The Workshop Programme

14.00 - 14.45	"Users and user needs: problems in the evaluation of knowledge discovery systems". Maghi King (ISSCO/ETI, Geneva, Switzerland)
14.45 - 15.30	"Issues in Evaluating a Question Answering System" Gerhard Fliedner (DFKI, Saarbrücken, Germany)
15.30 - 16.00	Coffee Break
16.00 - 16.45	"User-Oriented Evaluation in Information Extraction" Roman Yangarber (New York University)
16.45 - 17.30	"Position Paper: Evaluating the Results of Unsupervised Ontology Learning" Myra Spiliopoulou and Roland Műller (University of Magdeburg, Germany)
17.30 - 18.00	Coffee Break
18.00 - 18.45	"How to get the right systems to the right users" Nancy Underwood (ISSCO/ETI, Geneva, Switzerland)
18.45 - 19.30	"Towards Best Practice Standards for Enhanced Knowledge Discovery Systems" Roland Stuckardt (Johann Wolfgang Goethe University, Frankfurt, Germany)
19.30 - 20.00	"The Way Forward" (general discussion)

Workshop Organisers

Maghi King, ISSCO/ETI, University of Geneva
Nancy Underwood, ISSCO/ETI, University of Geneva

Workshop Programme Committee

Hilbert Bruins Slot, Knowledge & Information Science, Unilever Nederland BV
Michael Hess, Institute of Computational Linguistics,University of Zurich
Maghi King, ISSCO/ETI, University of Geneva
Agnes Lisowska, ISSCO/ETI, University of Geneva
Fabio Rinaldi, Institute of Computational Linguistics,University of Zurich
Myra Spiliopoulou, Faculty of Computer Science University of Magdeburg
Nancy Underwood, ISSCO/ETI, University of Geneva

Table of Contents

Author Index

Foreword

The workshop to which these papers are preliminary contributions is intended to be a brain storming session, whose primary purpose is to launch discussion of what we believe to be a relatively new area of endeavour. Consequently, we have not asked for formal papers of the sort usually to be found in proceedings. We hope that the reader will bear this in mind, and that he will find himself provoked by some of the points raised, to the point where he too wants to contribute to the nascent discussion.

Users and user needs: problems in the evaluation of knowledge discovery systems

Margaret King
Margaret.King@issco.unige.ch
ISSCO / TIM, Ecole de traduction et d'interprétation
Université de Genève
Boulevard du Pont-d'Arve
CH-1211 Genève 4
http://www.issco.unige.ch

Abstract

This paper is intended to serve as an introductory paper to a workshop on user-oriented evaluation of knowledge discovery systems. Its main purpose is to raise issues and ask questions. The background to the paper is previous work on evaluation in other areas, such as machine translation, much of it closely related to and inspired by the two ISO standards on evaluation of software in general, the 9126 series and the 14598 series.

1. INTRODUCTION

This paper is intended to set the stage for a workshop discussion on user-oriented evaluation of knowledge discovery systems. Complex knowledge discovery systems are relatively new. There is considerable and interesting work on how to evaluate the results of the systems themselves, considered in isolation from application of those results to real world problems. A good example of this kind of evaluation work comes from the evaluation of systems which carry out clustering, where there is work on metrics to measure how good the proposed clusters are. But there is little work on evaluation geared towards discovering whether a system satisfies a user's needs, and not much in the way of shared experience in using such systems to furnish informal guidance.

However, there has been considerable work on user-oriented evaluation in other domains, most of it inspired by the ISO 9126 standard. The main thrust of this paper then is to ask whether or not that previous work can be useful to the task in hand.

2. What users want

At the most general level, the primary requirement of a user of a knowledge discovery system is fairly easy to state. The user is faced with a mass of data or of text. He is convinced, or at least hopes, that somewhere buried in this mass is information that would be useful to him if only it could be discovered and made explicitly available. He recognizes however that it would be totally unrealistic to attack the problem with humans. No matter how intelligent, well-informed and energetic they might be, the mass is too great for them to master its content and analyze it to the extent of being able to spot connections between disparate elements scattered piecemeal and bring them together into a previously unsuspected coherent story: human life is not long enough and human brain capacity is not large enough.

The problem is made even worse if the mass of data or of text is not stable, but is constantly changing, as it might be if data is continually being up-dated or the pool of texts is continually being added to. The extreme case, of course is the web, where each day documents die, new documents are added, old ones are modified. Each modification changes the state of the world and in doing so changes the base on which knowledge discovery is performed, invalidating previous legitimate hypotheses and providing the seeds of new ones.

The sketches of a few typical applications below are intended to help by providing concrete background. They are based, very loosely, either on examples frequently found in the literature or on the interests of partners in the project which first caused us to become aware of the special evaluation issues raised by knowledge discovery systems, the Parmenides project. (Parmenides 2004)

2.1 Identifying hidden associations and trends

A great deal of data has been collected on purchases made in a supermarket chain. Analyzing this data could reveal associations between purchases: do those who buy nappies also frequently by beer, for example, or do those who buy flour also buy eggs? The same data, collected over a period of time, also contains hidden information on how purchasing trends develop: perhaps people buy more low-fat products than they did five years ago, or more organically produced vegetables. Discovering hidden associations may help with marketing, while uncovering the hidden trends has obvious implications in terms of future commercial orientations.

2.2 Re-using research work

A large manufacturing company does its own in-house research, and has done for many years. Over a long period of time, it has built up a significant archive of reports and papers on the research that has been done, including research that perhaps led to dead ends or showed why it was unprofitable to pursue some line of enquiry. The company has developed an uneasy feeling that current and planned research is repeating previous work. Human searching of the archive in order to identify previous relevant work, because of the mass of material, would have to be limited to a search of titles and perhaps of abstracts of promising candidate documents. Not only would such a search be extremely time

consuming, it would probably fail to identify work where, for example, shifts in terminology over time obscure the direct relevance of the document or where extremely pertinent work is buried in a larger document with a title reflecting quite other preoccupations. Being able to identify and re-use relevant earlier work can avoid a great deal of wasted time and reduplicated effort.
There is an obvious parallel here to scientific research in general. If we imagine an area, work on genetics for example, where a great deal of work is done in many different sites, even without taking time into account it may be difficult to identify relevant work, especially if it has led to a dead-end.

2.3 Making predictions

A firm of investment advisors makes a profession of predicting the future. They monitor the specialized and general press in a search for events which may affect the value of investments. Monitoring involves a first stage of scanning news wire stories, trade journals and newspapers for items likely to be of interest, and a second stage where analysts who are specialists in the domain glean from the selected items any information pertinent to making predictions.
Once again, the mass of material to be scanned is so great that human resources alone cannot carry out the task. If computer support can contribute to reducing the enormity of the task, it might be possible both to scan a larger amount of material and to reduce to some extent the human error introduced by inattention or fatigue. In either case, the information base on which the analyst can work becomes much firmer.
The analyst may also, of course, make use of software intended to search out hidden associations, discover trends or look for specific pertinent information. In general, the complexity of any given knowledge discovery task may be such that success depends at least in part on a judicious choice of appropriate technologies and tools.

2.4 Other types of knowledge discovery

The applications picked out above have all become feasible only comparatively recently – indeed some might be thought to be still leading edge research rather than the basis of established commercial products. Other kinds of knowledge discovery systems have been around rather longer, including question answering systems, fact and information extraction systems and information retrieval in general. Both older and newer systems however share one characteristic which, from the evaluator's point of view, is of considerable importance: the top level task which the system is designed to support is of an extremely high order of generality: creating new insights, discovering associations hitherto unsuspected, picking out of a mass of information just that information which is relevant – all these are skills which are important in a myriad of different contexts and which can be applied to achieve innumerable different goals. This means that the potential users of knowledge discovery systems are in their turn many and various. And that brings us to the central issue of user-oriented evaluation of such systems.

3. User oriented evaluation and ISO

User oriented evaluation in the sense intended here is very closely related to work on software evaluation in general as typified by the ISO/IEC 9126 series of standards (ISO 2001a, 2003a, 2003b, 2004). The basic notion is that a user has a set of needs which can be set forth in the form of requirements. The quality of a piece of software is to be judged in terms of whether it satisfies those requirements.
Before going further, it might be useful to note that "user" in this context is a person, or even an organisation, with a task to be achieved. This should not be confused with another sense of user common in certain kinds of evaluation work, where the user is the person who sits in front of the computer and directly interacts with the system. To return to the supermarket example of the previous section, the user could well be the general manager of the whole supermarket chain, who is considering whether introducing knowledge discovery software might contribute to the overall task of increasing sales and thereby profits in each of the individual supermarkets which make up the chain. He is unlikely himself ever to use such a system, but in the sense intended here he is a user with needs that can be specified in terms of requirements on a system.
IS0 9126 concentrates on the characteristics of systems which determine whether it can meet the quality requirements. Because the standard is located at a very high level of generality, the quality characteristics in the standard have to be made more specific for any given type of software. An essential point though is that reification of the characteristics must lead to a level where metrics can be specified such that a system's performance with respect to a given characteristic can be measured.
(This is a very brief and therefore inevitably imperfect summary of the ISO 9126 proposals: more detail can of course be found in the standards themselves. Also, a related set of standards, ISO 14598, (ISO 1998, 199a, 1999b, 2000a 2000b, 2001b) concerns the process of evaluation, a topic which will be completely neglected here.)

4. EAGLES and ISO

Early EAGLES work was largely inspired by the first version of ISO 9126, published in 1991. EAGLES was specifically concerned with working out a specialization, or rather a series of specializations, of the ISO standard which would be applicable to software in the realm of human language technology. A deliberate policy of starting with softwares whose technological basis was relatively simple was pursued, so that early work looked at the evaluation of spelling checkers, of grammar checkers and of translation memory systems.
The technological basis of such systems is simple mainly because the basic functionalities desired of them are also simple. In at least one critical way, this in turn simplifies the evaluator's task. It is possible with such systems to define one or at most a small set of central functionalities which, if well performed, will contribute in very great measure to user satisfaction. As an example, a core functionality of a translation memory system is to retrieve from an archive of translations those portions of a text which have already been wholly or mostly translated, presenting the existing

translation as a candidate for re-use. If the system succeeds in this aim, most users will be fairly happy, modulo of course usability, robustness, portability and other such practical but important issues.
In these cases, setting a framework for evaluation comes down to identifying potential users or classes of users, modelling them in terms of their needs and defining the metrics which can be used to find out whether a specific system matches their needs.
An essential point here is that because the basic functionalities of the software are quite limited and quite precise, the variety of potential users is not very great. Finding out who they may be and organising them into classes with similar characteristics is entirely feasible.

5. ISLE and FEMTI

An attempt was made in the ISLE project to pursue a similar approach to more complex software with a correspondingly much richer range of possible users, machine translation software. The result (so far – the job is not finished) is FEMTI, a framework for machine translation evaluation which can be found at
http://www.issco.unige.ch/projects/isle/femti
FEMTI takes the form of two loosely structured taxonomies. The first of these models potential users of machine translation systems. It does so by looking separately at the translation task the user hopes to accomplish, the characteristics of the users of machine translation systems and, finally, characteristics related to the text to be translated. (In line with the earlier remark about users, users here may be end users of machine translation systems, end users of the translations produced by such systems or organisational users of translation.) These three elements (translation task, user, input text) are only the topmost layer of a rather detailed breakdown characterizing the kinds of jobs that people want to use machine translation for. As an example of how detailed the breakdown can ultimately become, one sub-characteristic of the translation task is using translation as part of carrying out dissemination, dissemination in its turn is broken down into internal/in-house publication and external publication, external publication is broken down again into publication aimed at a single client or type of client and publication where the same basic document has to be tailored to the specific needs and capabilities of different clients. The diagram below may help to make this break down clearer.

