 1$):

$$F = (\beta^2 + 1) \times \frac{\text{Precision} \times \text{Recall}}{\beta^2\text{Precision} + \text{Recall}}$$

[4] Average over queries for which precision is defined, i.e. when the number of returns is non-zero.

Parser	*Search mode*	*No results returned*	*Nothing relevant returned*
Con. FDG	Synonym	20	20
Con. FDG	Approximate	0	8
LG–best	Synonym	16	18
LG–best	Approximate	1	11
LG–all	Synonym	15	18
LG–all	Approximate	4	12

Table 6: Numbers of times no relevant answers were found.

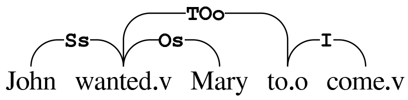
ered. This shows that, in the answer extraction task, it is better to use the logical forms of all possible sentence interpretations. Recall increases and, remarkably, precision increases as well. This means that the system is more likely to include new relevant answers when all parses are considered.

In many applications it is more practical to consider one parse only. Conexor FDG, for example, returns one parse only, and the parsing speed comparison (Section 3.2) shows an important difference in parsing time. If we compare Conexor FDG with Link Grammar set to return just the best parse — since Conexor FDG returns one parse only, this is the fairest comparison — we can see that recall of the system using Conexor FDG is higher than that of the system using Link Grammar, while retaining similar precision.

5 Discussion

The fairest extrinsic comparison between Conexor FDG and Link Grammar is the one that uses the best parse returned by Link Grammar, and the answer extraction method follows the approximate mode. With these settings, Conexor FDG produces better results than Link Grammar. However, the results of the extrinsic comparison are far less dramatic than those of the intrinsic comparison, specially in the precision figures.

One reason for the difference in the results is that the intrinsic evaluation compares grammatical relation accuracy, whereas the answer extraction system used in the extrinsic evaluation uses logical forms. A preliminary inspection of the grammatical relations and logical forms of questions and correct answers shows that high overlap of grammatical relations does not translate into high overlap of logical forms. A reason for this difference is that the semantic interpreters used in the extrinsic evaluation explore exhaustively the dependency structures returned by both parsing systems and they try to recover as much information as possible. In contrast with this, the generators of grammatical relations used in the intrinsic evaluation provide the most direct mapping from dependency structures to grammatical relations. For example, typically a dependency structure would not show a long dependency like the subject of *come* in the sentence *John wanted Mary to come*:



As a result, the grammatical relations would not show the subject of *come*. However, the subject of *come* can be traced by following several dependencies (`I`, `TOo` and `Os` above) and ExtrAns' semantic interpreters *do* follow these dependencies. In other words, the semantic interpreters use more information than what is directly encoded in the dependency structures. Therefore, the logical forms contain richer information than the grammatical relations. We decided not to optimise the grammatical relations used in our evaluation because we wanted to test the expressivity of the inherent grammars. It would be questionable whether we should recover more information than what is directly expressed. After all, provided that the parse contains all the words in the original order, we can theoretically ignore the sentence structure and still recover *all* the information.

6 Summary and Further Work

We have performed intrinsic evaluations of parsers and extrinsic evaluations within the context of answer extraction. These evaluations strengthen Galliers and Sparck Jones (1993)'s claim that intrinsic evaluations are of very limited value. In particular, our evaluations show that intrinsic evaluations may provide results that are distorted with respect to the most intuitive purpose of a parsing system: to deliver syntactic structures to subsequent modules of practical NLP systems. There is a clear need for frameworks for extrinsic evaluations of parsers for different NLP applications.

Further research to confirm this conclusion will be to try and minimise the occurrence of variables in the experiments by using the same corpus for both the intrinsic and the extrinsic evaluations and/or by using an answer extraction system that operates on the level of grammatical relations instead of MLFs. Additional further research will be the use of other intrinsic evaluation methodologies and extrinsic evaluations within the context of various other embedding setups.

Acknowledgement

This research is supported by the Macquarie University New Staff grant MUNS–9601/0069.

References

Srinivas Bangalore, Anoop Sarkar, Christine Doran, and Beth Ann Hockey. 1998. Grammar & parser evaluation in the XTAG project. In *Proc. Workshop on the Evaluation of Parsing Systems, LREC98.*

Ezra Black, S.P. Abney, D. Flickinger, C. Gdaniec, R. Grisham, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M.P. Marcus, S. Roukos, B. Santorini, and T. Strzalkowski. 1991. A procedure for quantitatively comparing the syntactic coverage of English grammars. In *Proc. DARPA Speech and Natural Language Workshop*, pages 306–311, Pacific Grove, CA. Morgan Kaufmann.

Ezra Black. 1996. Evaluation of broad-coverage natural-language parsers. In Ronald A. Cole, Joseph Mariani, Hans Uszkoreit, Annie Zaenen, and Victor Zue, editors, *Survey of the State of the Art in Human Language Technology*, pages 488–490. CSLU, Oregon Graduate Institute.

John Carroll, Ted Briscoe, and Antonio Sanfilippo. 1998. Parser evaluation: a survey and a new proposal. In *Proc. LREC98.*

John Carroll, G. Minnen, and T. Briscoe. 1999. Corpus annotation for parser evaluation.

Michael John Collins. 1996. A new statistical parser based on bigram lexical dependencies. In *Proc. ACL.* Santa Cruz.

Ann Copestake, Dan Flickinger, and Ivan A. Sag. 1997. Minimal recursion semantics: an introduction. Technical report, CSLI, Stanford University, Stanford, CA.

Donald Davidson. 1967. The logical form of action sentences. In Nicholas Rescher, editor, *The Logic of Decision and Action*, pages 81–120. Univ. of Pittsburgh Press.

Christiane Fellbaum. 1998. Wordnet: Introduction. In Christiane Fellbaum, editor, *WordNet: an electronic lexical database*, Language, Speech, and Communication, pages 1–19. MIT Press, Cambrige, MA.

Julia R. Galliers and Karen Sparck Jones. 1993. Evaluating natural language processing systems. Technical Report TR-291, Computer Laboratory, University of Cambridge.

Jerry R. Hobbs. 1985. Ontological promiscuity. In *Proc. ACL'85*, pages 61–69. University of Chicago, Association for Computational Linguistics.

Timo Järvinen and Pasi Tapanainen. 1997. A dependency parser for english. Technical Report TR-1, Department of Linguistics, University of Helsinki, Helsinki.

Dekang Lin. 1995. A dependency-based method for evaluating broad-coverage parsers. In *Proc. IJCAI-95*, pages 1420–1425, Montreal, Canada.

Diego Mollá, Rolf Schwitter, Michael Hess, and Rachel Fournier. 2000. Extrans, an answer extraction system. *T.A.L.*, 41(2):495–522.

Daniel D. Sleator and Davy Temperley. 1993. Parsing English with a link grammar. In *Proc. Third International Workshop on Parsing Technologies*, pages 277–292.

Richard F. E. Sutcliffe, Heinz-Detlev Koch, and Annette McElligott, editors. 1996. *Industrial Parsing of Software Manuals*. Rodopi, Amsterdam.

Pasi Tapanainen and Timo Järvinen. 1997. A non-projective dependency parser. In *Procs. ANLP-97*. ACL.

Ellen M. Voorhees. 2001a. Overview of the TREC 2001 question answering track. In Ellen M. Voorhees and Donna K. Harman, editors, *Proc. TREC-10*, number 500-250 in NIST Special Publication. NIST.

Ellen M. Voorhees. 2001b. The TREC question answering track. *Natural Language Engineering*, 7(4):361–378.

Adaptation of the F-measure to Cluster Based Lexicon Quality Evaluation

Angelo Dalli
NLP Research Group
Department of Computer Science
University of Sheffield
a.dalli@dcs.shef.ac.uk

Abstract

An external lexicon quality measure called the L-measure is derived from the F-measure (Rijsbergen, 1979; Larsen and Aone, 1999). The typically small sample sizes available for minority languages and the evaluation of Semitic language lexicons are two main factors considered. Large-scale evaluation results for the Maltilex Corpus are presented (Rosner et al., 1999).

1 Introduction

Computational Lexicons form a fundamental component of any NLP system. Unfortunately, good quality lexicons are hard to create and maintain. The labour intensive process of lexicon creation is further compounded when minority languages are concerned. Inevitably, computational lexicons for minor languages tend to be quite small when compared to computational lexicons available for more common languages such as English.

The Maltilex Corpus is used in this paper to evaluate a cluster based lexicon quality measure adapted from the F-measure. The Maltilex Corpus is the first large-scale computational lexicon for Maltese (Rosner et al., 1999). The choice of Maltese as the evaluation language presented some additional problems due to the Semitic morphology and grammar of Maltese (Mifsud, 1995). An innovative approach to lexicon creation using an automated technique called the Lexicon Structuring Technique (LST) was used to create an initial computational lexicon from a wordlist (Dalli, 2002a). LST decreased the amount of work that is normally required to create a lexicon from scratch by adapting a number of clustering, alignment, and approximate matching techniques to produce a set of clusters containing related wordforms. Lexicon clusters are thus analogous to lemmas in more traditional lexicons.

This approach has many advantages for a language having a Semitic morphology and grammar due to the large number of wordforms that can be derived for a single lemma. Instead of processing every wordform individually, the whole cluster can be treated as a single entity, reducing processing requirements significantly.

The close relationship of this lexicon definition and standard clustering systems (with lemmas corresponding to clusters), enabled the reuse of cluster quality evaluation measures to the task of lexicon quality evaluation. There are two main ways of evaluating cluster quality which are summarised in (Steinbach et al., 1999 pg. 6) as follows:

- Internal Quality Measure – Clusters are compared without reference to external knowledge against some predefined set of desirable qualities.
- External Quality Measure – Clusters are compared to known external classes.

Internal quality measures are not always desirable, since their very existence implies that better quality can be achieved by applying an internal quality measure in conjunction with some optimisation technique. An internal quality measure for cluster-based lexicons was not available either.

The two main external quality measures applicable lexicon quality evaluation tasks are entropy (Shannon, 1948) and the F-measure (van Rijsbergen, 1979; Larsen and Aone, 1999).

Entropy based quality measures assert that the best entropy that can be obtained is when each cluster contains the optimal number of members. In our context this corresponds to having clusters (corresponding to lemmas) that contain exactly all the wordforms associated with that cluster. The class distribution of the data is calculated by considering the probability of every member belonging to some class. The entropy of every cluster j is calculated using the standard entropy formula $E(j) = -\sum_i p_{ij} \log(p_{ij})$ where p_{ij} denotes the probability that a member of cluster j belongs to class i. The total entropy is then calculated as $E^* = \frac{1}{n}\sum_{j=1}^{m} n_j \cdot E(j)$ where n_j is the size of cluster j, m the number of clusters, and n the total number of data points.

