Evaluating a Normalized Conceptual Representation Produced from Natural Language Patient Discharge Summaries

P. Zweigenbaum, Ph.D., J. Bouaud, Ph.D., B. Bachimont, Ph.D., J. Charlet, Ph.D., J.F. Boisvieux, M.D., Ph.D.

DIAM — Service d'Informatique Médicale, Assistance Publique – Hôpitaux de Paris & Département de Biomathématiques, Université Paris 6

{pz, jb, bb, jc, jfb}@biomath.jussieu.fr

The MENELAS *project aimed to produce a normalized conceptual representation from natural language patient discharge summaries. Because of the complex and detailed nature of conceptual representations, evaluating the quality of output of such a system is difficult. We present the method designed to measure the quality of* MENELAS *output, and its application to the state of the French* MENELAS *prototype as of the end of the project. We examine this method in the framework recently proposed by Friedman and Hripcsak. We also propose two conditions which enable to reduce the evaluation preparation workload.*

INTRODUCTION

Medical language processing (MLP) has now grown mature enough to consider evaluation issues seriously. As a concrete outcome, Friedman and Hripcsak[1] recently proposed a collection of criteria to help make the evaluation of MLP systems more reliable. MLP systems aim at producing representations of the medical information found in free text documents. These representations are more and more "concept-oriented", which raises specific issues for evaluation. The main aim of the MENELAS project[2,3] (Jan 1992–Mar 1995) was to build a normalized conceptual representation of free text patient discharge summaries (PDSs) in French, English and Dutch, in the domain of coronary diseases. This conceptual representation was meant to be a pivot representation from which user-level services could be provided. As an illustration, the project included prototype modules for two such tools: coding into ICD9-CM[4] and query-answering,[5] both operating directly on the conceptual representation, without reference to or knowledge of the source language.

The need for methods to monitor prototype performance was identified from the design stage of the project. Effective tools were in place in the last project phase, and were used to assess the quality of the French and Dutch prototypes as they stood by project end.[6] The method used was inspired by the "MUC" evaluation methodology.[7] However, the type of representation and the target tasks presented specificities which required us to adapt this methodology in several respects.

This paper reports on this evaluation method and its application to the French MENELAS prototype (an evaluation of the Dutch MENELAS prototype is described by Spyns;[8] the English prototype is described in the MENELAS final report[6]). We first provide background on MENELAS and related MLP and conceptual systems, as well on the evaluation of natural language processing (NLP) systems. We then describe the MENELAS evaluation method. We summarize and comment the results of its application to the French MENELAS prototype. We then discuss the evaluation method itself, in relation with the framework proposed by Friedman and Hripcsak.[1]

BACKGROUND

MENELAS is an effort to produce a representation of natural language texts which has good informational and computational properties: a "normalized conceptual representation"[2] which supports detailed, compositional representation of medical information and inferences on it. This representation, based on Conceptual Graphs,[9] is very similar to the one promoted by the GALEN project.[10] More generally, there is a general movement of medical terminologies towards a more concept-oriented view, as was evidenced at the last IMIA WG6 Conference.[11] However, conceptual representations are complex to design;[12] they are also difficult to produce from natural language, and require methods[13,14] to bridge the gap between the flexibility of natural language and the strong constraints[15] on a normalized conceptual representation. MLP projects which aim at this class of representations include MENELAS[3], RECIT[16] and PARSETALK.[14]

As the field of NLP became more mature in the late 80's, and under the pressure of the US funding agencies, it started to examine more seriously the issues of evaluation. The catalyst was the series of "Message Understanding Conferences"[17] (MUC) sponsored by DARPA. Their organizers set up an evaluation protocol, as well as evaluation metrics borrowed from the field of information retrieval.[7] These have become a model for NLP evaluation in general, and the MENELAS evaluation in particular. MLP systems also more and more lend themselves to evaluation. Friedman and Hripcsak[1] have performed several evaluations of their MedLEE system, as well as a literature survey of MLP system evaluations. They propose a series of guide-

1091—8280/97/$5.00 © 1997 AMIA, Inc.

lines for principled evaluation of MLP systems, to which we return in the discussion.

MENELAS EVALUATION METHOD

We set up a *black box* evaluation method:[7] only the final, useful output of the system is examined: the data which is expected by the user, or which is to be used by further processes. In MENELAS, the processes which make use of the represention built by the analysis system are the Coding Service and the Consultation Service. Relevant tasks are those implemented by these services: code assignment and query-answering.