- Translation task
  - dissemination
    - in-house publication
      - …
      - …
    - external publication
      - Single-client
      - multi-client

The point here is not, of course, whether the choices made in building the model of machine translation users were judicious, or even whether the result is entirely satisfactory. The point is that, in the case of machine translation, a relatively complex software with a rather wide range of possible uses, it can be done: it is not total madness to attempt the task of modelling users of the software. And, of course, the question that we shall eventually come to is whether it is imaginable to do something of the same sort for users of knowledge discovery systems.
FEMTI's second taxonomy models machine translation systems as pieces of software. The taxonomy follows ISO in distinguishing characteristics that are internal to the system itself, such as what the theoretical basis of the system design might be or what classes of algorithms are used in its construction, and system external characteristics, characteristics that are observed when the system is in operation. The external characteristics are precisely those set out in the ISO standard – functionality, reliability, usability, efficiency, maintainability and portability, with the addition of one non-ISO characteristic, cost (considered a matter for management decisions rather than an evaluation issue by ISO). Again, this is only the top layer of a deep structure, which, as it deepens, becomes more specific to machine translation software. For example, maintainability, still following ISO, breaks down into a number of sub-characteristics, one of which is changeability, which, now specializing towards machine translation, covers, amongst other things, the ease of carrying out dictionary up-dating and the ease of changing grammar rules. The diagram below shows this break down:

- Maintainability
  - Changeability
    - Dictionary up-dating
    - Grammar rule modification
  - …

Once again, the critical point here is not what the elements of the breakdown actually are but that it has been possible to construct the taxonomy.
The two taxonomies are separate, but not unrelated. The basic principle is that tracing a path through the taxonomy modelling users and their needs will lead to pointers, embedded in the taxonomy, to items in the taxonomy modelling system characteristics. The existence of such a pointer is essentially saying "if this is one of the user's needs, this is a system characteristic which an evaluator should look at". In the ideal world, designing an evaluation for a specific user and context of use would involve checking through the user model to discover which characteristics are relevant in the case under consideration, collecting the pointers thus brought to attention, using the pointers to define relevant parts of the system model and incorporating those elements into the evaluation. (The real world is rarely, of course, that simple).
A critical point which has already been mentioned should be re-emphasized here: each system characteristic is the top node of a hierarchical structure of sub-characteristics, sub-sub-characteristics and so on to whatever depth is required. The leaf nodes **must** contain an indication of one or more metrics applicable to measuring a system's quality with respect to that node. Values for higher nodes are obtained by combining values for lower nodes. (The combining function

may be quite complex).
Once again, the detail of the FEMTI metrics is irrelevant: what is important is that the whole exercise of constructing a system model is pointless unless metrics can be devised. To put this in terms of a much more banal world, there is no point in telling anybody that an important quality sub-characteristic of clothes is that they should fit if there is no way of trying them on and no labelling system which renders trying them on unnecessary.
Thus FEMTI could be said to consist of three main elements:

- A model of (classes of) users
- A model of (classes of) systems
- A set of metrics whereby characteristics of systems can be measured

together with pointers linking the two models.

6. Evaluating knowledge discovery systems

All of the foregoing has been a lengthy prolegomenon to asking whether it would be possible to construct a framework of some sort for the evaluation of knowledge discovery systems. It is probably unrealistic to think that all possible kinds of knowledge discovery systems could be dealt with, but could we, perhaps, distinguish classes of knowledge discovery systems and set up a framework for each class? Does the ISO/EAGLES/ISLE work offer us any guidance in how to do so? This final section which tries to look at these questions will be something of a rag bag. In an attempt at structure, we shall pick up each of the main elements of the FEMTI model for machine translation and ask what the chief obstacles are to doing something similar for knowledge discovery systems. A final sub-section comments on some practical issues.

6.1 The users

The first of the two FEMTI taxonomies sets out to model users and their needs. When we ask if it would be possible to carry out the same exercise in the knowledge discovery domain, the main problem, it seems to me, is the enormous range of applications possible, even for one single type of system. Similarly, the domains of interest amongst potential users are enormously varied. Thus, finding the link between what a system can do and what will satisfy a user might prove an intractable problem. I think the distinction I am after here is close to the ISO distinction between accuracy and suitability as different sub-characteristics of functionality. A system is functionally accurate if it produces what its technical specifications say it will produce. It is functionally suitable if what it produces is actually useful to the user. To see that this is not just a specious distinction, let me take the example of commercially available terminology extraction systems. There is at least one such system where the aim embodied in the technical specifications of the system is to traverse a stretch of text, collecting all those sequences of two or more words (let's not worry about the definition of word) which occur twice or more in the text, thus producing a list where each such sequence is listed with an indication of the number of its occurrences in the text. The system does this job admirably: it is functionally **accurate**. But it lists sequences like "and the" with a very high occurrence rate, and such sequences are of no interest whatever to a terminologist. And it is an unfortunate fact of language that uninteresting sequences (on the definition built into the system) are much more numerous than interesting sequences, so much so indeed that I know of no terminologist who thinks that a tool based on these principles is of any use to support his work: the system is not functionally **suitable**.
The analogy with knowledge discovery systems is I think clear. But the problem with knowledge discovery systems is much greater. It is not too difficult to work out what would count as a suitable result for a terminologist: begging some issues, he wants something that will extract all and only potential terms. How do we discover what counts as suitable for users of knowledge discovery systems, especially when the users are so diverse and operate over a very wide range of domains. Is there anything to be gained by trying to define typical users and perhaps even typical domains?

6.2 The systems

The second FEMTI taxonomy tries to describe quality characteristics of systems, first in terms of system internal characteristics and then in terms of the external quality characteristics which make up part of the ISO standard. Here, clearly, it might be much easier to think in terms of classes of knowledge discovery systems rather than to try and find one description. But this is essentially a practical remark rather than a theoretical remark, since the union of descriptions of two or more classes could perhaps provide a larger composite model.
The theoretical problem seems to me almost an inverse of the suitability problem discussed in the last section. Knowledge discovery systems, as we have noted several times already, operate at a fairly high level of generality: they suggest associations, discover trends, hypothesize that two pieces of information are linked and so on. Their proposals, by the nature of the technology and of the tasks, can never be certain; they always require validation by a human. They are intended to provide support in achieving the task, not to perform an independent task whose results will have independent validity.
This raises the question of whether it is possible to link a particular task (whose accomplishment is a particular user need) unambiguously to a particular class of knowledge discovery system: can we tell, just by looking at someone's needs, what the system characteristics are that are pertinent to satisfying those needs? In other words, is there a level of generality at which evaluation designers can operate, or will it always be a case of scrutinising very closely the tasks each individual user wants to accomplish and working only in those individual terms?

6.3 Metrics

This last question brings us to the last link in the FEMTI framework, the metrics which are of necessity associated with each terminal quality sub-characteristic. It is notoriously difficult to find valid and reliable metrics even with much less complex systems. With knowledge

discovery systems it seems to me that the nature of the systems themselves adds greatly to this difficulty.
The first problem is a consequence of knowledge discovery systems inherently being intended as support tools. That means that it is not going to be possible to isolate some core functionality and provide a metric for it in the belief that the results obtained for this metric will say something about the overall satisfactoriness of the system. This may sound a little obscure: let me try to make it a little clearer through comparison with other kinds of software. If we look at machine translation systems, for example, it is highly likely that the speed of translation, measured in words per hour or whatever, is a sub-characteristic of efficiency of interest to many potential users. Once the system is running over the text to be translated, who the user might be can have no influence whatever on the results obtained for the metric: the system does all the work. Compare this to a translation memory system, where the user is working through the text sentence by sentence, launching a search for previous translations for each sentence, examining the translations proposed, accepting the proposals, rejecting them or modifying them. Here, how quickly the software searches the archive of previous translations is of relatively minor importance: it is the translator's ability as a translator which determines how quickly (and how well) the job gets done. A last example will get closer to knowledge discovery systems. In our department we teach intelligent web searching. As practical exercises we ask students to find obscure items of information, like the German name for the assembly of Spartan citizens in 400 B.C. All the students doing the exercise have been given the same training, but how quickly they find the answer varies enormously, and is mainly determined by their own ingenuity in thinking up the right question to ask. If knowledge discovery systems inherently suppose a user interacting intelligently with the system, how can we devise metrics which separate out system characteristics and user intelligence?

The second problem stems from the nature of the raw material over which the knowledge system is supposed to work. As we have already pointed out, it can be assumed to be a very large mass of potentially unstable material. This casts doubt on the utility of many conventional and well established techniques for establishing evaluation metrics, since they depend critically on being able to create a gold standard: a set of results which are by definition correct. (Good examples of such metrics can be found liberally in the ARPA/DARPA evaluation campaigns such as MUC and TREC, and in those European campaigns taking inspiration therefrom). A further problem stems directly from this observation: if it is assumed that the mass of raw data is too great for human analysis, how can silence be determined? How can we discover whether there are results which the system should produce but is failing to? Another way of saying all this would be to ask whether it is possible to produce any sort of gold standard dependent metric which would nonetheless accurately predict system behaviour over real text or real data.

6.4 Some practical remarks

The most obvious remark about the FEMTI exercise is that it was a considerable effort. First, any kind of general evaluation framework is unlikely to prove of practical use unless there is at least some level of consensus in the community concerned about the content of the framework. This was partly achieved with FEMTI through a series of workshops, eight in all, which brought together volunteers who were prepared to contribute to the elaboration of the framework. There was a small central group, funded partly by the National Science Foundation and partly by the Swiss Federal Office for Education and Science through the EU Framework programmes, but almost without exception, the participants in the workshops were self-funding and contributed their work on a voluntary basis. The participants came from both academia and private industry. An obvious question, then, is whether there would be sufficient support in the knowledge discovery community to sustain a consensus based attempt to find an evaluation framework? And, of course, even though the level of funding for FEMTI's central group was extremely modest, the existence of that group was key to the whole enterprise: without it, no workshops would have taken place, the results of workshops would have disappeared into the limbo of pious aspirations, nothing concrete would have been achieved.
Secondly, creation of an evaluation framework for a particular technology takes time. There are actually two aspects to this. The first is the time it takes for a technology to become familiar enough for common perceptions and common vocabulary to emerge. Machine translation has been around for something like fifty years. The first attempts at question answering systems or information retrieval go back almost as far, but data and text mining systems are very new technologies. Is the field mature enough for any attempt at generalisation about evaluation to make sense? The second aspect is simply elapsed time. Evaluation frameworks for complex systems are in themselves complex. Machine translation is relatively complex. After four years, a first version of FEMTI exists, is available on the web and seems to be appreciated. But it is far from a complete piece of work. It is incomplete even for the current state of the machine translation world and it contains inconsistencies. Although the metrics included are a fairly exhaustive collection of metrics documented in the literature, there has been very little work to date on correlations between metrics, and not much work on how to test the validity of individual metrics. The work has always been intended as a very long term on-going exercise, and will continue thanks to a modest amount of renewed funding. But knowledge discovery systems are by far more complex than machine translation systems: how long would it take to reach even the preliminary state of current FEMTI results? And how do long term efforts such as this fit in with a policy of research funding where funding typically comes in slices covering between one and two years?

7. Conclusion

Practical considerations are important, but in the end it is the theoretical issues which decide whether a project is worth

pursuing or not. This paper has raised a series of questions aimed at asking whether it is worth pursuing a programme of trying to create a pool of shared knowledge and expertise on how to evaluate knowledge discovery systems. The philosophy of the ISO 9126 standard has been used to structure the enquiry. In conclusion, it might be worth pursuing the ISO theme a little further.

The ISO standard assumes that there is a sort of quality chain: quality in terms of internal system characteristics predicts, at least in part, quality in terms of external system characteristics. And external quality characteristics predict, at least in part, what ISO calls "quality in use": the quality of a system as it is perceived by a user who is trying to get some job done using the system. He would like to get his job done efficiently, and in safety, and he would like to feel some satisfaction at how the job has been done. A good system, for his needs, is a system which will meet those criteria.

It is this quality chain which allows evaluation to be broken down into something more manageable than creating a trial installation and evaluating by seeing how users get on with using the system, a procedure as risky as it is expensive. The major question raised here is whether the quality chain holds, in some sense or another, for knowledge discovery systems, or whether the only kind of evaluation that makes sense is some kind of isolated quality in use evaluation with all the attendant problems that raises.

REFERENCES

- Parmenides 2004 : http://www.crim.co.umist.ac.uk/parmenides
- EAGLES 1996 : EAGLES Evaluation of Natural Language Processing Systems. Final Report. Available at
- ISLE Evaluation Working Group: http://www.issco.unige.ch/projects/isle
- ISO 1991: ISO/IEC. International Standard ISO/IEC 9126. Information technology – Software product evaluation. Quality characteristics and guidelines for their use. International Organization for Standardization/International Electrotechnical Commission, Geneva.
- ISO 1998: International Standard ISO/IEC 14598 –5. Software engineering – product evaluation – Part 5: Process for evaluators. International Organization for Standardization/International Electrotechnical Commission, Geneva.
- ISO 1999a: International Standard ISO/IEC 14598 –2. Software engineering – product evaluation – Part 1: General Overview. International Organization for Standardization/International Electrotechnical Commission, Geneva.
- ISO 1999b: International Standard ISO/IEC 14598 –4. Software engineering – product evaluation – Part 4: Process for acquirers. International Organization for Standardization/International Electrotechnical Commission, Geneva.
- ISO 2000a: International Standard ISO/IEC 14598 –2. Software engineering – product evaluation – Part 2: Planning and management. International Organization for Standardization/International Electrotechnical Commission, Geneva.
- ISO 2000b: International Standard ISO/IEC 14598 –3. Software engineering – product evaluation – Part 3: Process for developers. International Organization for Standardization/International Electrotechnical Commission, Geneva.
- ISO 2001a: International Standard ISO/IEC 9126-1. Software engineering – product quality – Part 1 : Quality model. International Organization for Standardization/International Electrotechnical Commission, Geneva.
- ISO 2001b: International Standard ISO/IEC 14598 –6. Software engineering – product evaluation – Part 6: Documentation of evaluation modules. International Organization for Standardization/International Electrotechnical Commission, Geneva.
- ISO 2003a: International Standard ISO/IEC 9126-2. Software engineering – product quality – Part 2 : External metrics. International Organization for Standardization/International Electrotechnical Commission, Geneva.
- ISO 2003b: International Standard ISO/IEC 9126-1. Software engineering – product quality – Part 3 : Internal metrics. International Organization for Standardization/International Electrotechnical Commission, Geneva.
- ISO 2004: International Standard ISO/IEC 9126-1. Software engineering – product quality – Part 4 : Quality in use metrics. International Organization for Standardization/International Electrotechnical Commission, Geneva.