The F-measure treats every cluster as a query and every class as the desired result set for a query. The recall and precision values for each given class are then calculated using information retrieval concepts. The F-measure of cluster j and class i is given by $F(i,j) = \frac{2 \cdot r(i,j) \cdot p(i,j)}{r(i,j) + p(i,j)}$ where r denotes recall and p the precision. Recall is defined as $r(i,j) = \frac{n_{ij}}{n_i}$ and precision is defined as $p(i,j) = \frac{n_{ij}}{n_j}$ where n_{ij} is the number of class i members in cluster j, while n_j and n_i are the sizes of cluster j and class i respectively. The overall F-measure for the entire data set of size n is given by $F^* = \sum_i \frac{n_i}{n} \max[F(i,j)]$.

2 Lexicon Quality Measure

Computational lexicons have an additional domain-specific external quality measure available in the form of existing non-computational language dictionaries. Dictionaries can be used to compare the results generated by the automated system against those produced by human experts. Generally it can be assumed that reputable printed dictionaries are of a very high quality and thus provide a gold standard for comparison. For some languages, especially minority languages, the only available quality data would be in printed dictionary form. Unfortunately most non-computational dictionaries are not amenable to automated analysis techniques since the process of re-inputting and re-structuring data into a computational dictionary format is generally so labour intensive that it becomes too expensive.

Additionally, since every cluster and class correspond to a lemma, the number of classes to be considered is expected to number in the thousands. This would make a straightforward application of the F-measure an overly long process. A modified statistical sampling technique based on the F-measure that gives results that are approximately as good as the full application of the F-measure and that caters for the particular nuances of lexicon quality evaluation is thus needed.

The L-measure is such a new measure based on the F-measure that attempts to measure the quality of a given lexicon in relation to other existing lexicons that are possibly non-computational lexicons (i.e. human compiled language dictionaries), taking into consideration that a full population analysis may not be practical under most circumstances.

2.1 Lexicon Extraction from Dictionaries

The L-Measure works by comparing two lexicons, one derived from a gold standard representation in the form of human compiled dictionaries and the other being a computational lexicon whose quality is being assessed. In order to avoid confusion, formal definitions of the terms dictionary, lexicon and wordlists are now presented.

A dictionary D is formally modeled as a sequence $<t_1 .. t_h>$ of tuples of the form (l, def) where l denotes a lemma (i.e. a dictionary headword in a more traditional sense) and *def* is a 5-tuple (*m*, *r*, *c*, *i*, *o*) with *m* containing morphological information that enables members of the lemma to be inferred or generated, *r* a set of relations to other lemmas, *c* a description of the different contexts where the lemma may be normally used, *i* containing meta-information about lemma *l* itself, and *o* an object containing additional information (such as etymology, examples of common use, etc.) Since multiple entries of the same headword may be present in *D* the sequence is not injective, i.e. the sequence can contain duplicate elements.

The main two differences between a dictionary and a lexicon are that different types of information are stored about every lemma in the *def* component, and secondly, that a lexicon has an injective sequence of tuples (i.e. a sequence that does not have duplicates and where the exact order is important) while a dictionary does not (since a dictionary does not need to force a headword to have one unique entry, especially in the case of printed dictionaries that often have the same headword appearing in multiple top-level entries).

A dictionary *D* can be thus transformed into a lexicon *L*, denoted by *L* = *lex*(*D*), by filtering the tuple sequence $<t_1 .. t_h>$ making up *D* to include only the *l* components of every tuple. The filtered sequence is then transformed into an injective sequence of unique lemmas $<l_1 .. l_u>$, satisfying the requirements for a lexicon. Appropriate transformations have to be defined to transform the def component from dictionary to lexicon format.

The sequence of lemmas is then expanded to a canonical wordlist *W*. A canonical wordlist *W* is a sequence $<w_1 .. w_u>$ of sets of strings generated from a lexicon *L*, denoted by *W* = *can*(*L*), by listing all possible instances of every lemma in the lexicon (i.e. all possible wordforms of a particular lemma), in effect creating a full form lexicon.

The canonical wordlist *W* thus has *u* sets of strings corresponding to *u* lemmas in the lexicon. The particular lemma used to generate a wordform *w* is obtained by the operator *lem*(*w*). The sequence of lemmas used to generate *W* is denoted as *lemmas*(*W*). The union of two wordlists $W_1 \cup W_2$ is defined to be the union of all sets of strings in both wordlists, i.e. $\forall x_i \in W_1, y_j \in W_2 \bullet W_1 \cup W_2 = \langle x_i \cup y_j \rangle$ provided that $lem(x_i) = lem(y_j) \vee lem(x_j) \notin lemmas(W_2) \vee lem(y_j) \notin lemmas(W_1)$ holds.

This definition ensures maximum coverage of the resulting canonical wordlist. An empty or null canonical wordlist results if no pair of strings obey the previously stated condition while the union of a wordlist with a null wordlist is the original wordlist itself.

Similarly the intersection of two wordlists $W_1 \cap W_2$ is defined to be the union of all sets of strings in both wordlists that have corresponding lemmas appearing in both wordlists, i.e. $\forall x_i \in W_1, y_j \in W_2 \bullet W_1 \cap W_2 = \langle x_i \cup y_j \rangle$ provided that $lem(x_i) = lem(y_j)$ holds.

Note that this definition is concerned mainly with the lemmas and their associated wordforms themselves. Since lexicons are not just a list of lemmas and wordforms, other linguistic annotations will have to be evaluated using other techniques appropriate to the particular linguistic annotations added to the lemma entries.

2.2 L-Measure Definition

Given a lexicon *L* and a set of dictionaries $D = \{D_1 .. D_k\}$ transform the set of dictionaries *D* into a set of lexicons $L' = \{L_1 .. L_k\}$ using the *lex* transformation on every dictionary, thus $L' = \bigcup_1^k lex(D_i)$. Define *W* as the canonical wordlist obtained from *L*, *W* = *can*(*L*) and *W'* as the canonical wordlist obtained from *L'*, $W' = \bigcup_1^k can(L_i)$ under canonical wordlist union.

Define *Y* to be the canonical wordlist of words common to both *W* and *W'*, $Y = W \cap W'$. The sample size *S* used for the L-measure is defined as $\alpha.|lemmas(Y)|$ where α is some value in the range (0..1) that controls the random sample size. Typically α should be set to somewhere between 0.01 and 0.1. It is expected that the sample size will be large enough to assume that the sample is representative of the whole population.

The L-measure of a lemma *j* in *lemmas*(*W*) and lemma *i* in *lemmas*(*Y*) is given by

$L(i,j) = \frac{2 \cdot r(i,j) \cdot p(i,j)}{r(i,j) + p(i,j)}$ where r denotes recall and p is the precision. Recall is defined as $r(i,j) = \frac{n_{ij}}{n_i}$ and precision is defined as $p(i,j) = \frac{n_{ij}}{n_j}$ where n_{ij} is the number of lemma i members in lemma j, while n_j and n_i are the sizes of lemma j and lemma i respectively. The overall L-measure for the entire sample of size n is given by $L^* = \sum_i \frac{n_i}{n} \max[L(i,j)]$. L^* is always in the range [0..1] and is proportional to the lexicon quality, with an L^* score of 1 representing a perfect quality lexicon with respect to the lexicon being used as a standard.

Y is used instead of *W'* since lexical word coverage is largely determined by the quality of the corpus used to create the lexicon. While this kind of analysis might be useful in determining the coverage of a lexicon the L-measure is oriented towards measuring quality rather than quantity, independently of the corpus that was used to create the lexicon.

3 Results

The L-measure has been used to measure the quality of the Maltilex Computational Lexicon in relation to existing paper based dictionaries. The most comprehensive dictionary of Maltese was used to produce L', the comparison standard lexicon (Aquilina, 1987-1990). The capability of the L-measure to work with a statistical sample made a manual analysis of results possible without having L' in digital form.

The value for the sample size *S* was determined through a parameter α that was set to 0.01, meaning that 1% of all lemmas in the Maltilex Computational Lexicon were covered by the statistical sample. Since around 63,000 lemmas exist in the combined lexicon the sample size *S* was determined to be 630. The set of 630 lemmas chosen at random from the Maltilex Corpus contained a total of 5,887 wordforms taken from the combined lexicon.

The precision and recall for the samples were calculated individually to obtain the individual L-measure for a range of lemmas. A fully worked out example of the calculation of the L-measure for the lemma `missier` (father) is given. Lemmas in the Maltilex Computational Lexicon are aligned automatically using a technique adopted from bioinformatics and hence the presentation of the wordforms in their aligned format (Dalli, 2000b; Gusfield, 1997).

The lemma `missier` (the Maltese word for *father* with the cluster showing different forms like my father, your father, etc.) taken from the Maltilex Computational Lexicon, which represents lemma *i*, contains seven members as displayed below:

```
m i s s ie r _ _ _ _ _ _   _ _ _
m i s s ie r e k _ _ _ _   _ _ _
m i s s ie r _ _ _ n _ a   _ _ _
m i s s ie r _ k o m _ _   _ _ _
m i s s i  r i _ _ _ j ie t n a
m i s s ie r i _ _ _ _ _   _ _ _
m i s s ie r _ h o m _ _   _ _ _
```

The lemma `missier`, taken from Aquilina's Dictionary, which represents lemma *j*, can be used to generate the following ten members as displayed below:

```
m i s s ie r _ _ _ _ _ _   _ _ _
m i s s ie r e k _ _ _ _   _ _ _
m i s s ie r _ _ _ n _ a   _ _ _
m i s s ie r _ k o m _ _   _ _ _
m i s s i  r i _ _ _ j ie t n a
m i s s ie r i _ _ _ _ _   _ _ _
m i s s ie r a _ _ _ _ _   _ _ _
m i s s ie r _ _ u _ _ _   _ _ _
m i s s ie r _ h o m _ _   _ _ _
m i s s i  r i _ _ _ j ie t _ _
```

For this example, n_j and n_i are thus equal to 10 and 7 respectively. Recall and precision values are calculated as $r(missier, missier') = \frac{7}{7} = 1$ $p(missier, missier') = \frac{7}{10} = 0.7$ respectively.

The L-measure for the lemma `missier` is $L(missier, missier') = \frac{2 \cdot 1 \cdot 0.7}{1 + 0.7} = \frac{1.4}{1.7} = 0.8235$

The overall L-measure for the entire sample of 5,887 wordforms is given by $L^* = \sum_i \frac{n_i}{5887} \max[L(i,j)]$. The contribution of the lemma `missier` to the final L^* score is thus given by $\frac{7}{5887} 0.8235 = 0.000979226$. A high precision floating point library was used to represent the individual contribution values since these are generally very small. Figures 1 and 2 show the precision and recall curves for the whole sample respectively.