The data used for the query-answering evaluation are a corpus of PDSs, taken from the French MENELAS PDS corpus, and a questionnaire. The questionnaire contains questions collected from potential users, which they would ask from a system such as MENELAS. For each PDS in the test corpus, a *physician* provided answers to each query in the questionnaire. The principle of the test consists in having the *system* automatically analyze the same PDSs and automatically answer the same queries, and then in comparing automatically the system's replies with the expected answers. The corresponding task in terms of classical information retrieval would be to list the PDSs answering positively each question (find the patients with the required characteristics). A similar procedure is applied to evaluate the coding task, where queries are substituted with codes.

The questionnaire. 32 queries were selected, based on lists of cardiologists' queries collected in consortium sites. The formal representation of each query was prepared by a knowledge engineer. Two kinds of queries can be formulated. (*i*) `Yes_No` queries expect a positive or negative answer: *e.g.*, "Was a coronary angiography performed?", or "Was an exercise stress test performed under treatment?". They are however too limited (propositional logic) with respect to the power of the representation. Therefore, we designed a more powerful kind of evaluation query: (*ii*) `Which` queries expect a specific answer, in the form of a list of concept types which specialize the arguments of the query: *e.g.*, "What was the age of the patient?", or "Was there a lesion, which lesion was it, and on which artery?". Each of the 32 queries is represented by a conceptual graph (CG) including a gloss, such as:

```
Which_Query    LESION_LOC(*o,*x)
  :English "Was there a lesion, which and where?"
  [morpho_anomaly:*x ^tfocus]--(state_of)
  -->[physical_object:*o ^tfocus]
```

The test PDSs. For demonstrating MENELAS capacities at the end of the project, we selected a test corpus of 37 PDSs (7 of which had served to debug the system), representing 393 sentences of up to 96 words. All were written in the same cardiology department and during the same month. The lexicon was upgraded to include most of the words occurring in the corpus. The other knowledge bases were also extended to cover the concepts induced by these additional words.

Physician answers. Once the CG questionnaire was set up, for each PDS a physician answered the (glossed) queries, writing the answers on a paper form for this PDS. These answers were then transcribed into a answer file (one for each PDS) by a knowledge engineer. An example answer is:

```
Answer LESION_LOC is
  (interventricular_anterior_artery occlusion)
```

Physician codes. In the same way, the codes expected for each PDS were prepared by a physician. In order to take into account the variability of human coding, the same preparation was done by a second coder — a professional coder. The two sets of expected codes were entered in files, in a fairly straightforward format.

System answers and codes. The program's reply to a query is obtained by graph matching. The conceptual graph representing the query is "projected" onto the PDS representation built by the analysis system; projection is a standard CG operation[9] which matches identical (=) or more specific (<) graphs. If this is possible, the reply is the vector of concepts in the representation which match "focus" concepts in the query (or "yes" for `Yes_No` queries). If not, the reply is "no".

The answer comparison procedure. The goal of the answer comparison procedure is to check whether the replies computed by the program conform to the expected answers. For each query, we compare the expected answer (E) and the system replies (S). The comparison cases are the following:

similar	$S = E$	correct
more specialized	$S < E$	correct
more generic	$S > E$	wrong
not comparable	$S \neq E$	wrong
no reply	$\nexists S$	correct iff $\nexists E$

Code comparison goes along the same lines, except that the code comparison operation is equality.

Computation of recall and precision. The questionnaire test can be viewed as an information retrieval (IR) task. The task of IR, given a query, is to find the documents which satisfy this query. Typically, the query can be glossed as "find all texts which contain <*query*>," or "find all texts where <*query*> is true." `Yes_No` queries exactly fit this last formulation. `Which` queries extend this basic functionality, and could be rephrased as "find all texts where <*query*$(x_1 x_2 \ldots x_n)$> is true, and show the values found in each text for variables $x_1 x_2 \ldots x_n$". A traditional evaluation of IR systems is

given by computing *recall* (sensibility) and *precision* (specificity).

Recall and precision can be computed straightforwardly for Yes_No queries. The consideration of Which queries in fact must be decomposed into two parts. First, each Which query can be generically considered as another kind of Yes_No query, corresponding to the first part of the above gloss: "find all texts where $<query(x_1 x_2 \ldots x_n)>$ is true." This generic query is positively answered by all texts for which any specific answer exists, and negatively answered by all texts for which no specific answer exists. Second, each specific answer expected for a Which query can be considered as a more specific Yes_No query, where the variables of the query are instantiated. This specific query is positively answered as any other Yes_No query, and negatively answered by all texts for which this Yes_No query is false, in particular those for which only different answers to the same generic query exist.