Issues in Evaluating a Question Answering System

Gerhard Fliedner

DFKI GmbH
Stuhlsatzenhausweg 3
D-66123 Saarbrücken

Computational Linguistics
Saarland University
D-66123 Saarbrücken
fliedner@coli.uni-sb.de

Abstract

We present a short outline of a meaning-oriented open-domain question answering approach for German currently under development. It is in several respects different from current Question Answering systems, and the resulting system will differ in the user interaction. Therefore, the question of usability and user satisfaction becomes especially important: Does the system actually fulfil the user's information needs? Does it do so in a better way than other systems? This paper does not try to solve these questions; it is rather an attempt to present our current view of the situation, to point out some of the problems, and to present some preliminary thoughts on how to proceed.

1. Introduction

In this paper, we describe issues in the evaluation of a German meaning-oriented open-domain Question Answering (QA) system currently under development in our group. As our system will differ in several respects from current systems, we believe that evaluation will be especially important, even more so than for other systems.

We present first thoughts on the possibilities and the problems of the evaluation. The issues are not yet clearly defined and described in terms of evaluation metrics, i. e. broken down into quantitative, measurable attributes (EAGLES, 1999). We think, however, that some of these thoughts apply to QA systems in general and are thus worth taking up for discussion.

The paper is organised as follows: First, we give a short sketch of our system, followed by a summary of general and special evaluation issues. We then turn to shared task competitions as a means of evaluation. We argue that for a number of issues, individual usability tests will probably be more suitable and give some ideas on the experiment design and conduct we currently envisage. The paper closes with some general remarks and an outlook.

2. QA Using FrameNet Representations

We are currently building a Question Answering system that uses FrameNet representations for the actual information search (Fliedner, 2004a; Fliedner, 2004b). This is ongoing work within the Collate project[1].

Most current QA systems work roughly as follows (Hirschman and Gaizauskas, 2001): The type of the question and its focus are determined. The content words are then used to search the document collection for candidate documents using word-oriented IR techniques, possibly enhanced by query expansion. The candidate documents are then searched for passages that best fit the search, again based on matching the words in the passage with that of the query. Some systems use deeper linguistic processing at this stage (Moldovan et al., 2002a; Hovy et al., 2001). The answer that is presented to the user then is either a likely text snippet or is generated from the processing results.

Our approach differs in that it directly uses a FrameNet representation of the users' queries and of the document collection derived off-line. FrameNet (Baker et al., 1998; Johnson et al., 2002) is a lexical semantic database resource that contains semantic valency information for words (similar to thematic roles). A frame in FrameNet describes a situation with the participants and objects belonging to it. This is roughly comparable to event templates in Information Extraction, but less specialised and less domain-dependent.

Direct matching of the FrameNet representation derived from the question against that of the document collection will help us to abstract away from text surface issues such as wording or syntactic variants. This will allow us to find information in a meaning-oriented way. The representation is also well suited for generating natural language answers.

We are currently also considering allowing an interactive querying process where the user can put a number of questions to narrow down (or broaden) the search. This would allow a simple dialogue with elliptic constructions, especially follow-up questions like *'How about X?'*.

3. Evaluation Issues

We expect our final system to differ from current systems in several important points:

1. Short, grammatical answers

   We aim for short, grammatical answers. This is well supported by using the structured FrameNet representation. Current IR systems (especially Internet search engines) generally return whole documents – the user must find the relevant passage(s). QA systems have greatly improved in answer conciseness over the last

[1] *Computational Linguistics and Language Technology for Real Life Applications*, conducted jointly by Saarland University and DFKI GmbH, funded by the German Ministry for Education and Research, Grant numbers 01 IN A01 B and 01 IN C02.

years. However, answers are still sometimes not grammatical or do not fit the question.

2. Justification

   Our system will be able to return a justification for its answer upon request. If users are not satisfied with an answer or if they doubt its correctness, the system should be able to justify the answer, e. g. by displaying the relevant passage/document.

3. Higher Precision

   Using a more meaning-oriented search, we hope to achieve higher precision, i. e. the user gets less 'junk'.

4. Dialogue Capabilities

   Referring back to the last questions and answers in an interactive query process is more similar to information inquiries in human/human interaction.

All the points above lead to a number of questions that we would like to be able to answer in a system evaluation. Some of them are very general and apply to all QA systems, others are more specific.

Some of the questions are hard to objectify. As the system should finally be gauged by the users' needs, sometimes goals that seem desirable from a technician's point of view will not actually be optimal from a user's perspective. There has been agreement that answers should be concise. This has led to either limiting the number of words (50, 250) or demanding 'exact answers' in shared task evaluations such as TREC and CLEF. In real life, however, a user may find an answer that contains some more detail and/or justification more helpful. This special distinction is captured by the differentiation between *answers* and *responses* (Webber, 1986), where an answer has the pure facts, whereas a response is a co-operative reply to the user's actual information needs, possibly not overtly expressed in the question. This is mirrored by the observation that '*Actually, no clear definitions of exact answer have been formalised, yet.*' (Magnini et al., 2003) and that '*As with correctness, exactness is essentially a personal opinion.*' (Voorhees, 2002).

This leads to the conclusion that evaluation for QA systems, especially for those introducing new features, should be centred around the needs of individual users. In our opinion there are two ultimate goals: That users can efficiently find an answer to their questions and that they are content with the process. This should go together, but might differ in certain cases, especially among users with differing needs and/or levels of expertise.

One example might be cases of ambiguity where a question may refer to one of two (or more) things: The most efficient way for the system might be to always use the more probable solution and let the user protest afterwards when the less probable possibility was intended. Here, a system that politely disambiguates first may lead to a higher user contentness, even though the average number of required dialogue moves is actually increased. Recall the recurrent TREC example for ambiguity (Voorhees, 2001) and imagine, e.g., a user inquiring for the location of the Taj Mahal, and actually *meaning* the casino in Atlantic City, NJ. A QA system could either react by answering '*Near Agra*' and leave it to the user to figure out what went wrong, or it could ask a question along the lines of '*Do you mean the famous mausoleum or the casino?*'. This is mirrored by the introduction of the important, user-centred 'quality in use' level in evaluation that may 'override' external quality characteristics such as 'short dialogues' (EAGLES, 1999).

This said, we will try to list the most relevant sub-questions for evaluation here (partly following (Hirschman and Gaizauskas, 2001)). If all these sub-goals are optimally achieved *from the user's point of view*, then an efficient and satisfying system use should in principle ensue.

1. Answer Relevance

2. Answer Correctness

3. Answer Conciseness

   The answer should not contain irrelevant material and should not be excessively wordy. Furthermore, it should be unambiguous.

4. Answer Completeness

   Does the system's output fully answer the user's question? To make this question even more complicated, the users may not have fully expressed their information needs (i. e. the system's output is a correct answer, but not a suitable response).

5. Justification

   As mentioned above, the system should be able to justify its answers. It remains an open question if every answer should be accompanied by a justification, or if it is more helpful if the user can easily request a justification when in doubt. This choice may be influenced by the system's own trust in its answer (especially with heterogeneous data collections).

6. Answer Format

   In which format should the answer be presented to the user? In many cases, a phrase (e. g. a name) or possibly a sentence will be the most natural. In some cases (especially when comparisons are involved) a table or a graph will do better.

7. User Adequacy

   Ideally, a system should be able to adapt to the needs of every user. Thus, the user's experience level should be taken into account (e. g. novice, expert). This could, e. g., be reflected by the wordiness of answers or by always supplying a justification. This may also mirror personal likes/dislikes (e. g. one user wants full sentences as answers, another does not). It will in general mean, that different users prefer different answers to the same question (Burger et al., 2001). So, this last point should be seen as super-ordinated to all the above points.

4. Shared Tasks

Over the last years, shared task evaluations have been established and have very much gained in importance. In the question answering domain, the Text REtrieval Conference (TREC, `trec.nist.gov`, English) and the Cross-Language Evaluation Forum (CLEF, `www.clef-campaign.org`, multi/cross-lingual) are among the most important. Recent QA tasks are described in (Voorhees, 2002; Voorhees, 2001; Magnini et al., 2003). The core idea is that all participants receive a large data collection that they may process in any way they wish. Then they receive a set of test questions and must return their systems' answers to be scored and compared with the other participating systems.

There is agreement in the community that such shared tasks boost research and development, as research groups are encouraged to participate and therefore have to really get their systems up and running in time for the shared task (and fine-tune them for the next participation).

We therefore think that taking part in such a shared task is an important goal: A shared task gives one the unique opportunity to evaluate system features that are crucial to the whole system and must be evaluated in any case, such as answer correctness, in a comparative evaluation. There are, however, a number of points that we would like to raise in this connection:

1. Language

   In general, shared tasks tend to be in English, as it is likely to attract most participation. This, of course, does not help researchers working on other languages. For German, CLEF has offered monolingual tasks, however, as the ultimate goal is to foster cross-lingual IR, there is no guarantee that it will continue to do so.

2. Time

   The sheer mass of documents to be processed for a shared tasks (currently around one million documents) combined with the time constraints on the competition may exclude deep pre-processing of texts simply due to lack of time, even though the time needed for this pre-processing may be acceptable for potential uses.

3. Adequacy of Measures

   Shared tasks have an important advantage: They rely on measuring relatively hard facts, usually answer relevance, correctness, and conciseness. Though the evaluation is done by human correctors, there seems to be evidence that these can at least be objectified insofar as the *relative* ranking of different systems will be preserved through different annotators (Voorhees, 2001). There might be cases, however, where these measures just do not give the whole picture.

4. Applicability

   Shared tasks tend to test open-domain QA systems. While this is, of course, the most challenging application, closed-domain systems may thus be excluded.

These points do not deduct at all from the utility and necessity of shared tasks, they only underline the point that shared tasks will always lead to a compromise of 'minimal agreement', where different users and specialization cannot be taken into account and where thus specialised systems ('wrong' language, 'wrong' domain) may not be able to participate at all or may not show to the best advantage. Besides, only the first three of the goals listed above are addressed at all by shared task evaluations, not taking different necessary parametrisations for different user groups into account. To put it in a nutshell: Shared tasks are an invaluable tool to evaluate a number of important features of systems, and thus form an important part of a system evaluation, but they cannot address a number of issues that are more user-oriented, i. e. quality in use issues. We now turn to some ideas on user-centred evaluation.

5. User Requirements

In the preceding, we have stressed the importance of centring evaluation around user requirements. We currently plan a Wizard of Oz experiment with a small number of users to get a better idea of users' requirements with respect to our QA system. We will sketch the experiment in this section. It is to form a basis for both system design and later system evaluation.

We believe that the first step in finding out about user requirements is to find out more about who the users actually are. This is, of course, more straightforward for an evaluation of systems for a given usage, e. g. in a company. For research systems such as ours, whose main aim it is to investigate new techniques and methods, the end-use is less obvious, especially as it may hinge on the achieved results. The shared tasks try to balance this by using questions from a broad range of areas. If available, the queries are based on real users' questions to QA systems, adapted from system logs (Voorhees, 2002).

We are currently developing a scenario for our first experiment that is to be conducted as a Wizard of Oz experiment. The current outline is as follows: The users are supposed to be interested in business news. They consider buying shares in a certain market segment or a certain company (currently we envisage focusing on the German market for renewable energies, especially manufacturers of solar or wind energy plants). They are then asked to use the QA system to find out more about the current status of the companies. This would not so much focus on the stock market figures of the company, but rather on background information such as recent company acquisitions or sales, changes in key personnel, important orders, etc. It would also assume, that the user has some information on the issue (e. g. on relevant legislative initiatives). We think that this (or a comparable scenario) is suited to be tested on user groups with different background experience (especially 'hobby investors' *vs.* professionals).

The experiment we plan is set up as follows: The users are told that they are taking part in an experiment to improve certain features of a prototype system. The system itself is simulated by a hidden experimenter ('wizard'). The users would enter their queries via keyboard; the wizard would then use a standard search engine to find suitable an-

swers, re-formulate them as an answer string according to a guideline (e. g. *'use short, concise answers'*), and send this answer to the users' terminal. At this stage, we would simulate a 'perfect' system that makes little or no language-specific mistakes. Very general questions of design (such as: should all answers automatically be accompanied by a justification) could already be tested on users by testing different groups with different wizard guidelines.