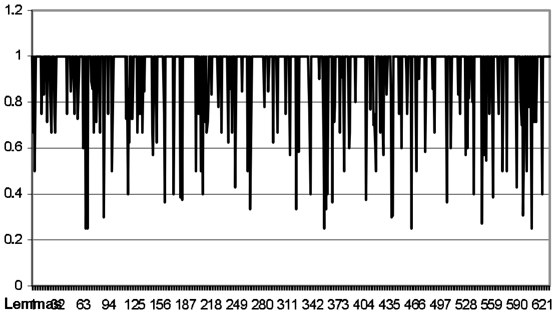


Figure 1 Precision

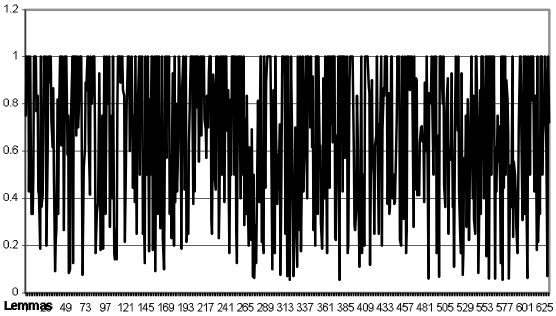


Figure 2 Recall

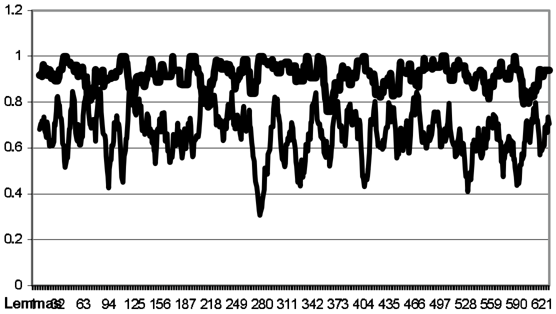


Figure 3 Precision and Recall Trends

Figure 3 shows moving average trendlines for precision and recall (precision is shown in a bold line on top, recall is the fainter line underneath). The average precision was 0.91748 and the average rate of recall was 0.661359.

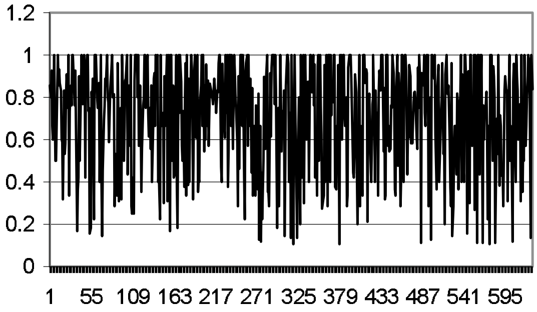


Figure 4 Individual L-Measure Values

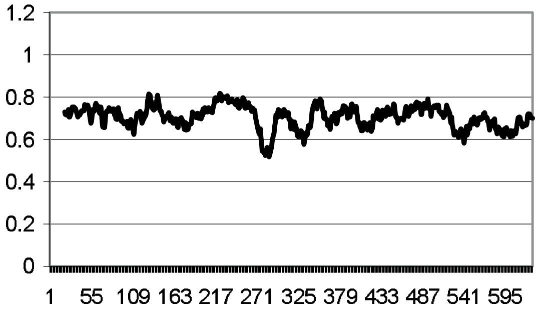


Figure 5 Individual L-Measure Values Trend

Figure 4 shows the individual L-measure values for the sample. The values displayed in Figure 4 are those used to calculate the final L^* value. Figure 5 shows the moving average trendline for the individual L-measure values.

The average individual L-measure was 0.707256882 while the average individual contribution of a lemma to the L^* value was 0.000748924. The variance in the L-measure individual values was 0.065504369.

The correlation between the L-measure and precision was 0.163665769 while the correlation between the L-measure and recall was 0.922214452.

The overall L^* score for the Maltilex Computational Lexicon was 0.4718. This score is quite intuitive when the various problems in the existing Maltese corpus used to create the Computational Lexicon are considered. This score means that the number of wordforms that are stored or that can be generated by the current lexicon needs to be expanded by around 53% in order to match the quality of the lexicon underlying Aquilina's dictionary (Aquilina, 1987-1990).

4 Conclusion

The L-measure is a useful evaluation metric that can be used to measure the quality of a computational lexicon based on clustering concepts. The small data sample required by L-measure to give meaningful results makes it a practical measure to use in a variety of situations where massive amounts of data might not be available. This makes L-measure ideal for use in the evaluation of Language Resources for minority languages and also for quick benchmark studies that evaluate the quality of a computational lexicon as it is being created.

Compared with the F-measure, the L-measure will give highly similar results using less data. Naturally the validity of the L-measure results depends on the choice of the α value, which in turn determines the sample size.

The lemma/cluster based approach of the L-measure is suitable for the evaluation of Semitic language lexicons that often prove problematic to evaluation techniques based on English or Romance languages.

The L-measure also has potential future applications in the comparison and evaluation of different lexicons. The individual L-measure scores can also be used to identify areas of similarities and differences between different lexicons quickly.

The L-measure can also be adapted to other areas of Computational Linguistics as long as the concept of a cluster and some means of determining its precision and recall exist. Minimal changes are needed to adapt the L-measure to other domains making future adaptations likely.

Acknowledgment

This work has been made possible with the collaboration of the Maltilex Project at the University of Malta.

References

Angelo Dalli. 2002a. *Computational Lexicon for Maltese*. M.Sc. Dissertation. Department of Computer Science and AI, University of Malta, Malta.

Angelo Dalli. 2002b. Biologically Inspired Lexicon Structuring Technique. *HLT2002*, San Diego, California.

Bjorner Larsen and Chinatsu Aone. 1999. Fast and Effective Text Mining Using Linear-time Document Clustering. *KDD-99*, San Diego, California.

C. Van Rijsbergen. 1979. *Information Retrieval*, 2nd ed. Butterworth, London.

Claude E. Shannon. 1948. A mathematical theory of communication. *Bell System Technical Journal* 27: 379-423, 623-656.

Dan Gusfield. 1997. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, Cambridge, UK.

Joseph Aquilina. 1987-1990. *Maltese-English Dictionary*. Midsea Books, 2 Volumes, Valletta, Malta.

Manwel Mifsud. 1995. *Loan verbs in Maltese a descriptive and comparative study*. Studies in Semitic languages and linguistics, Brill, Leiden.

Michael Rosner et. al. 1999. Linguistic and Computational Aspects of Maltilex. *ATLAS Symposium*, Tunis.

Michael Steinbach, George Karypis, and Vipin Kumar. 1999. A comparison of document clustering techniques, University of Minnesota, Technical Report 00-034.

No-bureaucracy evaluation

Adam Kilgarriff
ITRI, University of Brighton
adam@itri.brighton.ac.uk

SENSEVAL is a series of evaluation exercises for Word Sense Disambiguation. The core design is in accordance with the MUC and TREC model of quantitative, developer-oriented (rather than user-oriented) evaluation. The first was in 1998, with tasks for three languages and 25 participating research teams, the second in 2001, with tasks for twelve languages, thirty-five participating research teams and over 90 participating systems. The third is currently in planning. The scale of the resources developed is indicated in Table 1 (reproduced from (Edmonds and Kilgarriff, 2002)).[1]

In this paper we address five of the workshop themes from a SENSEVAL perspective:

1. organisational structure
2. re-use of corpus resources: pro and con
3. the web and evaluation
4. SENSEVAL and Machine Translation evaluation
5. re-use of metrics: a cautionary tale.

1 Organisation

One aspect of SENSEVAL of interest here is its organizational structure. It has no centralised sponsor to fund or supply infrastructure. Almost all work was done by volunteer effort with just modest local grant funding for particular subtasks, with organisers answerable to no-one beyond the community of WSD researchers. This was possible because of the level of commitment. People wanted the evaluation framework, so they were willing to find the time, from whatever slack they were able to concoct.

At the SENSEVAL-1 workshop, the possibility of finding an official sponsor –most likely the EU or a branch of the US administration– was discussed at length and vigorously. The prevailing view was that, while it was nice to have more money around, it was not necessary and came at a cost. Various experiences were cited where researchers felt their energies had been diverted from the research itself to the processes of grant applications, cost statements, and the strange business of writing reports which in all likelihood no-one will ever read. My experience, as co-ordinator of SENSEVAL-1 and chair of SENSEVAL-2, was that, without external funding but with great goodwill and energy for the task at various locations round the globe, it was possible to get a vast amount done in a short time, at some cost to family life but with a minimum of misdirected effort.

At several points, potential funders have said "All you need to do is fill in our form..." It is always worth asking whether this is a poisoned chalice. How much effort will it take to fill in, and how much more to follow it through? What is the cost to my engagement and enthusiasm of doing things their way (as I shall have to, if I take the king's shilling, as good governance demands that procedures are followed, forms are filled, any changes to the original plan are justified and documented ...).

I should note that, possibly, my perspective here is atypical. As the co-ordinator, without

[1] SENSEVAL data sets and results are available at http://www.senseval.org

Table 1: SENSEVAL-2, resources, participation, results.

Language	Task[a]	Systems	Lemmas	Instances[b]	IAA[c]	Baseline[d]	Best score
Czech	AW	1	–[e]	277,986	–	–	94
Basque	LS	3	40	5,284	75	65	76
Dutch	AW	1	1,168	16,686	–	75	84
English	AW	21	1,082	2,473	75	57	69
English	LS	26	73	12,939	86	48/16[f]	64/40
Estonian	AW	2	4,608	11,504	72	85	67
Italian	LS	2	83	3,900	21	–	39
Japanese	LS	7	100	10,000	86	72	78
Japanese	TM	9	40	1,200	81	37	79
Korean	LS	2	11	1,733	–	71	74
Spanish	LS	12	39	6,705	64	48	65
Swedish	LS	8	40	10,241	95	–	70

[a] AW: all-words task, LS: lexical sample, TM: translation memory.

[b] Total instances annotated in both training and test corpora. In the default case, they were split 2:1 between training and test sets.

[c] Inter-annotator agreement is generally the average percentage of cases where two (or more) annotators agree, before adjudication. However there are various ways in which it can be calculated, so the figures in the table are not all directly comparable.

[d] Generally, choosing the corpus-attested most frequent sense, although this was not always possible or straightforward.

[e] A dash '–' indicates the data was unavailable.

[f] Supervised and unsupervised scores are separated by a slash.

a funder as taskmaster, I had a particularly free hand to ordain as I saw fit. This was most agreeable, but it is quite possible that others involved saw me as their (more or less reasonable, more or less benevolent) dictator and bureaucracy, and did not share the pleasures of autonomy that I experienced.