MENELAS EVALUATION RESULTS

A formal evaluation of a system is really interesting on a final system. However, project resources were not sufficient to reach such a state: the final MENELAS prototype does not represent a full realization of the approach, and cannot be evaluated as such. Assessing its results nevertheless does have value as a tool to monitor progress as the system continues to be completed.

Results

The whole test was run once with the existing lexicons, knowledge bases, and questionnaire representation. Its global results were examined, in particular, the recall and precision figures for each code and query. This allowed us to identify the codes or questionnaire queries which were problematic for the system. The closer examination of actual system output around these codes and queries was performed only for the seven development PDSs. This allowed us to identify the origin of the problems, as relating to the representation of the queries or codes or to the representation built by the system, which itself can be caused by issues in the lexica, knowledge bases or analysis procedures.

Upgrade at this stage was kept minimal, and was performed only on system elements under the responsibility of our team: query representation, knowledge bases, pragmatic component and comparison procedures. Note that one fourth of the sentences in the corpus did not pass syntactico-semantic analysis (mainly because they were long and needed too much space or time to analyze); adding simple devices such as sentence chunkers would help improve results a lot.

Coding. Over the 37 PDSs of the test corpus, 17/39 codes were found by the system. According to either [CODER 1] or [CODER 2], 11/39 had a recall greater than 50 %. Tables 1 and 2 show the recall and precision obtained for each coder, sorted by recall, then precision.

Table 1: Codes with recall > 50% [CODER 1].

Code	E	S	Cor	Rec	Pre	Code gloss
92.05	3	3	3	100	100	Scintigraphy
88.57	34	36	34	100	94	Coronarography
89.52	6	18	5	83	27	ECG
250.0	5	4	4	80	100	Diabetes
272.9	4	4	3	75	75	Dyslipaemia
89.44	14	15	10	71	66	Exercise test
36.10	8	7	5	62	71	Aortocoro. bypass
413.9	22	15	13	59	86	Angina pectoris
305.1	9	7	5	55	71	Tobacco abuse
410.9	9	10	5	55	50	Acute MI
786.51	17	13	9	52	69	Precordial pain
Total	227	172	110	48	63	(avg. rec.&pre.)

Table 2: Codes with recall > 50% [CODER 2].

Code	E	S	Cor	Rec	Pre	Code gloss
88.57	34	36	34	100	94	Coronarography
250.0	3	4	3	100	75	Diabetes
427.31	1	2	1	100	50	Atrial fibrillation
92.05	1	3	1	100	33	Scintigraphy
410.9	1	10	1	100	10	Acute MI
786.51	1	13	1	100	7	Precordial pain
89.52	6	18	5	83	27	ECG
413.9	9	15	7	77	46	Angina pectoris
89.44	11	15	7	63	46	Exercise test
414.0	26	13	13	50	100	Arteriosclerosis
272.9	6	4	3	50	75	Dyslipaemia
Total	186	173	82	44	47	(avg. rec.&pre.)

Questionnaire. 22/30 generic queries obtained replies, and 15/30 a recall greater than 50 % (table 3). Table 4 illustrates specific answers, giving results for query Lesion. Five more answers were expected but not found (for aneurysm, atheromatosis, dissection, thrombosis).

Table 3: Generic queries with recall > 50%.

Query	E	S	Cor	Rec	Pre
Age	37	37	37	100	100
Sex	37	37	37	100	100
Ex._Test_Under_TT	2	2	2	100	100
Adm._Reason_State	30	34	30	100	88
Planned_Proc._Date	7	36	7	100	19
By_Pass	1	7	1	100	14
Coronarography	37	36	36	97	100
Coronarog._Date	27	36	26	96	72
Angioplasty	13	14	12	92	85
Angioplasty_Lesion	13	14	12	92	85
Lesion	30	30	24	80	80
Known_Coron._Dis.	15	23	11	73	47
Risk_Factor	21	14	14	66	100
Exercise_Test	12	15	7	58	46
Lesion_Loc	28	17	15	53	88
Total (avg. rec.&pre.)	429	369	285	66	77

Discussion of results

The overall results are of the same order as those ob-

Table 4: Specific lesions, recall > 50%.