This experiment will be accompanied by a questionnaire, possibly complemented by an after-experiment walk-through (see below). We think that this procedure should elicit important information on the following points:

- What sort of questions do users ask?
- Are users satisfied with the answers formulated according to the wizard's guidelines?
- Do users have suggestions for system improvement?

These answers will fundamentally influence system design: If, e. g., heavy inferencing would be necessary in the final system to answer two-thirds of the questions, then this module is essential. If, on the other hand, it is only required for a small margin of especially hard questions, then it could be postponed for later. During the design phase, additional experiments may become necessary, especially addressing how users deal with a less-than-perfect system.

Also, the findings will form an important basis for later evaluations of system prototypes – even if at that stage linguistic and processing questions will probably automatically take up more space.

6. Usability Evaluation

We believe that an individual evaluation of our eventual system can provide a great number of insights. This should be conducted as a usability test (Nielsen, 1994).

Ideally, such an evaluation should be conducted with many users in real life situations, i. e. by establishing their information needs and observing how they use the system to find suitable answers. Quizzing the users on their actual needs and their experiences with the system could dramatically help to improve overall system performance and system adaptability. The disadvantage is that this set-up will generally be very time-intensive and thus expensive.

We think, however, that insights can also be gained from evaluating a prototype system as a controlled laboratory experiment. This would be a more artificial situation with a somewhat artificial task: For reasons of comparability, all users, or at least user groups, will receive the same task.

As we have stressed above that efficiency and user contentness are the two main goals, we think that such an evaluation should be centred around the solution of one or more information inquiry tasks. This should not only be a list of single questions, but rather an information seeking scenario where the user is instructed to use the system to collect information on a certain subject.

We would build on the results of the preliminary Wizard of Oz experiment described above to find a similar scenario that the prototype system can cope with. The actual task will be broken down and adapted to the actual capabilities of the system. For example, questions that the system generally cannot handle should be avoided as far as possible.

While the task described above is interesting and challenging enough, it should still remain controllable within certain bounds. It is, however, not yet clear how much the individual user should be controlled and constrained: While we would, on the one hand, want as free an interaction as possible to observe individual user 'styles', this must, on the other hand, carefully been weighted against comparability: If the different users' query sessions differ too much from each other, the data can no longer be sensibly compared and evaluated.

In the actual experiment, a certain number of the measures mentioned above can be objectively measured (i. e. answer correctness), for others at least a certain measure of objectiveness should be obtainable (i. e. answer relevance). For other points (especially more user centred questions) we would need feed-back from the user. It would, for example, be very important to find out if the users consider individual answers complete.

For these more subjective issues, different methods could be used. We have tested most of them in other contexts (especially spoken natural language dialogue design) for other tasks.

- Immediate scoring

  The user might be asked to score the system's answers directly during the experiment, i. e. scoring conciseness, completeness, etc. on some numeric scale. Disadvantage: Disruption of the query process.

- Thinking aloud

  Users might be instructed to 'think aloud' during task completion. This might include eliciting comments like *'My, this was an extraordinary useless answer!'*.

- Questionnaire

  Users should be interrogated concerning their impression of the system after the experiment. Questions might take forms like 'While using the system, I found the system's answers relevant (1=always, 6=never).'

- After-experiment walk-through

  In this extended version of the questionnaire, users would walk through the recorded questions/answers step by step after the experiment and score separately.

Most probably, a combination of techniques will prove most helpful, e. g. using a questionnaire combined with a directed walk-through for special problem cases. Collecting some overall usability score has proven to be useful in practice (Nielsen, 1994), in spite of the difficulties in doing so and in comparing results.

So far, we have described only direct system evaluation. This should be combined and enhanced as far as possible with comparative evaluation. Comparative evaluation should make the life of the evaluators easier in principle: A certain feature must not be scored absolutely on some scale, but it is sufficient to say that a system (or system feature etc.) is doing *better* than some other. Three important comparison techniques are the following:

- System comparison

  By comparing different systems, one achieves insights into their relative strengths and weaknesses.

- Module performance

  To gauge the contribution of individual system modules, it can be helpful to compare the system's performance with that module switched on or off.

- System behaviour comparison

  We have mentioned above, that in many cases systems can be designed to behave differently. (Our example was the use of different methods to resolve ambiguities.) In such cases, a direct comparison between both possibilities can, of course, help to decide on one design.

It is, however, notoriously difficult to get the comparison right: In cases like the one here, it will be almost impossible to simply let the user compare directly. It is, e. g., dubitable if instructing the user along the lines of *'I'll now change the system answer behaviour. Please tell me afterwards which behaviour you liked better.*' will elicit the desired results. In general, we see three possible approaches:

- Tell the users explicitly what you are comparing and ask them to score.
- Confront them with different systems/system behaviours without telling them and elicit comments.
- Test different user groups on different systems/system behaviours and compare the results.

While the last possibility is attractive in that one must not ask one user to compare things (explicitly or implicitly), it has the disadvantage that user *groups* are, of course, more difficult to compare, as the members of the group are individuals and may differ in many respects. Therefore, one needs larger groups in general to get significant results.

Currently, we think that it would be very helpful to test different user groups on at least two systems with comparable functionality. This would mean that we would need to be able to test at least one other German QA system with comparable features. We will try to elicit (or simulate) at least three different user levels. Additionally, personal likes and dislikes should be established in the questionnaires and will lead to improvements on the configurability level.

This should be complemented by performance testing of the individual modules of our system to find out, how well they work and what needs to be improved (cf. (Moldovan et al., 2002b)).

7. Conclusion and Outlook

We have presented some thoughts on the evaluation of a German QA system introducing some novel features. Establishing user requirements early on using Wizard of Oz testing will form an important part of the evaluation.

While many questions remain open, we have identified a number of respects in which the system should be evaluated. Some of them can readily be objectified and measured, for others, this is more difficult.

Ideally, we will do individual system tests on the one hand, and comparative evaluations, on the other hand, and we will also take part in suitable shared tasks.

8. References

Baker, Colin F., Charles J. Fillmore, and John B. Lowe, 1998. The Berkeley FrameNet project. In *COLING 98*.

Burger, John, Claire Cardie, Vinay Chaudhri, Robert Gaizauskas, Sanda Harabagiu, David Israel, Christian Jacquemin, Chin-Yew Lin, Steve Maiorano, George Miller, Dan Moldovan, Bill Ogden, John Prager, Ellen Riloff, Amit Singhal, Rohini Shrihari, Tomek Strzalkowski, Ellen Voorhees, and Ralph Weishedel, 2001. Issues, tasks and program structures to roadmap research in question & answering (Q&A). Document Understanding Conferences Roadmapping Documents.

EAGLES, 1999. EAGLES. Evaluation working group final report. Technical Report EAG-II-EWG-PR.1, Expert Advisory Group on Language Engineering Standards.

Fliedner, Gerhard, 2004a. Deriving FrameNet representations for question answering applications. Accepted for *NLDB 2004*.

Fliedner, Gerhard, 2004b. Towards using FrameNet for question answering. Accepted for *LREC Workshop "Building Lexical Resources"*.

Hirschman, L. and R. Gaizauskas, 2001. Natural language question answering: the view from here. *Natural Language Engineering*, 7(4):275–300.

Hovy, Eduard, Ulf Hermjakob, and Chin-Yew Lin, 2001. The use of external knowledge in factoid QA. In *TREC 2001*.

Johnson, Christopher R., Charles J. Fillmore, Miriam R. L. Petruck, Colin F. Baker, Michael Ellsworth, Josef Ruppenhofer, and Esther J. Wood, 2002. FrameNet: Theory and practice. Internet: `http://www.icsi.berkeley.edu/~framenet/book/book.html`.

Magnini, Bernardo, Simone Romagnoli, Alessandro Vallin, Jesús Herrera, Anselmo Peñas, Víctor Peinado, Felisa Verdejo, and Maarten de Rijke, 2003. The Multiple Language Question Answering Track at CLEF 2003. In *CLEF 2003*.

Moldovan, Dan, Sanda Harabagiu, Roxana Girju, Paul Morarescu, Finley Lacatusu, Adrian Novischi, Adriana Badulescu, and Orest Bolohan, 2002a. LCC tools for question answering. In *TREC 2002*.

Moldovan, Dan, Marius Paşca, Sanda Harabagiu, and Mihai Surdeanu, 2002b. Performance issues and error analysis in an open-domain question answering system. In *ACL 02*.

Nielsen, Jakob, 1994. *Usability Engineering*. Morgan Kaufmann.

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

Voorhees, Ellen M., 2002. Overview of the TREC 2002 question answering track. In *TREC 2002*.

Webber, Bonnie Lynn, 1986. Questions, answers and responses: Interacting with knowledge-base systems. In Michael L. Brodie and John Mylopoulos (eds.), *On Knowledge Base Management Systems: Integrating Artificial Intelligence and Database Technologies*. Springer.

User-Oriented Evaluation in Information Extraction

Roman Yangarber

New York University
Courant Institute of Mathematical Sciences
715 Broadway, New York, NY 10003, USA
roman@cs.nyu.edu

Abstract

Our focus is the application of Information Extraction (IE) technology to a higher-level Knowledge Management task, namely, information surveillance, in any area, such as financial reporting, monitoring of the spread of infectious diseases or terrorist activity. Analysts and Knowledge Management workers in these areas commonly use keywords (KWs) to seek out facts in large collections of documents. Since KW search has been known for some time to be inadequate for many types of research scenarios, IE was introduced, in part, to fill this gap. While IE is a fairly mature technology, several technical bottlenecks continue to inhibit its wider use. One of them is the lack of effective **evaluation**. We propose to make the use of IE more productive by focusing evaluation in IE on the utility of the results to the end-user, in addition to other, more abstract notions of correctness.

1. Introduction

Information Extraction (IE) is a text understanding task which involves finding facts in natural language texts, and transforming them into a logical or structured representation - e.g., a table in a relational database. Our IE engine, (based on the system originally described in Grishman (1995)), has been customized to extract facts on a variety of topics, viz.:

- Executive Management Succession
- Corporate Mergers & Acquisitions
- Rocket/Missile Launches
- Airplane Crashes
- Natural Disasters
- Infectious Disease Outbreaks

For example, for **Infectious Disease Outbreaks**, the system finds reports of epidemics around the world. For each outbreak the system determines the *name* of the disease, the *location*, *date*, *number* of victims, whether they are sick or dead, a short *description* of the victims, and a *link* back to the original document (for further details, please see Grishman et al. (2002, 2003)).

The objective of IE is to populate such tables in a database—with no further specification of how these tables are used beyond IE. Thus it should be stressed that IE is not an end in itself, but is rather a *mid-level* technology, or a support tool for higher-level decision making.

Evaluation in IE has been established via the **MUCs**—the series of Message Understanding Conferences (e.g., MUC6 (1995); MUC7 (1998)). The scoring metrics used in the MUCs are: *recall*, which measures the proportion of facts present in the text correctly extracted by the system, and *precision*, which measures the proportion of the extracted facts that were extracted correctly. These are often combined into a single score, the *F-measure*, which is their harmonic mean. F-measure is a common way to compare performance across IE systems.

We will argue that in order to properly assess the utility of IE, rather than relying on such measures of performance, the developers of IE must engage experts from the respective subject domains in a rigorous program of evaluation and mutual feedback. In this way, we must establish a dynamic interdisciplinary association, and pay special attention to the association's optimal functioning. It is interdisciplinary because the ultimate users of IE are from outside the computational linguistics field.

A common—and natural—roadblock to success of interdisciplinary projects is the absence of common values among the professionals in the different disciplines. What seems a fair measure of success to the computational expert may not be relevant to the epidemiologist or the intelligence analyst. Thus we should develop new measures of performance, in terms useful not only to the computational scientist (as has been the trend in the past) but to the end-user as well, and devote effort to establishing and testing evaluation metrics that will be meaningful across the disciplines involved.

In this paper, we will propose specific methods to begin to assess IE in an end-user-oriented fashion.

2. A Sample Research Scenario

We will use the bio-medical scenario, Grishman et al. (2003), mentioned above, for illustration. The research task involves finding facts in the ProMED-Mail database of articles related to epidemics and spread of infec-

tious disease, to answer the higher-level research query. **ProMED-Mail** is the Program for Monitoring Emerging Diseases, sponsored by the International Society for Infectious Diseases (ISID); it is one of the world's largest publicly accessible emerging diseases detection programs with over 32,000 subscribers in 150 countries. For instance, ProMED-Mail was instrumental in the detection and surveillance of the recent SARS epidemic, and is routinely followed by individuals at the US Centers for Disease Control and Prevention, the National Institutes of Health, the Department of Defense and other international, national, and state agencies for information about emerging diseases and potential acts of bioterrorism.