I am not sure that I advocate the no-bureaucracy approach: clearly, it depends on there being some slack somewhere which can be redirected. It is however a model well worth considering, if only because it is such fun working with other committed volunteers for no better reason than that you all want to reach the same goal.

2 The re-use trap

Consider the following position (Redux, 2001):

> As followers of the literature will have noted, great strides have been made in statistical parsing. In two decades, system performance figures have soared to over 90%. This is a magnificent tale. Parsing is cracked. An enormous debt is owed to the producers of the Penn Treebank. As anticipated by Don Walker, marked-up resources were what we needed. Once we had them, the algorithm boys could set to work, and whoomph!
>
> The benefits of concentrating on the one corpus have been enormous. The field has focused. It has been the microscope under which the true nature of language has become apparent. Like Mendel unpacking the secrets of all species' genetics through assiduous attention to sweet peas, and sweet peas alone, Charniak, Collins, and others have unpacked the secrets of grammatical structure through rigorous attention to the Wall Street Journal.
>
> We would now like to point out the unhelpfulness of comments appear-

> ing on the CORPORA mailing list, reporting low performance of various statistical POS-taggers when applied to text of different types to the training material, and also of a footnote to a recent ACL paper, according to which a leading Penn-Treebank-trained parser was applied to literary texts but then its performance "significantly degraded". These results have not, I am glad to say, entered beyond that footnote into the scientific literature. The authors should realise that it is *prima facie* invalid to apply a resource trained on one type of data, to another. Anyone wishing to use a statistical parser on a text type for which a manually-parsed training corpus does not exist, must first create the training corpus. If they are not willing to do that, they may as well accept that ten years of dazzling progress is of no use to them.
>
> ...
>
> So now, our proposal. We are encouraged to see the amount of work based on the Wall Street Journal which appears in ACL proceedings. However we remain concerned about the quantity of papers appearing there which fail to use a rigorous methodology, and fail to build on the progress outlined above. These papers tend to fall outside the domain which has become the testing ground for our understanding of the phenomenon of language, viz, the Wall Street Journal. Outside the Wall Street Journal, we are benighted. May I suggest that ACL adopt a policy of accepting only papers investigating the language of the Wall Street Journal.

A similar position was discussed in relation to SENSEVAL. There was a move to use, in part or in whole, the same sample of words (ca 40 items) for SENSEVAL-2 (English lexical sample task) as had been used in SENSEVAL-1. This would have promoted comparability of results across the two exercises. However, we were anxious about continuing to focus our efforts on just 40 of the 10,000 ambiguous words of the language, as it seemed plausible that some issues had simply not arisen in the first sample, and if we did not switch sample, there was no chance that they would ever be encountered.

All SENSEVAL resources are in the public domain and can be (and have been) used by researchers wanting to compare their system performance with performance figures as in SENSEVAL proceedings. Of course such comparison will never be fair, as systems competing under the examination conditions of the evaluation exercise were operating under time pressure, and did not always have time to correct even the most egregious of bugs. However it is hard to see how the evaluation series can keep the sheer range and variety of language use on the agenda if samples are reused.

3 Language flow and the web

> You cannot step twice into the same river, for other waters are constantly flowing on.
> *Heraclitus* (c. 535-c. 475 BC)

We are currently planning a SENSEVAL-3 task where the test data will be instances of words in web pages, as located by a search engine. Test data will be defined by URL, line number and byte offset. The goal is to explore what happens when laboratory conditions are changed for web conditions. It will support exploration of how supervised-training systems perform when test set and training set are no longer subsets of the same whole. Partipants will be expected to first retrieve the web page and then apply WSD to it. This will allow systems to use a wider context than is possible in the usual paradigm of short-context test instances. They could, for example, gather a corpus of the reference URL, plus any pages it links to, plus other pages close to it in its directory tree, in order to identify the domain of the instance. In general, it makes space for a range of techniques which the SENSEVAL paradigm to date has ruled out.

Clearly, web pages may change or die between selecting URLs for manual tagging at set-up time, and the evaluation period, resulting in wasted manual-tagging effort. We shall minimize the waste by, first, drawing up a candidate list of URL's, then, checking them to see whether they are still available and unchanged a month or so later. The fact that some web pages have died will not invalidate the exercise. It just means there will be fewer usable test instances than test-URLs distributed.

One hypothesis to be explored is that supervised-training systems are less resilient than other system-types, in the real world situation where the data to be disambiguated "in anger" may not match the text type of the training corpus. The relation between the performance of supervised-training systems in the laboratory and in the wild is to my mind one of the critical issues at the current point in time, given the ascendancy that the paradigm has achieved in CL.

It may also shed light on the relation between a linguistic/collocational view of word senses and one dominated by domain. Inevitably, for some words, there will be a poor match between the domains of training-corpus instances and the domains of web instances. While this might seem 'unfair' and a problem following from the biases of the web, it is a fact of linguistic life. The concept of an unbiased corpus has no theoretical credentials. The task will explore the implications of working with a corpus whose biases are unknown, and in any case forever changing.

The web also happens to be the corpus that many potential customers for WSD need to operate on, so the task will provide a picture of whether WSD technology is yet ready for these potential clients.

4 SENSEVAL and Machine Translation evaluation

As noted above, overall SENSEVAL design is taken from MUC. We have also followed MUC and TREC discussions of the hub-and-spokes model and the need to forever look towards updating the task, to guard against participants becoming expert at the task as defined but not at anything else.

WSD is not a task of interest in itself. One does WSD in order to improve performance on some other task. The critical end-to-end task, for WSD, is Machine Translation (Kilgarriff, 1997).

In SENSEVAL-2, for Japanese there was a translation memory task, which took the form of an MT evaluation (Kurohashi, 2001). In that experimental design, each system response potentially requires individual attention from a human assessor. As in assessing human or computer translation, one cannot specify a complete set of correct answers ahead of time, so one must be open to the possibility that the system response is correct but different from all the responses seen to date. Thus the exercise is potentially far more expensive than the MUC model. In the MUC model, human attention is required for each data instance. In this model, human attention is potentially required for each data-instance/system combination.

Another consequence is that there is no free-standing, system-independent gold standard corpus of correct answers. New or revised systems cannot simply test against a gold standard (unless they limit their range of possible answers to ones already encountered, which would introduce further biases).

So it is a more complex and costly form of evaluation. However it is also far more closely related to a real task. It is a direction that SENSEVAL needs to take.[2] The MUC-style fixed-sense-inventory should be seen as what was necessary to open the chapter on WSD evaluation: a graspable, manageable task when we had no experience of the difficulties we might encounter, which also provided researchers with some objective datasets for their development work. For the future the

[2] It is also the route we have taken in the WASPS project, which is geared towards WSD for MT (Koeling et al., 2003).

emphasis needs to be on assessments such as the Japanese one, related to real tasks.

5 Metric re-use: kappa

Consider the (fictional) game show "Couples". The idea is to establish which couples share the same world view to the greatest extent. Each member of the couple is put in a space where they cannot hear what the other is saying, and is then asked twenty multiple-choice questions like

What is the greatest UK pop group of the 1960s?

> *The Beatles/The Rolling Stones*

or

Which month is your oldest nephew/niece's birthday?

> *Jan/Feb/Mar/Apr/May/Jun/Jul /Aug/Sep/Oct/Nov/Dec /No-nephew-or-niece*

The couple that gives the same answer most often wins.

Different couples get different questions, sometimes with different numbers of multiple-choice options, and this introduces a risk of unfairness. If one couple gets all two-way choices, while another gets all 13-way choices, and both agree half the time, the 13-way couple have really done much better. Random guessing would have got (on average) a 50% score for the couple who got the two-way questions, whereas it would only have got a 1/13 or 7.7% score for the others.

One way to fix the problem is to give, for each question, not a full point but a score modified to allow for what random guessing would have given. This can be defined as

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)}$$

where $P(A)$ is the proportion of times they actually agree, and $P(E)$ is the proportion of times they would agree by chance.

This is called the Kappa statistic. It was developed within the discipline of Content Analysis, and introduced into the HLT world by Jean Carletta (Carletta, 1996).

Inter-Annotator Agreement

For HLT, the issue arises in manual tagging tasks, such as manually identifying the word class or word sense of a word in the text, or the discourse function of a clause. In each of these cases, there will be a fixed set of possible answers. Consider two exercises, one where a team of two human taggers tag a set of clauses for discourse function using a set of four possible functions, the other where another team of two uses a set of fifteen possible functions. If the first team gave the same answers 77% of the time, and the second gave the same answers 71% of the time, then, at a first pass, the first team had a higher agreement level. However they were using a smaller tagset, and we can use kappa to compensate for that. The kappa figure for the first team is

$$\frac{0.77 - 1/4}{1 - 1/4} = \frac{0.52}{0.75} = 0.69$$

and that for the second team is

$$\frac{0.71 - 1/15}{1 - 1/15} = \frac{0.64}{0.93} = 0.69$$

The inter-annotator agreement (IAA) can be presented as simple agreement figures of 77% and 71%, or as kappa values of 0.69 in both cases.

IAA matters to HLT evaluation because human tagging is what is needed to produced 'gold standard' datasets against which system performance can be judged. The simplest approach is for a person to mark up a text, and to evaluate the system against those taggings. But the person might make mistakes, and there may be problems of interpretation and judgement calls where a different human may well have given a different answer. So, for gold standard dataset development, each item to be tagged should be tagged by at least two people.

How confident can we be in the integrity of the gold standard? Do we really know that it is correct? A central consideration is IAA: if taggers agreed with each other nearly all the time, we can be confident that, firstly, the gold standard corpus is not full of errors, and secondly, that the system of categories, or tags, according to which the markup took place is adequate to the task. If the tags are not well-suited to the task and adequately defined, it will frequently be arbitrary which tag a tagger selects, and this will show up in low IAA.

Reservations

Carletta presented kappa as a better measure of IAA than uncorrected agreement. In the specific cases she describes, this is certainly valid.

Those cases are very specific. Kappa is relevant where the concern is that an IAA figure based on a small tagset is being compared with one based on a large tagset. Where that is the focus of the investigation, kappa is an appropriate statistic.

Where it is not, there are arguments for and against the use of kappa. In its favour is that it builds in compensation for distortions that might otherwise go unnoticed resulting from different tagset sizes.

Against is, principally, the argument that kappa figures are hard to interpret. A simple agreement figure is just that: it is clear what it means, and the critical question of whether, say, 90% agreement is 'good enough' is one for the reader to form their own judgment on. With a kappa figure of .85, the reader needs to, firstly, understand the mathematics of kappa, and secondly, bear in mind the various complexities of how kappa might have been calculated (see also below), before forming a judgment. To "help" the reader with this task, there are various discussions in the literature as to how different kappa figures are to be interpreted. Sadly, these are contradictory (and even if they weren't, it is the duty of any critical reader to form their own judgment on what is good enough.)