Query	E	S	Cor	Rec	Pre
LESION=YES	30	30	24	80	80
LESION=stenosis	21	16	16	76	100
LESION=morpho_anom.	3	2	2	66	100
LESION=occlusion	11	7	7	63	100

tained by MUC competitors;[17] but the MUC evaluation was run on previously unseen texts, whereas MENELAS was upgraded before the test.

The two human codings display striking variants: 11 of the 17 codes found have different expected assignments, which results in a corresponding variation in the recall and precision measured against these 'gold standards'. For instance, code 305.1 (tobacco abuse) was assigned to 9 patients by [CODER 1] and to 5 patients by [CODER 2]; the system assigned it 7 times, all of which were correct according to the PDSs. However, they were not expected as a whole by either of the human coders. The precision score in such cases is therefore unduly decreased. Similar observations were made for codes 272.9 (Dyslipaemia) and 427.31 (Atrial fibrillation). This might explain why the coding precision values are generally lower than those obtained for the questionnaire test. This may also show that what must be coded is open to interpretation: all that a system such as MENELAS can do is to *propose* a set of potentially relevant codes, and let the human coder take the final decision. Nevertheless, these results point out the usefulness of such a system to help human coders improve coding consistency.

The real potential of the approach is already visible in the current prototype. A more detailed examination of the results shows that MENELAS can perform better than traditional IR methods on a variety of queries, such as *knowledge-based* queries and *complex* queries. Answering some queries requires knowledge which is expressed neither in the query itself, nor in the input PDS. For instance, asking for patient "risk factors" keeps two pieces of knowledge implicit: (*i*) in the present context, one refers to risk factors *for coronaropathies*, and (*ii*) the set of cardio-vascular risk factors. Recall and precision for the `Risk_Factor` query however show a good level: $R = 66$ and $P = 100$ for the generic query, while 6/7 specific queries range above 57 %, with 3 at 100 %.

Most queries look for more than just one atomic concept or expression: they rely on the co-specification of several pieces of information, linked with a specific relation. For instance, query `Lesion_Loc` asks for an instance of (a sub-class of) lesion (`morpho_anomaly`) and its localization. Another instance of 'complex' query is `Exercise_Test_Under_Treatment`, which asks for patients who (*i*) have had an exercise stress test, and who (*ii*) were under treatment during this test. In both cases, a detailed examination of input sentences evidences robustness to the variability of the input and sensitivity to meaningful syntactic markers such as negation.

DISCUSSION OF METHOD

Friedman and Hripcsak's collection of criteria[1] include 20 items in 5 groups. We discuss were the MENELAS evaluation method stands with respect to them.

Minimizing Bias. (1–6) Biases are important: the evaluation was designed, prepared and performed by the developer team; the test set could be seen by the evaluation team, and some (limited) system modification was performed during the test. This was deemed sufficient for a prototype only just assembled and was necessary given project resources.

Establishing a Reference Standard. (7) We made some attempt at assessing the variability of experts; (8) test set is large enough to distinguish levels of performance; (9) choice of reference standard was based on objectives of prototype — however, this is necessarily a reductionist approach for query-answering: the questionnaire only tests part of system output; (10) coders correspond to the two most usual coding situations: a clinician [CODER 1] and a professional coder [CODER 2]; the questionnaire was filled by a clinician helped by a cardiologist.

Describing the Evaluation Methods. (11–13) We attempted to provide information on the method, as space constraints allowed. More detail is provided in the MENELAS final report.[6]

Presenting Results. (14) Results pertain to the complete test set, including the sentences that could not be processed; (15) raw data is given on inter-coder agreement; (16) no confidence interval was computed.

Discussing Conclusions. (17) Limitations of the study are discussed (here); (18) some implications for the possible use of such a system are mentioned; (19) there is only moderate overgeneralization of the results; (20) nearly no analysis of failures is presented.

We agree with Friedman and Hripcsak's guidelines as an ultimate, ideal goal, necessary for comparative evaluation of systems and to assess their technical usability in a clinical setting. When we designed this evaluation, we considered a large part of the above criteria, but had to drop some because the induced workload was too heavy. This is especially true of bias minimization. Nevertheless, our main interest at the prototype stage was to have an automated procedure for monitoring performance during development, as the lexicon and knowledge bases were extended and as system mod-

ules were added or improved: biases are much less relevant in this context. To our knowledge, this is the only instance of a formal evaluation of an MLP system for producing a normalized conceptual representation where scoring was performed automatically rather than by a posteriori expert judgment.