The simulated research task for an analyst to perform using the ProMED-Mail database,[1] is:

> "Dengue fever (DF) is an acute febrile, viral disease frequently characterized by headaches, bone or joint and muscular pain, rash, and leucopenia as symptoms. Dengue hemorrhagic fever (DHF) is a life threatening complication of dengue fever and is characterized by four major clinical manifestations: high fever, hemorrhagic phenomena, often with hepatomegaly, and in severe cases, signs of circulatory failure. Such patients may develop hypovolemic shock, resulting from plasma leakage. This is called dengue shock syndrome (DSS) and can be fatal.
>
> A ProMED posting (31 July 2001) describes the capture in a second Tempe, Arizona neighborhood by Maricopa County health officials of an Aedes aegypti mosquito. A. aegypti is the primary vector for dengue fever and its variants, as well as yellow fever. The mosquito's appearance marks the first time the species has been found this far north (a suburb of Phoenix). Until now, it was found only as far north as Tucson. It is common in Central and South America.
>
> Use the system to determine the incidence of DF, DHF, and DSS in the Western Hemisphere roughly north of 20 N 00, but also including Mexico City (19N54)."

The task requires the analyst to identify *where* exactly dengue has been spreading, above 20N. That means identifying as many locations as possible, where incidents have occurred. Note, that this is not the same as identifying every report about dengue above 20N, or even of identifying every incident of dengue above 20N. Specifically, if there are multiple reports about an incident, or multiple incidents reported at the same location, that is not relevant; we are only interested in identifying the location, "placing a dot on the map."[2]

If one had access to only KW-based IR tools, one could build a complex query containing a disjunction of all countries north of 20 N 00. At first glance that might seem easy, since there are few countries there; but states in the US often appear in text without explicit mention of the US. One might then extend the disjunction to include states; but, large cities, like Los Angeles and San Francisco, are also often listed without reference to the state. And text may refer to other types of geographical locations altogether, ("the Lake Tahoe area", etc.)

Time may be another constrained variable; we may not be interested in incidents that have occurred prior to a certain date, say, 50 years ago.

Perhaps more to the point are two other objections: as the query grows and the number of KWs expands, it is more prone to inaccuracies which will degrade precision and require more "debugging"; further, in the documents returned by the query, the end-user will need to search for the text containing the KWs, verify that it describes an incident of interest, and only then determine what happened, where and when.

One informant on disease surveillance, (Plummer (2003)) sums it up: "There is a great deal of information out there that is published by a myriad of organizations. The limiting factors on the surveillance side are ... timeliness, access, and organization [prior to the analysis process]. [...] As an example, Promed, as far as I can tell, provides a great deal of information as prose, and does not organize the information in a more useful tabular format. Useful, yet labor intensive."

In current practice, there is a heavy reliance on IR-like KW-based search—and consequently, even heavier reliance on analyst's prior knowledge.

Thus, an IE-style database of facts may be an appropriate tool to relieve such reliance.

3. Difficulties in Evaluation of Utility

In the IE community, MUC-style evaluation metrics have for some time been the "received" measure of success, the bottom line in evaluation of IE. Researchers have used the MUC-style quantitative results to indicate the degree to which they are succeeding in extracting facts. Recently, this science is coming under more critical scrutiny.

On the technical side, e.g., Kehler et al. (2001), have called for a re-assessment of the effects of recall, precision and F-measure metrics on the development process, citing conditions where these metrics can be misleading.

[1] As suggested by an editor of ProMED, Pollack (2002)

[2] Another query might call for deeper information, e.g., the severity of each incident, or "the size of each dot," but that would be a different task.

Further, Knowledge-Management (KM) specialists are increasingly aware that a true evaluation of "performance" of a Text Understanding system, must take into account measures of *utility to the end user*, in addition to (if not in place of) measures of *correctness* in terms of formal metrics.

In a MUC-style evaluation, the metrics and scores may have meaning to IE developers, but not to an end-user, whose task involves utilizing the extracted information. The question that an evaluation should answer is not whether the Knowledge Discovery (KD) system achieves some F-measure on a test corpus, but whether the end user finds the information identified useful in accomplishing her/his (higher-level) task.

Specifically, in reference to our sample research scenario, the key question is: how many *distinct areas* of incidence of the disease was the user able to identify correctly—using the ProMED-Mail document base, and possibly aided with an IE system.

Surely, for the end-user, the disease surveillance specialist, that is the bottom line, and not how many relevant documents the system retrieved, or how many incidents of dengue outbreaks the IE system detected in the document base.

The evaluation, then, should focus on this question *directly*, to be of value to the end-user.

4. Developer- vs. User-Oriented Evaluation

We propose that in delivering IE systems to information surveillance professionals, we conduct several different kinds of evaluation:

- "Classic," MUC-style evaluation;
- "Dry," developer-oriented evaluation;
- "Wet" user-oriented evaluation.

Before discussing the different kinds of evaluation, we should clearly distinguish different *levels* of facts. We will call those facts in which the end-user is ultimately interested *higher-level* facts, in contrast to *mid-level* facts which the IE system produces.

In our running example, a higher-level fact is "dengue has affected location X". The IE system may not necessarily extract this type of fact. The IE system is intended to fill a table (or tables) in a database with facts that should be useful in answering a wide range of higher-level scenarios.

These are the mid-level facts; the developer and the end-user need to agree upon what these facts will be. This agreement is reached through negotiation: the end-user specifies what is desired, and the developer specifies what is feasible. In our example, these facts might be of the kind "disease D affected N people at location L at time T."

The mid-level facts are more widely useful, because they carry more specific information, which can be aggregated for making generalizations at a higher level.

We make the following two claims concerning effective evaluation:

Claim 1 *The developer-oriented evaluation should focus on mid-level facts, while end-user evaluation should focus on higher-level facts.*

Claim 2 *Evaluation is a phase in a process, where the IE system is (at least partially) operational and is being used by the end-user.*

MUC-style evaluation: For compatibility with other IE results, we may conduct standard MUC-style evaluations. To this end, we need to employ specialists to build small test corpora (50–100 documents per scenario), which involves some expense. The collective MUC experience provides us with the scoring tools, corpus markup tools, and guidelines necessary to carry out such evaluations.

"Dry" developer-oriented evaluation: As the IE system evolves over the life of a project, developers will naturally want to monitor its performance, to assure that it is steadily improving. When a new version of the system, $version_i$, is released by the developer, to replace the previous $version_{i-1}$, it should be evaluated against the document collection.

As a corollary to claim 1, part of the end-user's research, and normal use of the IE database, should involve the end-user in validating facts that are correctly identified by the IE system, and invalidating (and possibly correcting) facts incorrectly identified by the system.[3]

Developer-oriented evaluation is then straightforward. First, the *verified* portion of the fact base accumulated so far—i.e., records manually verified by human review—are permanently archived, as fact base $version_{i-1}$. (The manually-verified fact base remembers the *incorrect* records, as well as the *corrected* records.) Then the upgraded IE system $version_i$ is applied to the entire document collection. An automatic evaluation procedure can then track the number of:
cp = *correct* facts from $version_{i-1}$ preserved in $version_i$,
ip = *incorrect* facts from $version_{i-1}$ preserved in $version_i$,

[3]This functionality can be easily built into the end-user's interface. After reviewing a fact during his/her research, the user can press a button to accept or reject the fact. This input is needed for keeping track of the IE system's performance, and without it this type of developer-oriented evaluation is not feasible.

cl = *correct* facts from $version_{i-1}$ lost in $version_i$,
il = *incorrect* facts from $version_{i-1}$ lost in $version_i$.

These numbers will provide a base for making estimates about the improvements of the system's performance. For example, the *relative* recall can then be estimated as $rr_i = \frac{cp}{cp+cl}$, relative precision as $rp_i = \frac{cp}{cp+ip}$, etc.[4]

Dry evaluations will be useful for estimating the incremental contributions of individual *components* of the IE system as they are refined over time.

"Wet" evaluation: we have conducted evaluations on the *utility* of IE results in the epidemiology domain Grishman et al. (2003). The published results were a bit weak, since they employed the *developers'* view on utility; however, that view was developed in consultation with the specialists who proposed the task.

A "wet" evaluation should involve human subjects. The specialist specifies an actual or a simulated ("mock") research task. This task is a typical problem in the target area, requiring the Knowledge Discovery system for accessing information for further analysis. The task shapes the evaluation experiment. Two or more analysts are then set against each other to perform the task; while one is using the IE system and its extracted fact base, the others use standard tools, such as search engines. The results of the analysts are then collated and statistics are collected, establishing the degree of utility of the systems and tools.

The dimension of **time** is a crucial aspect in the wet evaluations: obviously, given a Google-like search engine plus unlimited time, one can solve any knowledge-discovery task. We should measure utility under realistic *time constraints*. The evaluation tools should register the results each analyst has gathered so far at, say, 15 minute intervals. This will enable the evaluators not only to judge who got the best final results, but also to draw curves tracking results against a time scale. This will make the evaluation much more informative.

5. Conclusion

The application of IE systems to higher-level Knowledge Management tasks is a complex undertaking, involving collaboration among specialists from different disciplines. We have proposed that an optimal way to proceed in such a setting involves separating the concerns of developers from those of the end-users, and providing developer-oriented evaluation in terms of mid-level facts, and a separate, user-oriented evaluation in terms of high-level facts.

This should help reconcile the (possibly orthogonal) points of view as to the utility of the IE system to the Knowledge Discovery process, as held by the computational specialist vs. the domain specialist.

Finally, given these different sources of evaluation information, the interesting question can be addressed, whether, and to what extent, there is a meaningful correlation between the results of the different evaluations.

[4] So, for example, $rr_i = 1$ implies that $cl = 0$, or that no correct records were lost, going from $version_{i-1}$ to $version_i$, whereas $rr_i = 0$ implies $cp = 0$, or that all correct records were lost, etc.

Acknowledgements

Kind thanks to Gary Ackerman of the Monterey Institute of International Studies for his thoughtful comments.

References

Grishman, R., 1995. The NYU system for MUC-6, or where's the syntax? In *Proc. 6th Message Understanding Conf. (MUC-6)*. Columbia, MD.

Grishman, R., S. Huttunen, and R. Yangarber, 2002. Event extraction for infectious disease outbreaks. In *Proc. 2nd Human Language Technology Conf. (HLT 2002)*. San Diego, CA.

Grishman, R., S. Huttunen, and R. Yangarber, 2003. Information extraction for enhanced access to disease outbreak reports. *Journal of Biomedical Informatics*, 35(4).

Kehler, Andrew, Douglas Appelt, and John Bear, 2001. The need for accurate alignment in natural language system evaluation. *Computational Linguistics*, 27(2).

MUC6, 1995. *Proceedings of the 6th Message Understanding Conference (MUC-6)*. Columbia, MD.

MUC7, 1998. *Proceedings of the 7th Message Understanding Conference (MUC-7)*. Fairfax, VA: `www.itl.nist.gov/iaui/894.02/`.

Plummer, Andrew, 2003. Private communication.

Pollack, Marjorie, 2002. Private communication.

Position Paper

Evaluating the Results of Unsupervised Ontology Learning[1]

Myra Spiliopoulou and Roland Müller
Otto-von-Guericke-Universität
Magdeburg
Germany

The importance of meta-information for effective search in document collections has led to intensive research on the subject of ontology learning. The establishment of an ontology through a human expert is a cumbersome and expensive process. Moreover, it is very fragile against change. As document collections evolve, expert-defined templates become obsolete in the sense that their recall declines. Given these shortcomings, the automation of part of the ontology learning process and the minimisation of expert involvement seem desirable alternatives.

Automated ontology learning takes two forms:

- The human expert specifies a set of templates, a software engine learns these templates and exploits them to build an ontology connected to a document collection.
- A software engine scans a document collection and derives patterns, which serve as ontology components connected to the collection.

The first form is a typical supervised learning task, which expects that the templates have already been specified in the corresponding labour intensive process. The second form, which is our subject here, is unsupervised and completely data driven. Its objective is to derive patterns that describe the data collection and are appropriate as components of an ontology, e.g. as new concepts or as new relationships among existing concepts.

The involvement of supervised or unsupervised learning algorithms in the process of ontology establishment calls for a methodology of evaluating their findings. We are interested here in evaluating the *quality* of the algorithm's output rather than in the performance of the software in terms of resource consumption or execution speed.

For supervised learning, the evaluation typically takes place against a reference model built by human experts, sometimes referred to as a "gold standard". The supervised learning algorithm builds an approximation of this reference model and is evaluated with respect to the quality of this approximation. Statistically speaking, this approximation is a global model of the population being studied, which in our case is a document collection. In terms of methodology, the algorithm is trained on a subset of the labelled data encompassing the gold standard and is tested on its ability of guessing the labels for the remaining data.