Complexities in the calculation

Above we present kappa in its simplest form. Naturally, when used in earnest additional issues arise. The observations below arose principally from the consideration of how we might use kappa in SENSEVAL. The task was to produce a gold standard corpus in which words were associated with their appropriate meanings, with the inventory of meanings taken from a dictionary.

Firstly, tagset size is assumed to be fixed. In the SENSEVAL context, there were three issues here.

1. There were two variants of the task: 'lexical sample' and 'all-words'. In the all-words variant, all content words in a text are tagged. Some will be highly polysemous, others not polysemous at all. It is not clear how to present kappa figures that are averages across datasets where the tagset size varies.

   In the lexical sample task, first, a sample of sentences containing a particular word is identified, and then, only the instances of that word are tagged, so the issue does not arise immediately. It does still arise if a kappa figure is to be computed which draws together data from more than one lexical-sample word.

2. In addition to the dictionary senses for the word, there were two tags, U for 'unassignable' and P for 'proper name', which were always available as options for the human taggers. If included, for purposes of calculating kappa, a word that only has two dictionary senses is classified as a four-way choice, which seems inappropriate, particularly as U and P tags were quite rare and absent entirely for some words.

3. There were a number of other 'marginal' senses which, if included in the tag count, extend it greatly (for some words). In the SENSEVAL-1, taggers largely worked within a given word class, so noun instances of *float* were treated separately from verb instances, but, in e.g., noun cases where none of the noun instances fitted, they were instructed to consider whether any of the verb senses were a good semantic match (even though they

evidently could not be a syntactic match). Also some words formed part of numerous multi-word units that were listed in the dictionary. Where a tagger found the lexical-sample word occurring within a listed multi-word unit, the instruction was to assign that as a sense.

One response to issues 2 and 3 is to use a more sophisticated model of random guessing, in which, rather than assuming all tags are equally likely for the random guesser, we use the relative frequencies of the different tags as the basis for a probability model . The method succeeds in giving less weight to marginal tags, at the cost of making the maths of the calulation more complex and the output kappa figures correspondingly harder to interpret.

Secondly, the SENSEVAL tagging scheme allowed human taggers to give multiple answers, and also allowed multiple answers in the tagging scheme.

Thirdly, in SENSEVAL the number of humans tagging an instance varied (according to whether or not the instance was problematic).

Fourthly, there is a distinction between two kinds of occasion on which two taggers give different tags. It may be a problematic case to tag, or it may be simple human error (such as a typo). Arguably, simple typos and similar are of no theoretical interest and should be corrected before considering IAA. A related point is the distinction between agreement levels (between individual taggers) and replicability (between teams of taggers). Where the concern is the integrity of a gold standard resource, replicability is the real matter of interest: would another team of taggers, using the same data, guidelines and methods, arrive at the same taggings? A tagging methodology which guards against simple errors, wayward individuals, and wayward interpretations will tend to produce replicable datasets.

All of these considerations can be addressed using a variant of kappa. My point is that kappa becomes harder and harder to interpret, as more and more assumptions and intricacies are built into its calculation.

Kappa has been widely embraced as an example of an aspect of evaluation technology that carries across different HLT evaluation tasks, giving a shimmer of statistical sophistication wherever it alights. My sense is that it is a bandwagon, which HLT researchers have felt they ought to jump on in order to display their scientific credentials and ability to use statistics, which, in many places where it has been used, has led to little but gratuitous obfuscation.

6 Conclusion

Clearly, we would like new HLT evaluation exercises to benefit from evaluation work already done. This paper explores several issues that have arisen from the SENSEVAL experience.

References

Jean Carletta. 1996. Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22(2):249–254.

Philip Edmonds and Adam Kilgarriff. 2002. Introduction to the special issue on evaluating word sense disambiguation systems. *Journal of Natural Language Engineering*, 8(4).

Adam Kilgarriff. 1997. What is word sense disambiguation good for? In *Proc. Natural Language Processing in the Pacific Rim (NLPRS '97)*, pages 209–214, Phuket, Thailand, December.

Rob Koeling, Roger Evans, Adam Kilgarriff, and David Tugwell. 2003. An evaluation pf a lexicographer's workbench: building lexicons for machine translation. In *EACL workshop on resources for Machine Translation*, Budapest.

Sadao Kurohashi. 2001. SENSEVAL-2 japanese translation task. In *Proc.* SENSEVAL-*2: Second International Workshop on Evaluating WSD Systems*, pages 37–40, Toulouse, July. ACL.

Swift Redux. 2001. A modest proposal. *ELSnews*, 10(2):7.

Living up to standards

Margaret King
TIM/ISSCO
ETI
University of Geneva
Margaret.King@issco.unige.ch

Abstract

This paper attacks one part of the question "Are evaluation methods, metrics and resources reusable" by arguing that a set of ISO standards developed for the evaluation of software in general are as applicable to natural language processing software as to any other. Main features of the ISO proposals are presented, and a number of applications where they have been applied are mentioned, although not discussed in any detail.

Acknowledgements

The work recorded here is far from being all my own. I would like first to record my thanks to Nigel Bevan, technical editor of the ISO standards discussed for much interesting and enlightening discussion. Then many thanks must go to all my colleagues in the EAGLES and ISLE projects, especially Sandra Manzi and Andrei Popescu-Belis. Finally, I must thank all those whose work on applying the standards reported here provoked reflection and helped to convince me of the value of the approach: Marc Blasband, Maria Canelli, Dominique Estival, Daniele Grasso, Véronique Sauron, Marianne Starlander and Nancy Underwood.

1 Introduction

This paper is constructed around a syllogism:

1. ISO standards 9126 and 14598 are applicable to the evaluation of any type of software
2. Natural language processing software is a type of software
3. ISO standards 9126 and 14598 are applicable to the evaluation of natural language processing software.

In support of the major premise, I shall set out some of the major features of the ISO standards in question. The minor premise needs no support: indeed, it is almost a tautology. The truth of the conclusion will logically depend therefore on whether I have managed to convince the reader of the truth of the major premise. There will be little explicit argument in this direction: simply setting out key features of the approach should suffice. I will try, however, to reinforce the conclusion by briefly reviewing a number of natural language processing applications where the ISO standards have been followed with encouraging results. My hope, of course, is to encourage readers to apply the standards themselves.

2 ISO standards work on software evaluation

ISO has been publishing standards on software evaluation since 1991. The bibliography gives a detailed picture of what standards have already been published and of what standards are in preparation. ISO/IEC 9126 was the first standard to appear. It has subsequently been modified, and in its new versions the original content of 1991 has been refined, modified and distributed over a series of separate but inter-related standards.

The keystone of ISO work is that the basis of an evaluation is an explicit and detailed statement of what is required of the object to be evaluated. This statement is formulated very early in the process of defining an evaluation and is called a "quality model". The process of evaluation involves defining how measurements can be applied to the object to be evaluated in order to discover how closely it meets the requirements set out in the quality model.

"The object to be evaluated" is a clumsy phrase. It has been used because, in the ISO picture, evaluation may take place at any point in the lifecycle of a software product, and may have as its object not only the final product but intermediate products, including specifications and code which has not yet been executed. It follows from this that a quality model may apply to a set of specifications just as much as to a piece of finished software. Indeed, one might envisage using quality models as a way of guiding the whole process of producing a software product, from initial research and prototyping through to delivering and field testing the final product. That this is in line with best practice in software engineering constitutes, to my mind, an argument in favour of the ISO proposals.

As well as a set of standards relating to the definition of quality models (the 9126 series) ISO also offers a set of standards relating to the process of evaluation (the 14598 series). One document sets out a standard for the evaluation process seen at its most generic level, further proposals relate definition of the process to the particular viewpoints of software developers, of acquirers of software and of evaluators typically working as third party evaluators. Other documents in the 14598 series provide supporting material for those involved in evaluation, offering standards for planning and management of evaluations and for documentation of evaluation modules. Of the 9126 series, only the first document which directly deals with quality models has as yet been published. Documents in preparation deal with standards for the metrics which form a critical accompaniment to any quality model. It would be unrealistic in the space of a single paper to discuss even the documents already published in any detail. In what follows, we concentrate on outlining the foundations of the ISO proposals, the quality model and the process of evaluation.

3 Quality models (ISO 9126)

A quality model consists of a set of quality characteristics, each of which is decomposed into a set of quality sub-characteristics. Metrics measure how an object to be evaluated performs with respect to the quality characteristics and sub-characteristics. The quality characteristics and sub-characteristics making up the quality model of ISO 9126-1/01 are shown in figure 1, on the next page. All that figure 1 shows are names: ISO 9126-1/01 gives both definitions and discussion.

The quality characteristics are intended to be applicable to any piece of software product or intermediate product. They are thus necessarily defined at a rather high level of generality, and need to be made more specific before they are applicable to any particular piece of software. They are also defined through natural language definitions, and are thus not formal in the mathematical or logical sense. This being so, they are open to interpretation. Defining a specific evaluation implies deciding on an appropriate interpretation for that evaluation.

ISO 9126/01, whilst not barring the possibility that a quality model other than that contained in the standard might be used, requires that if another model is used, it should be clearly described.

"*Software quality shall be evaluated using a defined quality model. A quality model shall be used when setting quality goals for software products and intermediate products. This part of ISO/IEC 9126 provides a recommended quality model which can be used as a checklist of issues relating to quality (although other ways of categorising quality may be more appropriate in particular circumstances). When a quality model other than that in this part of ISO/IEC 9126 is used it shall be clearly described.*" (ISO 9126/01, 1.5, Quality relationships).

Work within the EAGLES project on defining a general framework for evaluation design extended this model by allowing the quality sub-characteristics in their turn to be decomposed; the process of decomposition being repeated if necessary.