The main workload in the MENELAS test came from (*i*) the design, implementation and tuning of the evaluation procedure, (*ii*) the preparation of the test data, and (*iii*) the upgrade and debugging of the prototype during the test. (*i*) is hopefully performed only once. (*iii*) belongs to prototype development work, the results of which the evaluation procedure precisely aims at monitoring. Test data (*ii*) can be reused several times, but new test data need to be prepared when system performance stabilizes on it. One way to save evaluation cost and to minimize bias would be to share test data. This seems easy for codes; queries and answers should be sharable across systems at least in their external (glossed) forms. Sharing test corpora augmented with expected results[18] (codes, query answers) would distribute some of the test set development effort. By applying test sets prepared by other teams, one would at the same time reduce bias.

An effort is still necessary to convert external test specifications such as natural language test queries into formal representations (here, CGs) usable by the automated test procedure. This is a delicate operation, which requires a good knowledge of the formal representation. A knowledge engineer performed this task in our experiment. An alternative would be to analyze these natural language descriptions, using the same NL processor, into the target formal representations.

Acknowledgments

This work has been partly supported by the European Union project MENELAS (AIM 2023). We acknowledge the work of all our consortium partners, in particular Dr. F Volot, who gave a major contribution in the preparation of the test set.

References

1. Friedman C and Hripcsak G. Evaluating natural language processors in the clinical domain. In: Chute [11], 1997:41–52.
2. Zweigenbaum P, Bachimont B, Bouaud J, Charlet J, and Boisvieux JF. A multi-lingual architecture for building a normalised conceptual representation from medical language. In: Gardner [19], 1995.
3. Zweigenbaum P and Consortium MENELAS . MENELAS: an access system for medical records using natural language. *Comput Meth Prog Biomed* 1994;45:117–20.
4. Delamarre D, Burgun A, Seka LP, and Le Beux P. Automated coding system of patient discharge summaries using conceptual graphs. *Meth Inform Med* 1995;34:345–51.
5. Nangle B and Keane MT. Effective retrieval in hospital information systems. *Artif Intell Med* 1994;6(3):207–28.
6. Zweigenbaum P, Bachimont B, Bouaud J, et al. MENELAS final report. Deliverable report AIM-MENELAS 17, DIAM-SIM/INSERM U.194, 1995.
7. Lehnert W and Sundheim B. A performance evaluation of text-analysis technologies. *The Artif Intell Mag* 1991(Fall):81–94.
8. Spyns P and De Moor G. A Dutch medical language processor: Part II: Evaluation. *J Bio-Medical Engineering* 1997. In preparation.
9. Sowa JF. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, London, 1984.
10. Rector AL, Bechhover S, Goble CA, et al. The GRAIL concept modelling language for medical terminology. *Artif Intell Med* 1997;9(2):139–71.
11. Chute CC, ed. *Proc Conference on Natural Language and Medical Concept Representation*, Jacksonville, Fl, 1997. IMIA WG6.
12. Zweigenbaum P, Bachimont B, Bouaud J, Charlet J, and Boisvieux JF. Issues in the structuring and acquisition of an ontology for medical language understanding. *Meth Inform Med* 1995;34(1/2):15–24.
13. Bouaud J, Bachimont B, and Zweigenbaum P. Processing metonymy: a domain-model heuristic graph traversal approach. In: Tsujii JI, ed, Proc 16 th COLING, Copenhagen, Denmark. 5–9 August 1996:137–42.
14. Hahn U, Schnattinger K, and Romacker M. Automatic knowledge acquisition from medical texts. In: Cimino JJ, ed, Proc AMIA Annual Fall Symposium, Washington DC. AMIA (JAMIA supplement), 1996:383–7.
15. Rector A. Thesauri and formal classifications: Terminologies for people and machines. In: Chute [11], 1997:183–95.
16. Rassinoux AM, Wagner JC, Lovis C, et al. Analysis of medical texts based on a sound medical model. In: Gardner [19], 1995:27–31.
17. Advanced Research Projects Agency. *Fifth Message Understanding Conference (MUC-5)*, San Francisco, Ca, August 1993. Morgan Kaufmann.
18. Zweigenbaum P, Bouaud J, Bachimont B, et al. From text to knowledge: a unifying document-oriented view of analyzed medical language. In: Chute [11], 1997:21–30.
19. Gardner RM, ed. *Proc Nineteenth Annu Symp Computer Applications in Medical Care*, New Orleans, 1995. AMIA (JAMIA supplement).