In unsupervised learning, there is no a priori gold standard describing the population (here, the document collection). Rather, unsupervised learning algorithms are invoked to generate global or local models that describe that population. This raises the question of appropriate evaluation methods. One straightforward answer would be to build a dedicated gold standard. However, since the ultimate goal of applying an unsupervised learning algorithm is to gain interesting insights into the behaviour of a population, it is also reasonable to evaluate the findings of an unsupervised learning algorithm on the basis of *interestingness* measures.

In the talk during the workshop, we will first elaborate on the two aforementioned approaches for the evaluation of the results of an unsupervised learning algorithm. We claim that the evaluation on the basis of interestingness measures is not appropriate because it corresponds to evaluating the human expert who selects the measures and specifies their thresholds. We also claim that the evaluation against an a priori designed gold standard is inappropriate because it defeats the purpose of using unsupervised learning algorithms that *discover unknown models*. Then, we propose a mid-field solution to the evaluation problem. We design a process that builds a reference model taking into account the expert-imposed constraints under which the unsupervised learning algorithm operates. We then explain how an algorithm can be evaluated on the basis of this model.

[1] This work is partially funded under the EU contract IST-2001-39023 PARMENIDES, http://crim.co.co.umist.ac.uk/parmenides

How to get the right systems to the right users?

Nancy L. Underwood
Nancy.Underwood@issco.unige.ch
ISSCO / TIM, Ecole de traduction et d'interprétation,
Université de Genève, Boulevard du Pont-d'Arve,
CH-1211 Genève 4,
http://www.issco.unige.ch

Abstract

This paper considers a number of issues in evaluating Knowledge Discovery systems from the users' point of view. It is intended as a contribution to the discussion on how to ensure truly user-oriented evaluation of such systems. As such it puts forward a number of problems and open questions to be discussed during the workshop..

1. Introduction

This paper is intended to provide input to a discussion during the workshop on how to ensure trully user-oriented evaluation of KD systems. For the purposes of discussion we first present a working definition of KD at a general level (Section 2). Section 3 considers the nature of user-oriented evaluation and points out some of the shortcomings of more traditional metrics which have been applied to similar, but simpler systems for extracting information from data. In Sections 4 and 5 we present some more concrete suggestions for better metrics and the beginnings of a checklist for modelling users of KD systems. We conclude with a number of open questions concerning the next steps which should be taken to achieve truly user-oriented evaluation of KD systems.

2. Knowledge Discovery: a working definition

In this paper we take the goal of Knowledge Discovery (KD) to be the uncovering of knowledge or rather information from data (in the form of patterns and relationships) which was not previously apparent or readily accessible because of the amount and complexity of data available to the user. Thus KD systems are typically applied to very large amounts of data which may also be heterogenous (e.g. police intelligence gathered from different sources) and/or constantly updated (e.g. newswires, customer purchase records, the web).

In order to reveal such knowledge, rather complex systems involving, for example, data mining and intelligent information extraction techniques have been developed. In general, data mining techniques produce results which by their very nature are statistical and indicative rather than factual and which need to be interpreted by the user in order to be useful. Thus a KD system is very much a supporting technology to aid the user in carrying out a higher level task based on the knowledge which is uncovered.

These properties of KD give rise to particular for problems specifically user-oriented evaluation.

3. User oriented evaluation of KD systems

Carrying out a user-oriented evaluation of a system involves evaluating how well that system addresses the needs of the user. In ISO terms (ISO/IEC, 2001), the question we need to address concerns how we can reliably evaluate the "Suitability" of a system or piece of software. This should not be confused with evaluating the "Accuracy" of a system which will only tell us how well the system performs the tasks which it has been designed to carry out in the first place. Although accuracy may sometimes be a good inicator of suitability, it is perfectly possible to have an well designed and implemented system which fulfills all its design specifications but which is nonetheless not suitable for a particular user or group of users.

3.1 Shortcomings of traditional metrics

Much previous work in evaluation (see e.g. EAGLES (1996), the MUC and TREC web sites) has focussed on developing metrics for system attributes which are also assumed to predict the utility of a system for a typical user or group of users. In the field of KD evaluation we do not rule out the possibility of defining some such metrics. However, the open-ended nature of KD systems calls this approach into question.

In the fields of IE and IR evaluation for example, the standard metrics have traditionally been precision and recall. These metrics rely crucially on the construction of gold standards, which specify in advance the correct result which a system should produce. However by definition, the results obtained from KD should be new and unpredictable. So in principle, it should not be possible to pre-determine the results a system should produce and therefore such metrics seem to be difficult, if not impossible, to apply in the case of KD. It may of course be possible to try and plant

patterns or trends in the data which the system is supposed to discover, but artificially rigging the data calls into question the validity of the metric in terms of how well it reflects the real-life application of the system and for those users whose data is also constantly updated, the reliability of such precision and recall metrics in predicting the utility of a system is even more doubtful.

Another problem in defining metrics which will predict the suitability ofKD systems for users in general is the difficulty in defining a typical user of KD systems. It seems that the goals of (potential) KD system users can only be defined at the highest level of abstraction in terms of the desire to gain new insights from data which they may not be able to achieve themselves. The type of insights they are looking for and what they will do with the results produced by a KD system can be very varied indeed. These may range from seeking commercial intelligence which might help a user to decide whether to invest in a particular business or market, through trying to identify unusual or illegal events and activities which a view to preventing terrorist attacks to scientific or medical research looking for previously unnoticed correlations between instances of diseases.

Added to this, the level of experience and the expectations of users may also be very different. Given the complexity and relative newness of the technology, this can be expected to have a significant effect on the suitability and utility of a system for a specific user.

In the next section we present some preliminary ideas on the sorts of metrics which might be applied to evaluate the utility of a system for a particular user.

4. Metrics for evaluating utility

Since direct testing of system attributes cannot completely reliably predict the utility of a KD system for a particular user, then one approach would be to ask users themselves to rate the utility of the results which are produced. This could be broken down into the following characteristics of results.

- **Novelty** (do the results tell me something I didn't know before?)
- **Credibility** (does this result accord with my own knowledge? do I believe it?)
- **Relevance** (does this result contribute to the task I am trying to carry out?)
- **Understandability** (can I understand the results presented?)
- **Subsequent use of the result** (was the information useful to me? did I go on to use it in a meaningful way?)

Such "soft" metrics relying on the user's more or less subjective judgement of the utility of the results produced, should not be confused with those commonly applied by developers of KD systems when, for example, measures of confidence and support are calculated for an association rule to determine the reliability that rule. Even if a revealed pattern or trend is statistically very well supported by the data, if the user doesn't find it credible, he may well ignore it and it is therefore not useful to him.

As with all software evaluations the issue of usability is also an important one in determining utility. The more user-friendly a system is, the more likely users are to use and thus benefit from it. Given the complex nature of KD systems and the fact that, for some systems at least, the user is required to expend a great deal of effort in preparing and cleaning his data, as well as iteratively refining parameters and re-processing the data until a useful or interesting result has been achieved, questions of usability may be particularly important for these types of systems. As a starting point the following usability characteristics from the ISO quality model

- understandability
- learnability
- operability
- attractiveness

could be expanded in relevant sub-characteristics to answer the question of how easy it is to understand and learn to use the system.

In addition to the rather subjective characteristics listed above, it is also necessary to look at more objective measures of utility which address the effects of deploying the system. The following metrics can apply to the overall organisation rather than simply the end-user(s) of the system

- **Efficiency** (has the system affected efficiency?)
- **Cost** (what has been the financial costs? savings?)
- **Quality** (how has the overall quality of work been affected?)

However these metrics and others like them, have a major drawback in that they all presuppose a running system which has been deployed for some time. KD systems tend to be both complex and expensive to deploy and need to be fine-tuned for a specific user's needs before a reasonable evaluation of this type can be carried out. It is not likely that a user could take an off-the-shelf system on a trial basis (as for example might be possible with an MT system) in order to find out how useful it might be to him before deciding whether to invest in a system. This is not to say that evaluations of deployed systems are worthless because they can be used for providing feedback to system developers or for an organisation to monitor the results of deploying the system with a view perhaps to deploying it elsewhere. Such an evaluation may even provide general

insights to a potential new user of KD systems, if his requirements seem similar to those of the current user.

But for the potential user trying to decide whether to invest in such a system from scratch an evaluation based solely on such metrics is not feasible. In such a case, then, it is vital to understand as much as possible about the user's requirements which will affect the a system's suitability for him. This leads us to the question of which types of user characteristics we need to understand in order to carry out a reliable evaluation. In the next section we propose an high-level intial sketch of how one might begin to model a user of a KD system.

5. Modelling the user

Questions of utility and usability crucially depend not only on the task which the user has to carry out but also on his experience and expectations[1]. One can imagine a number of different characteristics of a user we may find it useful to model but at the highest level we need to know who the user is and want he wants to do with the KD system.

In the following we present a first pass at a high-level checklist which will hopefully be discussed and further fleshed out during the workshop:

5.1 Who is the user?

The term "user" can cover a multitude of different actors with different tasks and roles.

- **The user's role**
  - medical researcher
  - meteorologist
  - intelligence analyst
  - research manager
  - marketing manager
  - .....

Above is a more or less random list of the types of roles, potential users may have. The role which a user may have will affect the type of information they are seeking and the sorts of results which may be acceptable to them in terms of how much interpretation of results (and refining of the KD process) they are prepared to engage in.

- **The user's experience and level of expertise**
  The question of experience and expertise applies both to the subject domain of the data and to the use of similar (if considerably simpler) technology:

  a) Expertise in the subject domain
  - domain expert
  - newcomer to the field
  - expert in a related dicipline
  - .....

Such differences will affect the user's ability to assess the results produced by the KD system (does this trend seem plausible? do these association rules accord with my knowledge of this domain? what can/should I do with these results?)

  b) Experience with similar systems/tasks
  - sophisticated user of similar but "simpler" tools (e.g. search engines, DB queries)
  - novice user of databases, search engines etc)
  - .....

The user's experience with similar technologies and techniques will affect both his acceptance of the system and his ability to use it optimally in carrying out his task.

- **The user's expectations of the system**
  The expectations which a user may have are very likely to be dependent on his level of experience and expertise. However, given the hype which surrounds any relatively new technology, it is worth finding out what a potential user expects of the system. For example, in talking to a variety of potential users of a KD system under development we have come across the following expectations, some of which are more or less realistic:

  – the system will monitor a set of data sources and alert me to particular trends or events
  – the system will help me to improve on the results which I already get by traditional searching or information extraction techniques
  – the system will produce results which I can use immediately without further interpretation.
  – I expect to work further on interpreting the results

Even if some user expectations seem unrealistic, it may be that the system in question is potentially useful if the user were to amend or lower their expectations.

User expectations are clearly related to the question of what the user wants to do with the system.

[1] In a sense this is true of all software evaluations, if a potential buyer of an MT system is not a professional translator expects it to produce perfect translations of free texts which can be immediately published then he will be sorely disappointed. And indeed many users of MT systems have learnt to temper their expectations the hard way. With a highly complex and very expensive technology like KD systems the problem of matching expectations and experience with a new technology seems even more acute..

5.2 What does the user want to do?

This question refers in general to the task or goal which the user wants to accomplish with the help of the KD system.

- **The task** (describing the task or goal of a user may be rather complex since his overall goal may comprise a number of smaller sub-goals which the KD system is expected to support him in achieving)

- **The data** (in KD the nature and quality of the data which is available is a key factor in both how KD should be applied and the quality of results which can be achieved)

In addition users of KD will typically not be working alone but rather as part of a larger organisation and therefore we also need to consider the setup into which the system would be incorporated:

- **User setup**
  - workflow
  - hardware
  - software
  - the current divisions of labour
  - .....

We will no doubt discover other characteristics of the user which should be modelled.

6. What next?

We have presented some suggestions for user-centred evaluation metrics and a sketch of how to begin modelling users. Neither of these should be considered exhaustive and would need to be further refined in greater detail. Incomplete as they are, they do however give rise to a number of questions:

- Having said that users of KD systems can differ widely, will it be possible to generalise over user types at all? Is it, for example, possible to predict users expectations from their previous experience and levels of expertise?

- A model of a user must clearly inform the evaluation of the system, but how far can the features in the model be directly translated into metrics which can be applied to the system? How desirable would this be?

- When new technologies and techniques become available they are often associated with quite ambitious claims which affect the user's expectations of what it is possible to achieve. By modelling the user and his expectations can we also contribute to a more realistic set of expectations? In other words should we also be educating the potential user as to the real potential of KD systems?

The importance of user modelling in evaluation is not by any means a new concept (see e.g. EAGLES 1996).. However often in the past it seems that only lip-service has been paid to this aspect of evaluation.