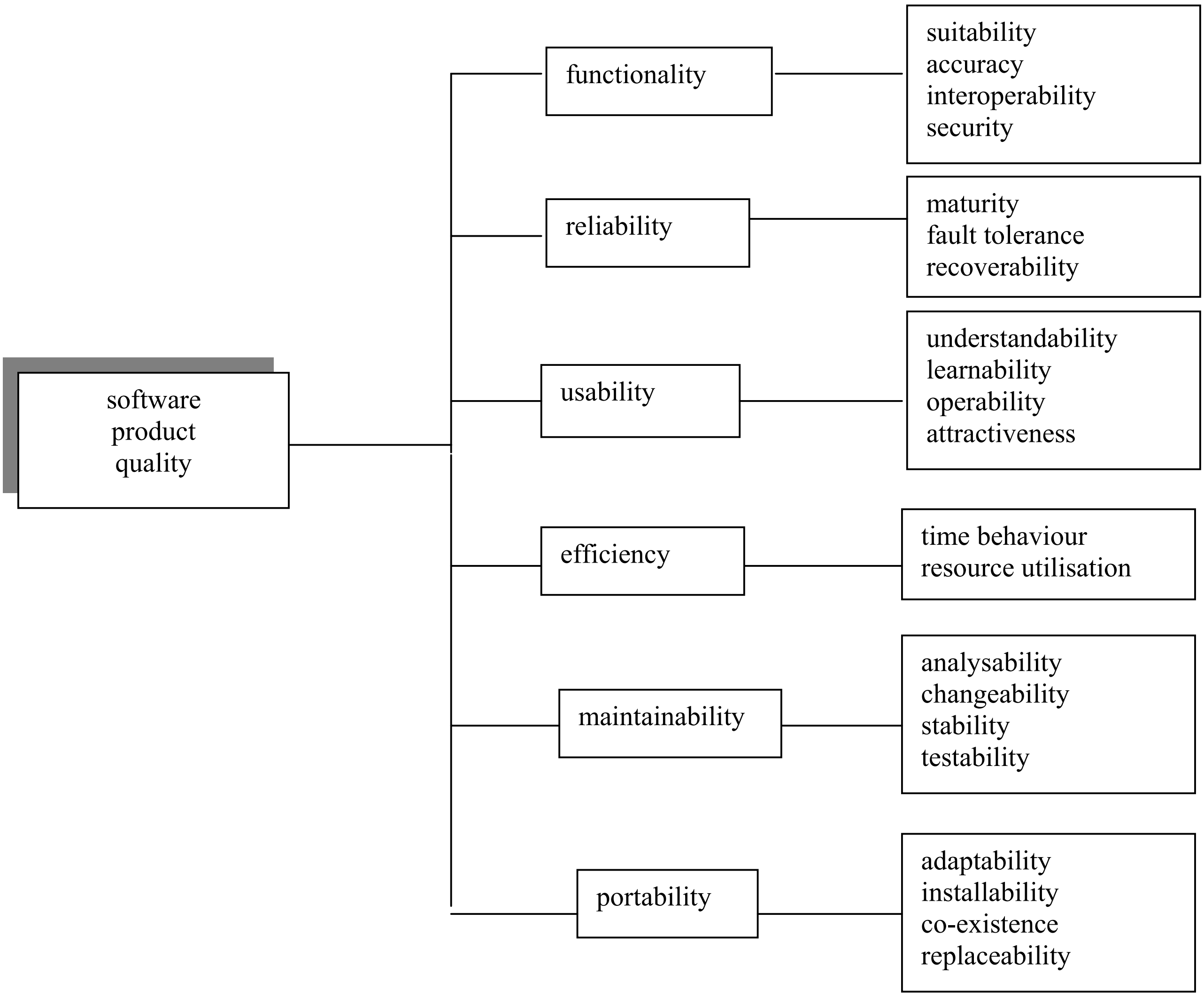


Figure 1

The structure thus obtained is hierarchical, and, theoretically of unlimited depth. ISO 9126-1/01 does not rigidly specify the relationship between quality characteristics and metrics. The EAGLES extension requires that each terminal node of the structure has at least one metric associated with it. The structure then becomes a hierarchy of attribute value pairs, where each node is labelled with the name of an attribute. The values of the attributes at the terminal nodes are directly obtained by the application of metrics. The value of a higher level node is obtained by combining the values of attributes nodes immediately dominated by the higher level node: values percolate upwards. Exactly how the combination of values is done is determined by a combining function which reflects the relative importance of the attributes in a particular evaluation. This formalization provides an operational semantics for any particular instantiation of the quality model. Once the evaluation designer has decided what attributes to include in his quality model and how to organise them, and once he has defined and assigned metrics to the terminal nodes, what functionality, for example, means within that quality model is defined by the decomposition of the functionality node and by the associated metrics.

Metrics will be discussed only briefly here. The ISO standard distinguishes between internal metrics, external metrics and quality in use metrics. The difference between them is determined by what kind of an evaluation object they are applied to.

Internal metrics apply to static properties of software, that is software considered independently of its execution. Examples might be the number of lines of code or the programming language used. As can be seen from the inclusion of the programming language in this list, metrics are not necessarily quantitative in their nature, although they should, of course, be as objective as possible. (This is one of the points we shall not go into further here.)

External metrics apply to software when it is being executed, to the behaviour of the system as seen from outside. Thus they may measure the accuracy of the results, the response time of the software, the learnability of the user interface and a host of other attributes that go to make up the quality of the software as a piece of software.

Quality in use metrics apply when the software is being used to accomplish a particular task in a particular environment. They are more concerned with the effects of using the software than with the software itself. Quality in use metrics are therefore very dependent on a particular environment and a particular task. Quality in use is itself a super-ordinate aspect of quality, for these same reasons. It is clearly influenced by the quality characteristics which make up the quality model, but is determined by the interaction of different quality characteristics in a particular task environment.

The ISO standards published so far say little about what makes a metric a good metric. Some work elsewhere (Popescu-Belis, 1999, Hovy et al, 2003) has made some suggestions.

First, metrics should be coherent, in the sense that they should respect the following criteria:

- A metric should reach its highest value for perfect quality (with respect to the attribute being measured), and, reciprocally, only reach its highest level when quality is perfect.
- A metric should reach its lowest level only for the worst possible quality (again, with respect to the attribute being tested)
- A metric should be monotonic: that is, if the quality of software A is higher than that of software B, then the score of A should be higher than the score of B.

We might compare two metrics (or more strictly two rating functions: see the section on process below) by saying that a metric m_1 is more severe than a metric m_2 if it yields lower scores than m_2 for every possible quality level. Conversely, one metric may be more lenient than another.

To these rather formal considerations, we might add:

- A metric must be clear and intuitive
- It must correlate well with human judgements under all conditions
- It must measure what it is supposed to measure
- It must be reliable, exhibiting as little variance as possible across evaluators or for equivalent inputs
- It must be cheap to prepare and to apply
- It should be automated if possible

4 Evaluation process (ISO 14598)

A first section of ISO 14598-1/99 is concerned with an overview of how all the different 9126 and 14596 documents concerned with software evaluation fit together. This overview can be summarized quite briefly. It is fundamental to the preparation of any evaluation that a quality model reflecting the user’s requirements of the object to be evaluated be constructed. The 9126 series of documents is intended to support construction of the quality model.

The 14598 series is concerned with the process of evaluation, seen from different viewpoints. Separate documents in the series tackle evaluation from the point of view of developers, acquirers and (third party) evaluators. All of these make use of the 9126 series, and are further supported by the second half of 14598-1, which sets out a generic picture of the process of evaluation, and by two further documents, the first concerned with planning and management of a software evaluation process, the second with guidance for documenting evaluation modules.

Although these other documents in the series are clearly important, we limit ourselves here to summarizing the process of evaluation, as set out in ISO 14598-1.

The evaluation process is conceived as being generic: it applies to component evaluation as well as to system evaluation, and may be applied at any appropriate phase of the product life cycle.

The evaluation process is broken down into four main stages, each of which is considered separately below:

Stage I: Establish evaluation requirements.

This step is broken down into a further three steps:

a) Establish the purpose of the evaluation

The commentary on this point reveals just how wide the scope of the standard is intended to be. The purpose of evaluating the quality of an intermediate product may be to:

- Decide on the acceptance of an intermediate product from a sub-contractor
- Decide on the completion of a process and when to send products to the next process
- Predict or estimate end product quality
- Collect information on intermediate products in order to control and manage the process

(The reader will remember that intermediate product means, for example, specifications or code before it is executed).

The purpose of evaluating an end product may be to:

- Decide on the acceptance of the product
- Decide when to release the product
- Compare the product with competitive products
- Select a product from among alternative products
- Assess both positive and negative effects of a product when it is used
- Decide when to enhance or replace the product.

It follows from this very broad range of possibilities that the standard is meant to apply not only to any kind of intermediate or final software product, but to any evaluation scenario, including comparative evaluation.

b) Identify types of products to be evaluated

Types of products here does not mean application software, but rather is concerned with the stage reached in the product's life cycle, which determines whether and what intermediate product or final product is to be evaluated.

c) Specify quality model

The quality model is, of course, to be defined using ISO 9126-1/01 as a guide. However, a note quoted again below adds:

"The actual characteristics and sub-characteristics which are relevant in any particular situation will depend on the purpose of the evaluation and should be identified by a quality requirements study. The ISO/IEC 9126-1 characteristics and sub-characteristics provide a useful checklist of issues related to quality, but other ways of categorising quality may be more appropriate in particular circumstances." (ISO 14598-1/99)

An important word here is "checklist": the basic purpose of the ISO quality model is to serve as a guide and as a reminder for what should be included in evaluating software. Arguing about the exact interpretation of the quality characteristics is pointless. Their interpretation is given by the model in which they are incorporated.

Stage II:Specify the evaluation

This too breaks down into three steps:

a) **Select metrics**

b) **Establish rating levels for metrics**

c) **Establish criteria for assessment**

Quality characteristics and sub-characteristics cannot be directly measured. Metrics must therefore be defined which correlate to the quality characteristic. Different metrics may be used in different environments and at different stages of a product's development. Metrics have already been discussed to some extent in the section on quality models above.

A metric typically involves producing a score on some scale, reflecting the particular system's performance with respect to the quality characteristic in question. This score, uninterpreted, says nothing about whether the system performs satisfactorily. To illustrate this idea, consider the Geneva education system, where marks in examinations range from 1 to 6. How do you know, without being told, that 6 is the best mark and 1 the worst? In fact, most people guess that it is so: they may then have a difficult time in Zurich where 1 is the highest mark. Establishing rating levels for metrics involves determining the correspondence between the uninterpreted score and the degree of satisfaction of the requirements. Since quality refers to given needs, there can be no general rules for when a score is satisfactory. This must be determined for each specific evaluation.

Each measure contributes to the overall judgement of the product, but not necessarily in a uniform way. It may be, for example, that one requirement is critical, whilst another is desirable, but not strictly necessary. In this case, if a system performs badly with respect to the critical characteristic, it will be assessed negatively no matter what happens to all the other characteristics. If it performs badly with respect to the desirable but not necessary characteristic, it is its performance with respect to all the other characteristics which will determine whether the system is acceptable or not.

This consideration feeds directly into the third step, establishing criteria for assessment, which involves defining a procedure for summarizing the results of the evaluation of the different characteristics, using for example decision tables or weighting functions of different kinds.

Stage III: Design the evaluation

Designing the evaluation involves producing an evaluation plan, which describes the evaluation methods and the schedule of the evaluator action. The other documents in the 14598 series expand on this point, and the plan should be consistent with a measurement plan, as described and discussed in the document on planning and management. (ISO 14598-2/00)

Stage IV: Execute the evaluation

This final stage again breaks down into three stages:

a) **Measurement**

b) **Rating**

c) **Assessment**

These steps are intuitively straightforward in the light of the discussion above. Measurement gives a score on a scale appropriate to the metric being used. Rating determines the correlation between the raw score and the rating levels, in other words, tells us whether the score can be considered to be satisfactory. Assessment is a summary of the set of rated levels and can be seen as a way of putting together the individual ratings to give an overall picture which also reflects the relative importance of different characteristics in the light of the particular quality requirements. Final decisions are taken on the basis of the assessment.

5 ISO, EAGLES and natural language applications in practice.

It would be impossible of course to claim knowledge of all applications of the ISO standards, even within the limited area of work on natural language. In this concluding section only those applications that came to the author's cognisance through her involvement with work in the EAGLES, ISLE and Parmenides projects are mentioned.

The ISO model of 9126/91 as extended and formalized by the first EAGLES project has been tested by application to a number of different language engineering applications. Within the TEMAA project it was applied to the evaluation of spelling checkers, and initial work was done on quality models for grammar checkers and translation memory systems. As part of the EAGLES project itself, a number of projects in the general field of information retrieval were asked to apply the framework, and produced, in those cases where the project included a substantial evaluation component, encouraging results. The second EAGLES project was, for the evaluation group, essentially a consolidation and dissemination project, where an attempt was made to encourage use of earlier results. During this time, the model was also applied in the context of the ARISE project, which developed a prototype system whereby information on railway timetables could be obtained through spoken dialogue. Similarly, an Australian manufacturer of speech software used the framework to evaluate a spoken language dialogue system. Case studies undertaken in the context of post-graduate work have applied the ISO/EAGLES methodology to the evaluation of dictation systems, grammar checkers and terminology extraction tools. One part of the ISLE project, now coming to an end, has been applying the methodology to the construction of a large scale quality model of machine translation systems. Many of the results of this work can be consulted by looking at the EAGLES and ISLE web sites.

Recently, work has begun on the Parmenides project. This project is concerned with ontology based semantic mining of information from web based documents, with a special interest in keeping track of information which changes over time. Evaluation plays an important role in the project. Three separate user groups are supplying the basis for case studies. At the time of writing, user requirements are being defined, which will be translated into quality requirements for the software to be developed within the project and which will serve as the basis for the quality models to be used in on-going and final evaluation.

6 Conclusion.

The workshop for which this paper has been written addresses the question of whether there is anything that can be shared between evaluations. The answer which I hope to have made convincing is that one thing which can be shared is a way of thinking about how evaluations should be designed and carried out. Adhering to an acknowledged standard in the construction of quality models and in developing the process of a specific evaluation can only make it easier to share more detailed aspects of evaluation and provides a common framework for discussion of such issues as metrics and their validity.

References.

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

EAGLES Evaluation Working Group. 1996. *EAGLES Evaluation of Natural Lnaguage Processing* Systems. Final Report, Center for Sprogteknologi, Copenhagen, Denmark.

Hovy, E, King, M and Popescu-Belis, A. 2002. *Computer-aided Specification of Quality Models for MT Evaluation.* Third International Conference on Language Resources and Evaluation (LREC).

Hovy, E, King, M and Popescu-Belis, A. 2003. *Principles of Context Based Machine Translation Evaluation.* ISLE report.

ISO/IEC 9126-1:2001 *Software engineering – product quality – Part 1: Quality Model.* Geneva, International Organization for Standardization and International Electrotechnical Commission.

ISO/IEC DTR 9126-2 (in preparation): *Software engineering – product quality – Part 2: External metrics.* . Geneva, International Organization for Standardization and International Electrotechnical Commission

ISO/IEC CD TR 9126-3 (in preparation): *Software engineering – product quality – Part 3: Internal metrics.* . Geneva, International Organization for

Standardization and International Electrotechnical Commission

ISO/IEC CD 9126-4 (in preparation): *Software engineering – product quality – Part 4: Quality in use metrics*. . Geneva, International Organization for Standardization and International Electrotechnical Commission

ISO/IEC CD 9126-30 (in preparation): *Software engineering – Software product quality requirements and evaluation – Part 30: Quality metrics* – Metrics reference model and guide. . Geneva, International Organization for Standardization and International Electrotechnical Commission

ISO/IEC 14598-1:1999 *Information technology – Software product evaluation – Part 1: General Overview*. Geneva, International Organization for Standardization and International Electrotechnical Commission

ISO/IEC 14598-2:2000– *Software engineering - product evaluation – Part 2: Planning and Management.* Geneva, International Organization for Standardization and International Electrotechnical Commission

ISO/IEC 14598-3:2000– *Software engineering - product evaluation – Part 3: Process for developers*. . Geneva, International Organization for Standardization and International Electrotechnical Commission

ISO/IEC 14598-5:1998 *Information technology – Software product evaluation – Part 5: Process for evaluators* Geneva, International Organization for Standardization and International Electrotechnical Commission

ISO/IEC 14598-4:1999– *Software engineering - product evaluation – Part 4: Process for acquirers* Geneva, International Organization for Standardization and International Electrotechnical Commission

ISO/IEC 14598-6:2001– *Software engineering - product evaluation – Part 6: Documentation of evaluation modules* Geneva, International Organization for Standardization and International Electrotechnical Commission

King, M. 1996. *Evaluating Natural Language Processing Systems*. Communications of the Association for Computing Machinery (CACM), Vol. 39, Number 1.

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

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

Setting up an Evaluation Infrastructure for Human Language Technologies in Europe

Kevin McTait & Khalid Choukri
ELDA
55 – 57 rue Brillat-Savarin
75013 Paris, France
{mctait,choukri}@elda.fr

Abstract

This paper describes ELRA/ELDA's vision of an evaluation infrastructure for Human Language Technologies in Europe. Drawing on its experience in national and Europe-wide evaluation projects and also its experience in the production, validation, packaging and distribution of language resources, such as electronic text corpora, lexica and speech databases, ELDA's evaluation department seeks to set up a European clearing house for evaluation related resources and software packages, in the same way that ELDA has become the European clearing house for language resources. ELDA's vision for a European evaluation infrastructure is inspired by both European and international evaluation initiatives, including the DARPA/NIST evaluation programme in the United States.

1 Introduction

In 1995, the European Language Resources Association (ELRA) was set up under the auspices of the European Commission as a non-profit making body with the aim of making language resources (LR) available to the language engineering community. Such resources are essential to both public research institutions and private companies wishing to construct, develop and test Human Language Technology (HLT) systems, such as speech recognisers, machine translation systems, terminology support tools etc. ELRA's operational body, the Evaluation and Language resources Distribution Agency (ELDA) was set up to act as the European clearing house for such LRs. ELDA is active in the specification, production, validation, packaging and distribution of LRs and also deals with the legal issues involved. Today, its catalogue contains several different types of LR, such as speech databases, electronic lexica and text corpora (monolingual, parallel multilingual, multimodal etc) in several different languages. ELDA's clients include not only large commercial organisations, but also public sector research laboratories and universities.

ELDA's evaluation department is active in evaluation projects on both the national (French) and European levels. For example, the *Technolangue* programme, funded by the French Ministries of Research, Industry and Culture, contains several projects dedicated to the advancement of HLT in France. One project under the *Technolangue* programme is entitled EVALDA, and is dedicated to setting up permanent and lasting evaluation protocols and packages for the major linguistic technologies, namely:

- Corpus Alignment
- Terminology
- Machine Translation
- Syntactic Parsers

- Q/A Systems
- Broadcast News Transcription Systems
- Speech Synthesis
- Dialogue Systems

In the context of the EVALDA project, players from academia, public sector research and the private sector are invited to take part in competitive and comparative evaluation campaigns, culminating in a workshop. A scientific committee was set up for each of the 8 linguistic technologies shown above, in order to discuss, define and agree on evaluation protocols including evaluation methodologies (whether automatic, assessed by human evaluators or both), metrics, evaluations tasks, resources and evaluation software.

In order that the evaluation campaigns are ethical and valid, an independent organisation with the necessary skills is required to oversee and manage the evaluation campaigns. In this case, ELDA is well placed to take on this role.

In addition to the EVALDA project, ELDA is involved in the TC-STAR_P project (Preparatory Action for the project Text and Corpora for Speech to Speech Translation). The purpose of this preparatory action is to write a proposal to the EU commission, under the 6th Framework, requesting funding for a 5 year project entitled TC-STAR.

In this project/proposal, ELDA undertakes all issues relating to LRs for Speech-to-Speech (SST) components i.e. speech recognition, speech centred translation and text-to-speech, and evaluation (of the SST system as a whole and the individual components). Therefore, ELDA undertakes the collection, specification, production and distribution not only of the LRs required by the research and development teams, but also undertakes to commission, produce and distribute resources for evaluation, including the software packages necessary.

ELDA is also involved in the CLEF project, a French national initiative whose purpose is to develop and maintain an infrastructure for the evaluation of cross-language information retrieval systems (CLIR). In this project, ELDA is responsible for data acquisition and negotiation of rights. With respect to information retrieval in French, ELDA was also involved in defining evaluation procedures in the French AMARYLLIS project.

Again, as an independent HLT organisation, ELDA is well placed to manage the evaluation campaigns. In fact, the networks for LRs and LR expertise set up by ELRA/ELDA prove to be an invaluable source of expertise in such projects.

2 Evaluation

2.1 Why Evaluate?

Evaluation forms a fundamental part of the development of language engineering products. It is essential for validating research hypotheses, for assessing progress and for choosing between research alternatives.

In more detail, it enables R&D teams to assess the impact of innovations on system performance. For example, does changing parameter *x* entail an increase in system performance validating the change?

Evaluation also identifies promising technology or research directions enabling industry to assess its market value. However, language engineering displays a paradoxical property in that in many areas, the state of the technology has reached a level barely sufficient to be usable in practice. Nevertheless, many commercial language-based applications do exist (e.g. machine translation, text summarisation, dictation, spoken dialogue systems). Comparative evaluation could help clear up the issues, where the advertised performance claims are difficult to assess and compare objectively.

Evaluation also allows funding agencies to determine whether their investment has led to significant progress. Many national, European or international projects require progress reports every *x* months. Therefore, the results of evaluation campaigns enable the progress of the project to be tracked. It also gives funding agencies the data necessary to quantitatively evaluate the progress made possible by their investment, and thus suggests priorities on where to plan research efforts and support for application development. Evaluation campaigns also provides useful input when deciding whether a technology is mature enough to be considered as a candidate for starting commercial application development.