References

- EAGLES (1996). *EAGLES Evaluation of Natural Language Processing Systems*. Final Report. EAGLES Document EAG-EWG-PR.2, ISBN 87-90708-00-8. Center for Sprogteknologi, Copenhagen.
- ISO/IEC (2001) International Standard ISO/IEC,9126-1. Software engineering – Product quality – Part 1: Quality model. International Organization for Standardisization / International Electrotechnical Commission. Geneva.
- MUC (Message Understanding Conference) website: www.itl.nist.gov/iaui/894.02/related_projects/muc/
- TREC (Text REtrieval Conference) web site trec.nist.gov/

A Note on User Oriented Evaluation of Knowledge Discovery Systems

Andreas Persidis Ph.D
andreasp@biovista.com,
Biovista, Athens, Greece.

Abstract

This note looks at some questions related to user attitudes to knowledge discovery systems and to features of those systems which render them more acceptable to end users.

1. Background

This note is written from the point of view of one who hopes to make use of knowledge discovery systems in the general area of creating and distributing knowledge as a commercial resource. The emphasis is on a knowledge expert, a user whose job is to discover and develop new knowledge and who is making use of a computer system exploring a collection of text or data in order to facilitate his task and perhaps to suggest insights that he might not have otherwise developed.

2. Knowledge discovery as hypothesis generation

Within the framework of the perspective described above, the essential aim of knowledge discovery systems is to support the generation of hypotheses. In a sense, the hypothesis is already present in the existing literature which the system uses as its raw material. But the user, unassisted, would probably not hit upon the hypothesis found by the system, perhaps because there is too much data to be processed, perhaps because he would not make the imaginative leap required. By uncovering links between seemingly disparate chunks of knowledge, a knowledge discovery system supports the user in discovering and investigating novel hypotheses. This does not mean, of course, that the hypothesis is necessarily valid: once proposed, it must be further investigated and validated or invalidated. How the hypothesis may be shown to be valid is usually specific to the particular domain being investigated.

There are three main ways in which a knowledge discovery system can be said to create new knowledge.

First, the system may uncover a developing trend, increasing or decreasing, or may confirm that a trend stays stable. This is an example of the creation of new knowledge and is represented by data mining and related approaches.

Secondly, the system may apply a chunk of knowledge in a problem area different from the one it was created in. This is transfer of knowledge from one domain to another, resulting in the creation of new knowledge. Case-based reasoning and analogical reasoning are examples of research in this mode of knowledge discovery.

Thirdly, in a more general sense, it may identify a link between chunks of knowledge where the previously unsuspected link itself constitutes the new knowledge.

3. Judging the value of the knowledge discovered

Creating scientific hypotheses is an essential step in the R&D process. Creating validated hypotheses however is a much more interesting and difficult proposition, which in many cases, such as drug development, has important financial implications. Seen in this light, it is clear that the value of the discovered knowledge increases with the quality of the hypotheses made. For example, in the pharmaceutical industry, it is relatively easy to generate drug candidates. However, the cost of developing a drug candidate into an approved commercial drug is enormous: it would certainly be prohibitively expensive to do exploratory development work on every candidate proposed. Thus, for this industry, the best knowledge discovery system is the one which proposes as candidates only those candidates which have the best chances of passing the various toxicity, efficacy tests and so on, leading to the cost-effective development of an industrial product.

The measure, then, of the value of a knowledge discovery system is the quality of the predictions it makes.

4. Quality of Hypotheses

In saying this, all we have done is to displace the problem; the inevitable question now is how to judge whether one hypothesis or prediction is better than another. Below is a list, almost certainly not exclusive, of possible ways in which the quality of an hypothesis might be measured.

First, one hypothesis may be supported by more facts than a different hypothesis. For example the ‘fact’ that one protein is implicated in some biological pathway may be supported by research reported in 5 different scientific papers. All other things being equal we could say that an hypothesis based on this fact is more valid than one where the fact is supported by only three scientific papers.

Secondly, the “facts” themselves may differ in degree of believability. for example we may place greater weight on a fact reported by a research group with a high reputation in a peer reviewed journal with a high citation index than a fact reported by a not so well known group in a journal with a

lower citation index. The believability of the source is therefore a second parameter affecting the overall value of a hypothesis.

Thirdly, hypotheses are usually validated through a variety of tests (experiments) that either confirm or refute them. For example, in the drug development process a drug candidate is evaluated against animal models, toxicity and efficacy tests and finally clinical trials in humans. The more such tests it passes the higher the chances are that the candidate will eventually become a commercial drug. In general then the more screening tests an hypothesis survives the higher its value is.

Finally, an hypothesis might benefit from a degree of cognitive comfort. Any hypothesis is made in the context of facts which support it. If the supporting facts are in a context which is close to the context of the problem to be solved, it is easier and more straightforward to make the analogical reasoning steps which would lead to accepting the validity of the hypothesis.

5. User perspective

To quite a large extent what has been said in the last section already directly addresses the perspective of users of knowledge discovery systems: indeed, when we talk of the quality of hypotheses it is quality in the eyes of the beholder – the user using the system to investigate a domain and to create new knowledge for his own purposes – which is in question. This is most clearly the case when we talk about cognitive comfort, but even the importance of facts and their relative reliability is directly affected by a particular user's interests.

However, there are other features which will contribute strongly to whether a user finds it easy to work in partnership with a system or not.

First, it is important that the user should be able to understand how the system got to the hypothesis it has generated. He should be able to ask for justification of the system's conclusions, and the system should be able to provide them.

Secondly, the hypotheses offered should be homogeneous, at least to the extent that it should be possible to compare them one to another and rank them on the basis of different criteria supplied by the user.

Thirdly, a system that offers ways to proceed further is likely to promote user acceptance. To continue with the drug development scenario, for example, the system could itself suggest wet lab experiments that could be performed in order to validate the hypothesis.

Finally, a system might offer the possibility to work at different levels of resolution, allowing the user to select various levels of detail at which analysis might be performed. For example, a data mining system provides a different level of analysis from a system that identifies specific relationships between domain concepts and hence choosing one or other of these ways of looking at the data gives a very different picture. Ideally, it would be useful to be able to "zoom in and out" of levels of abstraction of a knowledge domain, seeing what kinds of hypotheses and conclusions are generated at each level.

6. Some final remarks on evaluation

The system features mentioned in the last section are not very problematical for evaluation – they are almost what early EAGLES work would have called "facts" – items that can be checked just by looking to see if the system has them, in the same way that one can check the languages a machine translation system translates between or the standard price of a commercial software.

Things are not so easy with the suggestions of section 3; there are many open questions about how to count facts, how to judge or rate their reliability or how to measure cognitive comfort. Nonetheless, we hope to have suggested fruitful avenues of enquiry.

REFERENCES

– EAGLES (1996). *EAGLES Evaluation of Natural Language Processing Systems*. Final Report. EAGLES Document EAG-EWG-PR.2, ISBN 87-90708-00-8. Center for Sprogteknologi, Copenhagen.

Towards Best Practice Standards for Enhanced Knowledge Discovery Systems

Roland Stuckardt

Johann Wolfgang Goethe University Frankfurt am Main
Im Mellsig 25, D-60433 Frankfurt am Main, Germany
roland@stuckardt.de

Abstract

Assessing enhanced knowledge discovery systems (eKDSs) constitutes an intricate issue that is understood merely to a certain extent by now. Based upon an analysis of why it is difficult to formally evaluate eKDSs, it is argued for a change of perspective: eKDSs should be understood as intelligent tools for qualitative analysis that support, rather than substitute, the user in the exploration of the data; a qualitative gap will be identified as the main reason why the evaluation of enhanced knowledge discovery systems is difficult. In order to deal with this problem, the construction of a *best practice model* for eKDSs is advocated. Based on a brief recapitulation of similar work on spoken language dialogue systems, first steps towards achieving this goal are performed, and directions of future research are outlined.

1. Elaboration of Problem Statement

The user-oriented assessment of enhanced knowledge discovery systems is a sophisticated problem that is understood merely to a certain extent by now. It imposes a series of *challenges* for which no ready-made solutions are available:

1. In contrast to less complex applications, there is no direct correlation between the performance of the natural language processing base technology and the usability as perceived by the user. For applications such as spell checking and voice recognition, quantitative evaluation measures (percentage of recognized incorrectly spelled words, transcription accuracy) can be expected to correlate quite well with perceived usability. In contrast, for enhanced knowledge discovery systems, no suitable quantitative criteria seem to be readily available.

2. It is difficult to formally define a prototypical task that matches the knowledge discovery needs of all, or at least of a large fraction of users. Too much depends on the specific application scenarios (and of their user-specific perception), which seem to be difficult to standardize and, hence, to model beforehand.

3. Enhanced knowledge discovery typically works on huge amounts of data. Due to this *and to the complexity of the knowledge discovery task*, it is regarded to be unfeasible to construct respective reference data intellectually through human annotators. In this regard, it is important to understand the difference to restricted knowledge discovery scenarios such as basic information extraction, the task of which consists in the document-*local* combination of information only, which may, with considerable efforts, be modeled by suitable text annotation schemes. This is impossible with enhanced knowledge discovery, which, in general, involves relating information contributed by different documents.

4. The homogenity of the data may vary, particularly regarding *type* (e.g., domain and genre of documents) and *reliability*. In contrast to the prototypical application cases that have been considered during the classical evaluation studies (such as the Message Understanding Conferences (MUCs), cf. (MUC 7, 1998; MUC 6, 1996)), the document sets to be processed are not necessarily well-behaved. In particular, they may contain non-factual texts that express differing opinions or points of view on a particular topic. This makes the task of constructing reference data considerably harder.

5. The data as well as its homogenity may vary over time, as in the case of web-based knowledge discovery applications. The same may hold with respect to the typical tasks of the users.

From all this, it follows that it is difficult to define how a "good" output of the knowledge discovery process looks like. Tasks like the identification of market trends seem to be simply too abstract to arrive at a level of formal transparency as achievable for more restricted tasks.

The subsequent sections elaborate upon the issues pointed out above. In section 2., previous work on formal evaluation is recapitulated; in particular, the notions of intrinsic vs. extrinsic evaluation are discussed and related to the problem of assessing enhanced knowledge discovery systems. Building up on this analysis, section 3. proposes a change of perspective: enhanced knowledge discovery applications should be considered as tools that, in large parts, assist in rather than carry out for themselves the analysis of the data: *as enhanced browsers for the qualitative exploration of data, they support rather than substitute the user*, who hence remains responsible for the central intellectual component of the task. This leads to the identification of the *qualitative gap* (section 4.), which will be singled out as the main reason why the evaluation of enhanced knowledge discovery system is difficult. Section 5. draws some important conclusions and suggests promising ways to deal with this problem. In particular, it is argued in favour of the statement of *best practice guidelines* for enhanced knowledge discovery applications. Based on a brief recapitulation

of similar work on spoken language dialogue systems, first steps towards achieving this goal are performed, and directions of future research are outlined.

2. Intrinsic vs. Extrinsic Evaluation

According to, e.g., (Mani, 2002), efforts of evaluating natural language processing systems may be categorized along various dimensions. The intrinsic vs. extrinsic distinction turns out to be of particular importance here:[1]

> **Intrinsic** *evaluations test the system in itself;* **extrinsic** *evaluations test the system in relation to some other task [...].*

According to the above problem statement, tasks to be assisted by the application of enhanced knowledge discovery systems are typically too complex in order to infer application performance from experiments at intrinsic evaluation level only. On the other hand, *generic* extrinsic evaluation imposes problems as well, since, as initially identified, standardizing the knowledge discovery task across users and across specific application scenarios is regarded to be unfeasible in many cases. Clearly, it is possible to extrinsically evaluate systems in *specific* application contexts. However, results are unlikely to generalize; thus, such evaluations cannot be taken as expressive substitutes of in-situ experiments in the particular application scenario an enhanced knowledge discovery system is aimed for.

So how to deal with this situation, according to which, in the case of *enhanced* knowledge discovery tasks, intrinsic evaluation is feasible, but not sufficiently expressive, whereas extrinsic evaluation would be expressive, but is not expected to yield results that generalize across users and application scenarios? Let's take a closer look at the issue why extrinsic evaluation is unlikely to yield once-for-all predictions regarding the performance of enhanced knowledge discovery systems. Figure 1 illustrates the generic application scenario of knowledge discovery systems. The input to the system consists of potentially heterogeneous collections of source documents, which might contain texts as well as tabular data and graphics. These documents are submitted to the knowledge discovery application system, which, possibly driven by a user query, yields an output that can be considered as a *view* on the input document collection. There are many types of operations that might be involved to generate this view, be it textual or graphical information extraction, information retrieval, data mining, or categorization based on similarity criteria.