A further side effect of evaluation campaigns is the production of high quality evaluation resources, in the form of training and test data along with evaluation software packages, distributed or produced during evaluation campaigns. Also, the availability of evaluation packages enables *all* researchers in a particular field to evaluate, benchmark and compare the performance of their systems.

2.2 Evaluation in the US and Europe

In the USA, the DARPA government funding agency is active in the evaluation of the principal areas of HLT: speech dictation, spoken language understanding, broadcast news transcription, named entities extraction, topic detection and tracking, text retrieval, message understanding, machine translation, speaker verification, character recognition, etc. It organises competitive evaluation campaigns and publishes the results in a workshop. The tasks within the different language technologies have been made more and more difficult, in agreement with the improvement in the various technologies over time. In order to have the necessary logistics for such evaluations, two entities play a major role in this framework: NIST, the National Institute for Standards and Technology, and the LDC, the Linguistic Data Consortium, which was created for the purpose of distributing language resources.

It would appear that the US-based evaluation programmes follow a top-down strategy i.e. the US government strongly influences the campaigns, but provides abundant funding and a long-lasting infrastructure. In Europe, the strategy has been rather more bottom-up, starting from individual research groups and HLT systems.

The US campaigns have inspired efforts at creating a lasting and permanent evaluation infrastructure in Europe. However, the picture in Europe is more fragmented for several reasons. First, there have been much less resources devoted to evaluation and secondly, evaluation efforts have come from many different sources, the result of which is that there is no equivalent European evaluation infrastructure. However, there have been several initiatives, either at the EU level (CLEF, SQALE, TSNLP, the proposed EAGLES evaluation methodology, ETSI/Aurora, DiET, DISC, TEMAA, and SPARKLE etc.), or on a national level (Grace, Aupelf ARC, in France, Verbmobil and the Morpholympics in Germany and SENSEVAL/ROMANSEVAL cosponsored by several EU-projects, ELSNET, ELRA and the British government). But all these initiatives were funded within limited duration projects, and there is no permanent entity designed to organise evaluation campaigns and capitalise on the resources and packages created during these independent initiatives. Therefore, the result is that European research teams are obliged to evaluate their technologies in US evaluation campaigns, using US evaluation packages which are subject to the geo-political incentives of the US research funding bodies.

However, inspired by the DARPA evaluation framework, the EU funded ELSE project was set up in order to draw up an executive summary for a general infrastructure for the evaluation of HLT systems in Europe. The idea was to focus on comparative as well as competitive evaluation techniques, taking into account the special situation in Europe i.e. multiple languages, a union of nations, industrial and commercial relevance, general EU programme policy etc.

From the analysis conducted within ELSE, comparative technology evaluation (in conjunction with DARPA style competitive evaluation) brings many interesting features. It forces researchers and technology developers to go deeper in their research field when they try to figure out how to measure the performance of a system for a given task. It gives technology developers objective information in order to make choices in system development. It gives industry the possibility of comparing their technology with others by participating in evaluation campaigns, or by acquiring the test data and comparing their systems performance with what has been achieved and reported so far. In particular, it provides SMEs with an efficient and easy market watch.

ELSE has provided recommendations for setting up such an evaluation infrastructure in Europe. It has identified the advantages of using the comparative evaluation paradigm and has listed several language technologies which could immediately make use of the evaluation infrastructure based on their relevance for research and industry.

The ELSE project report proposed two possible schemes for implementing this evaluation infrastructure. The first is a proactive approach, responding to the needs of individual research groups or technologies and the second being reactive, responding to FP calls for proposals. The report recommends that both be followed in parallel, in a bootstrapping manner.

Finally, the ELSE project investigated the relationship between technology evaluation and usage evaluations, requiring best practice guidelines and handbooks. Also recommended is basic research in evaluation to be considered over a longer timescale in order to constantly improve knowledge about evaluation metrology.

3 A European Evaluation Infrastructure

ELDA has a proven track record in the efficient and cost-effective distribution of LRs on both a European and worldwide level. It has set up an *organisational model* for LR networks dedicated to the specification, commissioning, production, validation, packaging and distribution of LRs with the legal issues resolved.

Along with its experience in national and European evaluation projects, ELDA's evaluation department capitalises on this experience to create an *organisational model* for efficient and cost-effective evaluation management. This entails the creation of a European, even international, network or infrastructure of evaluation centres providing evaluation resources, software packages, technology, forums of scientific expertise and R&D centres for the independent, ethical evaluation of human language technologies.

A starting point for an evaluation infrastructure in Europe is the European ELSE project, whose aim as to draw up a blueprint for a comparative, as opposed to competitive, evaluation infrastructure for the major linguistic technologies. ELSE provides recommendations for the establishment of such an infrastructure. ELDA's vision for a European infrastructure is also inspired by the evaluation activities organised by the DARPA/NIST institutions in the US.

The European infrastructure, as the ELSE project, would be organised along two major principles, proactive and reactive evaluation schemes. ELDA's evaluation department is currently taking part in reactive evaluation in that it has answered calls for proposals for national and European projects, such as EVALDA, TC-STAR_P, CLEF and AMARYLLIS to be involved in the specification and production of evaluation resources, packages and protocols. An exit strategy is defined for each project where the evaluation resources, packages, software and knowledge (final project reports) produced in each evaluation campaign for each linguistic technology is made available to external players through ELDA's catalogue for a modest price. ELDA is well placed to carry out this mission due to its significant experience in the specification, production, packing and distribution of LRs – a related task.

In parallel, the evaluation infrastructure would be proactive. ELDA endeavours to make available evaluation resources and packages for all linguistic technologies in as many languages as possible. At the very least, a European evaluation infrastructure would have to make available evaluation resources and packages for the official EU languages.

A European initiative is required due to the international and multilingual nature of linguistic technologies. All major developers work on several languages even if they do not create truly multilingual systems. Furthermore, the major players operate on an international level. International cooperation has also been the key to the success of many projects or systems. Therefore, porting linguistic technologies across more and more language barriers leads to a greater need for a multilingual evaluation framework.

Finally, many European language markets are too small to sustain their own evaluation programmes. For example, a language with relatively few speakers i.e. Dutch or Danish, can only rely on European cooperation to organise the evaluation campaign that they need. With the arrival of the new member states in 2004, ELDA faces the challenge of providing evaluation resources and packages for these new languages and therefore seeks cooperation with the new national agencies, research centres and private concerns to make available, commission and produce language and evaluation resources in the new languages.

It would not have to stop there. ELDA's long term goal in this respect is to cover as many world languages and human language technologies as possible, therefore creating an international evaluation infrastructure, dealing not only with European languages, but languages such as Chinese, Japanese etc.

In either case, the evaluation packages, in the form of training data, test data, test suites, evaluation protocols, software packages, toolkits, agreed methodologies, metrics and even *savoir-faire*, created through evaluation campaigns or by commissioning in a proactive manner, would be made available to the wider research community via ELDA's catalogue in the same way that ELDA makes LRs available. In this way, ELDA can take on the role of European clearing house or centre for evaluation technology, resources and expertise.

It is envisaged that a research or development team wishing to evaluation their system, whether for assessing development progress, focussing research efforts, providing feedback to a funding body or higher management etc., would contact ELDA's evaluation department and be supplied with the relevant evaluation packages. In the case of open source software or resources, the packages could simply be downloaded from ELDA's website.

Using its experience in the legal aspects of LR distribution, the legal issues pertaining to evaluation resources and packages would also be resolved by ELDA.

As the centre of a European evaluation infrastructure, ELDA would also become the forum or focus of knowledge on evaluation issues and evaluation metrology. In the course of evaluation campaigns and the commissioning of evaluation packages, ELDA will have acquired a good deal of expertise in evaluation over the entire range of linguistic technologies. In so doing, ELDA would become a centre of knowledge on evaluation in HLT and would be well placed to disseminate this knowledge.

In commissioning evaluation packages, agreement will have to be reached, in conjunction with other research groups, on evaluation protocols, methodologies and measures (as was the goal of the EAGLES project). Therefore, ELDA seeks to standardise evaluation protocols and make these standards available, along with the scientific justification behind it. Furthermore, using its expertise in evaluation, ELDA seeks to advance basic research in the subject of evaluation. In so doing, ELDA would be advancing the field of metrology in language engineering evaluation.

References

Adda, Gilles, Josette Lecomte, Joseph Mariani, P. Paroubek, M. Rajman. 1998. *The GRACE French Part-of-Speech Tagging Evaluation Task* in Proceedings of the First International Conference on Language Resources and Evaluation (LREC), Granada, May 1998.

Harman, Donna. 1998. *The Text REtrieval Conference (TRECs) and the Cross- Language Track*, in Proceedings of the First International Conference on Language Resources and Evaluation (LREC), Granada, May 1998

Kilgarriff, Adam. 1998. *SENSEVAL: An Exercise in Evaluating Word Sense Disambiguation Programs*, in Proceedings of the First International Conference on Language Resources and Evaluation (LREC), Granada, May 1998

Mariani, Joseph. 1998. *The Aupelf-Uref Evaluation-Based Language Engineering Actions and Related Projects*, in Proceedings of the First International Conference on Language Resources and Evaluation (LREC), Granada, May 1998

Peters, Carol, Martin Braschler, Julio Gonzalo and Michael Kluck (Eds). 2001. *Evaluation of Cross-Language Information Retrieval Systems*. Second Workshop of the Cross-Language Evaluation Forum, CLEF 2001, Darmstadt, Germany, September 2001 (Revised Papers).

Walker, M., D. Litman, C. Kamm, A. Abella. 1997. *PARADISE: A Framework for Evaluating Spoken Dialogue Agents*, in Proceedings of the 35th Annual Meeting of the Association of Computational Linguistics, ACL 97, 1997

Young, S.J., M. Adda-Decker, X. Aubert, C. Dugast, J.L. Gauvain, D.J. Kershaw, L. Lamel, D.A. Leeuwen, D. Pye, H.J.M. Steeneken A.J. Robinson, and P.C. Woodland. 1997. *Multilingual large vocabulary speech recognition: the european SQALE project*. Computer Speech and Language, 11(1):73-89.

http://www.limsi.fr/TLP/ELSE/

http://www.nist.gov/

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

http://www.itri.brighton.ac.uk/events/senseval/

Author index

Bontcheva, Kalina 3
Casacuberta, Francisco 27
Choukri, Khalid 73
Dalli, Angelo 51
Devillers, Laurence 11
Harman, Donna 1
Hutchinson, Ben 43
Kilgarriff, Adam 57
King, Gary 19
King, Margaret 65
Lowe, Will 19
McTait, Kevin 73
Mas, Josep Àngel 27
Maynard, Hélène 11
Mollá, Diego 43
Paroubek, Patrick 11
Pastra, Katerina 35
Rosset, Sophie 11
Saggion, Horacio 35
Tomás, Jesús 27