The essential distinction, however, regards two different stages of processing: (1) the *algorithmic symbol transformation* performed by the knowledge discovery system, which comprises the different types of operations mentioned before, (2) and the *qualitative intellectual interaction* of the understanding user with the system, which comprises the statement of appropriate queries, the analysis of the output, eventually followed by the drawing of conclusions regarding the particular knowledge discovery need. By definition, intrinsic evaluation is related to the processing at the algorithmic stage. In general, these evaluation experiments are based on intellectually annotated corpora and

[1](Mani, 2002), p. 223-4, typographical emphasis by Mani

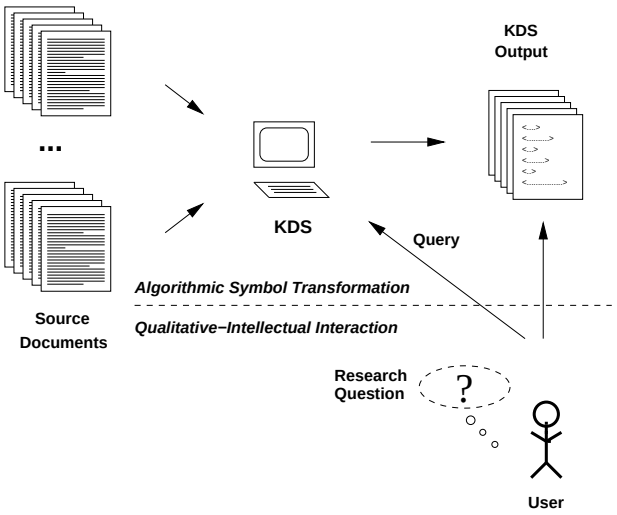

Figure 1: Generic application scenario of KDSs

on formally defined performance measures, which can be computed without further human involvement. In contrast, extrinsic evaluations refer to the output at the qualitative-intellectual stage of analysis, which might aim at the solution of quite abstract and heterogeneous tasks. This leads to a central observation regarding why evaluation of enhanced knowledge discovery systems is hard:

> *Enhanced knowledge discovery systems typically cannot be evaluated extrinsically according to general standards because the main surplus value of the knowledge discovery process is generated in a heterogeneous way at the qualitative-intellectual stage of analysis.*

3. Change of Perspective: Enhanced KDSs Support Qualitative Analysis

This hints at adopting a different perspective: enhanced knowledge discovery systems should *not* be looked at in the same way as at their ancestors with restricted scope, e.g. textual information extraction systems as considered during the MUCs, which can be meaningfully assessed by intrinsic evaluation. Instead, they should be understood as intelligent browsers that support, rather than substitute, the user in the qualitative exploration of the data.

In this sense, their contribution is similar to the contribution of software solutions for computer-*supported* content analysis, which are employed in the Social Sciences (e.g. (Fielding and Lee, 1991; Huber, 1992)). Essentially, these systems assist the user in retrieving and browsing data that might be relevant with respect to the specific research question. In particular, they provide enhanced functionality for the creative-explorative play with the data, such as cut-and-paste tools to manually extract parts of the data and facilities for the intellectual classification of the data according to user-defined categorization schemes. This enables the user to intellectually generate views over the data in order to gain insight in her field of research. This contrasts with computer-*based* content analysis systems, which employ (usually elementary) *automatic* categorization of textual data (cf. (Mohler, 1989)): while software systems that

support content analysis might offer tools for retrieving relevant content as well,[2] the central decisions of how to classify the data and of which conclusions to draw regarding the research question are left to the discretion of the user.

4. The Qualitative Gap

Thus, as in the case of software systems for computer-*supported* content analysis, rather than aiming at an automatic deep analysis the output of which is near to the answer, enhanced knowledge discovery systems are designed to foster intellectual understanding. Regardless of whether one subscribes to the point of view that there is a principal upper bound concerning the algorithmic explicability of cognitive processes, this can be interpreted as acknowledgement of the fact that the scope of algorithmic knowledge discovery will always be limited due to restricted coverage of state-of-the-art technology as well as due to practical feasibility issues, and that the actual *understanding* of the data remains up to the discretion of the user anyway. Based on these observations, the notion of the *qualitative gap* can be defined:

> *Software solutions that support the intellectual exploration of the data through the user, such as enhanced knowledge discovery systems, implicitly acknowledge the existence of a qualitative gap, which is due to practical limitations of the technology for automatically identifying relevant content: to optimally support the user in gaining insight in particular fields of research, tools for automatic content analysis are supplemented with features that allow for creatively browsing the data. Since, typically, the qualitative gap between the scope of algorithmic analysis and the requirements according to the research topics to be investigated is considerable, intrinsic evaluation can be expected to be not expressive enough.*

As further argued in section 2., whether generic extrinsic evaluation could be employed instead depends upon whether the knowledge discovery task (in particular, its qualitative component) can be standardized across users and across specific application scenarios.

5. Implications - Towards Best Practice Guidelines for Developing eKDSs

The above discussion has revealed that one central property of enhanced knowledge discovery systems is the typically considerable gap between the contributions of the individual technology components and the (typically quite abstract) insight into the research topic gained by the user through usage of the system as a tool that supports, rather than substitutes, understanding of the data. Regarding the issue that the components that can be subjected to intrinsic evaluation contribute only quite indirectly to the success in particular application scenarios, enhanced KDSs closely resemble other classes of complex Natural Language Technology applications.

[2] be it basic string search or enhanced retrieval and extraction functionality

5.1. Best practice guidelines for NLP applications: objectives and development issues

A related topic has been investigated for *spoken language dialogue systems* (SLDS), which exhibit the analogous property that their usability and perceived degree of usefulness highly depends on the particular application context, i.e. on the information needs and on the communicative or interactional preferences of the typical user. This identification of a gap between technology-related intrinsic criteria and the observed usefulness at extrinsic level has led to the development of *best practice guidelines* for spoken language dialogue systems, which, according to (van Kuppevelt et al., 2000), p. 207, are to be understood as

> *[...] a mapping from functional parameters to parameters of design and development*

Developing best practice guidelines hence means (ibd., p. 207f),

> *[...] to determine exactly what the mapping is like, and how its salient properties are best explained to a broad spectrum of laymen and professionals who find themselves confronted with the problem of getting an SLDS that answers their needs.*

According to these definitions, best practice guidelines have general scope in the sense that they are intended to fulfil the requirements of all involved stakeholders, viz. deployers, developers, customers, and users, i.e. (ibd, p. 206)

> *[...] to enable them to make accurate and successful design and implementation decisions, in accordance with broad consensus of what must be best practice in this particular engineering domain.*

Acknowledging the intricacies of designing appropriate spoken language dialogue interfaces thus in effect amounts to recognizing this issue as a creative intellectual engineering activity based on well-founded standards rather than as a matter of solid craftsmanship that merely relies upon the application of basic schematic knowledge.

The above discussion urges upon the conclusion that the statement of best practice guidelines should be the approach of choice for coping with the challenges of development, selection, and optimization of enhanced knowledge discovery systems. As in the case of spoken language dialogue systems, the degree of success of a solution highly depends on factors determined by the particular application context. Hence, the postulated objectives for the development of best practice guidelines for SLDSs can be taken as guiding principles for respective work on eKDSs. According to (van Kuppevelt et al., 2000), best practice guidelines should cover three closely related issues: (1) *stock-taking of the state-of-the-art* in order to enable the stakeholders to quickly inform themselves about the range of options for design, implementation, and evaluation; (2) *quality control* through the provision of criteria that support the selection of options that are best suited to particular application requirements; (3) *economic control*, to be achieved by making available a repository of resources in order to foster the

reuse of existing components, design know-how, and gathered experience.
Hence, as required for dealing with the typical scenario sketched in the workshop description, best practice models in particular provide criteria for the design of optimal solutions that fit best within particular application contexts.

5.2. Best practice guidelines for eKDSs

Thus, instead of entering into the ad-hoc discussion of how to deal with this typical scenario, a principled approach is advocated. The DISC best practice model, which covers the three above issues identified by (van Kuppevelt et al., 2000), is centered around a series of fundamental *aspects* of SLDS (system components as well as abstract development issues); it discusse them along a common scheme of *main items* (cf. (DISC, 2000)).[3]
It is proposed to take this approach as the point of departure for respective work on eKDSs. Regarding enhanced knowledge discovery systems, some main *aspects* are:

1. *Information Extraction Engines*, qualified by type of data (textual, graphical etc.),
2. *Information Retrieval Engines*, qualified by type of data,
3. *Data Mining Engines*, qualified by type of data,
4. *Categorization Engines*, qualified by type of data,
5. *Indexing Schemes*, qualified by type of data
6. *Query Engines*, qualified by type of data
7. *Knowledge Sources*, e.g., supported types of data, covered encoding schemes (.doc, .pdf, .ps, email archives, .jpeg, .tiff, etc.), static vs. dynamic data, intranet and/or internet resources etc.), amount of data to be processed, reliability and homogenity issues,
8. *Graphical User Interface*,
9. *Human Factors* (types of users, their degree of experience and previous knowledge etc.),
10. *Systems Integration*,
11. *Knowledge Discovery Objective* (as specific as possible, as abstract as necessary).

While some of the abstract aspects are immediately adopted from the SLDS best practice model (*Human Factors*, *Systems Integration*, the more concrete ones are not, as they correspond to specific system components of knowledge discovery systems without counterpart in the realm of dialogue systems. There is a further important difference that should be noticed here: regarding eKDSs, the extent to which particular systems differ with respect to the individually relevant aspects is considerably larger than regarding SLDS, which typically instantiate *all* aspects of their best practice model. A particular knowledge discovery solution might include an information extraction engine for graphical data, whereas another system might cover *textual* input only. Hence, the recommendations provided for the *Systems Integration* aspect are necessarily situated at a more abstract level; they strongly interdepend with the particular knowledge discovery objective, which, as a consequence, should be covered by a separate dedicated aspect. Again, this illustrates that, compared to many other natural language engineering problems, the development of eKDSs is a particularly intricate matter.
As far as applicable, each aspect should be discussed along several dimensions[4]: (a) grid (factual properties), (b) life cycle (development issues), (c) evaluation, (d) checklists, (e) glossary, and (f) references. Much specific previous work has been done on these issues. For instance, it might be referred to the experiences and resources gathered at the various DARPA- and EC-funded evaluation contests, e.g. TREC (information retrieval) and MUC (information extraction). Thus, to a large extent, modeling best practice amounts to an in-depth analysis of the state-of-the-art of the above-identified aspects of knowledge discovery solutions. Further aligning these different sources of knowledge according to the standardized scheme of a best practice model necessitates a considerable research effort.

6. The Next Steps

Proceeding along similar lines as followed during development of the DISC model, the elaboration of the best practice guidelines for eKDSs might be accomplished in three stages: (a) analysis of the state-of-the-art through data collection from different evaluation sources, (b) identification of particular constraint-oriented (i.e. application context sensitive) evaluation criteria, and (c) criteria integration, the output of which consists in the best practice methodology proper that provides high-level criteria for the informed choice among the technological options. Obviously, the last-mentioned stage embodies the major intellectual challenge.
According to the above considerations, modeling best practice regarding eKDS can be regarded to impose even more intricate challenges than in the case of SLDSs. Mainly due to the qualitative gap, the extent to which particular eKDSs differ with respect to their individual relevant aspects is considerably larger. Thus, the development of a sufficiently expressive best practice model constitutes a major research effort that should be addressed by a joint project with partners from commercial as well as non-commercial research, comprising all types of stakeholders (developers, deployers, customers, users). This project is necessarily interdisciplinary, as it covers issues from a broad range of disciplines (linguistic and mathematical models of content analysis, software system engineering, human-computer interaction).

[3] In accord with its objective, the resource repository of the DISC best practice guide has been made freely available at the web page (DISC, 2000). DISC is extensively documented in numerous online and offline publications, e.g. the deliverables made available at (DISC, 2000) or the book (Bernsen et al., 1998).

[4] according to the DISC terminology, *main items*

7. References

Bernsen, Nils Ole, H. Dybkjaer, and Laila Dybkjaer, 1998. *Designing Interactive Speech Systems: From First Ideas to User Testing*. Berlin, Heidelberg: Springer Verlag.

DISC, 2000. The disc best practice guide. available (March 1st, 2004) at http://www.disc2.dk/.

Fielding, Nigel G. and Raymond M. Lee, 1991. *Using Computers in Qualitative Research*. SAGE Publications.

Huber, Günter L., 1992. *Qualitative Analyse. Computereinsatz in der Sozialforschung.*. München, Wien: R. Oldenbourg Verlag.

Mani, Inderjeet, 2002. *Automatic Summarization*. Amsterdam/Philadelphia: John Benjamins.

Mohler, Peter Ph., 1989. Computergestützte inhaltsanalyse: überblick über die linguistischen leistungen. In Batori and Lenders (eds.), *HSK 4. Handbuch zur Sprach- und Kommunikationswissenschaft*.

MUC 6, 1996. *Proceedings of the Sixth Message Understanding Conference (MUC-6)*. Morgan Kaufmann.

MUC 7, 1998. *Proceedings of the Seventh Message Understanding Conference (MUC-7)*. Published online, formerly (December 9, 1999) available at http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/co_task.html.

van Kuppevelt, Jan, Ulrich Heid, and Hans Kamp, 2000. Best practice in spoken language dialogue systems engineering - introduction to the special issue. *Natural Language Engineering*, 6(3 & 4):205–212.