Available online at www.sciencedirect.com
ScienceDirect
Journal of Biomedical Informatics 39 (2006) 589–599
Journal of Biomedical Informatics
www.elsevier.com/locate/yjbin

Natural language processing to extract medical problems from electronic clinical documents: Performance evaluation

Stéphane Meystre *, Peter J. Haug

Department of Medical Informatics, University of Utah School of Medicine, Salt Lake City, UT, USA

Received 3 June 2005
Available online 5 December 2005

Abstract

In this study, we evaluate the performance of a Natural Language Processing (NLP) application designed to extract medical problems from narrative text clinical documents. The documents come from a patient's electronic medical record and medical problems are proposed for inclusion in the patient's electronic problem list. This application has been developed to help maintain the problem list and make it more accurate, complete, and up-to-date. The NLP part of this system—analyzed in this study—uses the UMLS MetaMap Transfer (MMTx) application and a negation detection algorithm called NegEx to extract 80 different medical problems selected for their frequency of use in our institution. When using MMTx with its default data set, we measured a recall of 0.74 and a precision of 0.756. A custom data subset for MMTx was created, making it faster and significantly improving the recall to 0.896 with a non-significant reduction in precision.

Keywords: Medical Records; Problem oriented; Natural Language Processing; MetaMap Transfer; NegEx; Program evaluation

1. Introduction

The problem list is an important piece of the problem-oriented medical record. It centralizes all patients' medical problems in a concise view, facilitates associating clinical information in the record to a specific medical problem, and encourages an orderly process of clinical problem solving and clinical judgment. The problem list in a problem-oriented patient record also provides a context in which continuity of care is supported, preventing both redundant and repeated actions [1]. To serve the functions it is designed for, the problem list must be as accurate, complete, and timely as possible. In our institution, this document is currently usually neither complete nor accurate, and may be totally unused, especially in the inpatient domain. To address this deficiency, we developed an application using Natural Language Processing (NLP) to harvest potential problem list entries from the multiple free-text electronic documents available in our Electronic Health Record (EHR). The problems detected in the clinical documents are proposed to the physicians for addition to the official problem list. Development of an NLP system of sufficient accuracy will allow us to pursue the global aim of our project, to automate the process of creating and maintaining a problem list for hospitalized patients, and thereby to help guarantee the timeliness, accuracy, and completeness of this information.

2. Background

To supply many key advantages of an electronic problem list, the entries must also be coded. Coded data are classified and standardized, facilitating storage and retrieval, clinical research, and administrative functions like billing, and are also desirable to enable exchange and sharing of data [2]. Medical vocabularies

* Corresponding author. Fax: +1 801 581 4297.
E-mail address: s.meystre@utah.edu (S. Meystre).

1532-0464/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jbi.2005.11.004

used in problem lists are numerous, ranging from ICD-9-CM [3], to SNOMED [4], the Unified Medical Language System (UMLS®) [5–7], as well as locally developed vocabularies [8]. Coding of medical problems may be achieved by manually assigning a code when the problem is entered, or by using NLP techniques to map free-text problem list entries with an appropriate code. The former method is usually facilitated by the use of pick lists or search engines [9]. Both of these features are available in the application used to maintain our institution's electronic problem list. NLP techniques promise coded data while allowing the use of natural language, still the most user-friendly and expressive way of recording information, and give the advantages of coded data.

Techniques for automatically encoding textual documents from the medical record have been evaluated by several groups. Examples are the Linguistic String Project [10,11], and MedLEE (Medical Language Extraction and Encoding system) [12]. MedLEE has been recently adapted to extract UMLS concepts from medical text documents, achieving 83% recall and 89% precision [13]. Other systems automatically mapping clinical text concepts to a standardized vocabulary have been reported, like MetaMap [14,15], IndexFinder [16], and KnowledgeMap [17]. MetaMap and its Java version called MMTx (MetaMap Transfer) were developed by the US National Library of Medicine (NLM). They are used to index text or to map concepts in the analyzed text with UMLS concepts. MetaMap has been shown to identify most concepts present in MEDLINE titles [18]. MetaMap has been used for Information Retrieval [19–21], for Information Extraction in biomedical text [18,22], and to extract different types of information like anatomical concepts [23] or molecular binding concepts [24]. MetaMap has also been used with patient's electronic messages to automatically provide relevant health information to the patients [25]. Finally, in a study by Shadow and McDonald [26] the system extracted the most critical pathology findings in documents.

Independent negation detection is required when using MMTx, which does not discriminate between present and absent concepts. In the medical domain, negation detection is crucial due to the fact that findings and diseases are often described as absent. A few negation detection algorithms have been developed, like NegEx, a computationally simple algorithm using regular expressions [27], or the more complex general-purpose Negfinder [28]. These algorithms have been evaluated and have shown good results. NegEx has demonstrated a specificity of 94.5% and a sensitivity of 77.8% [27], and Negfinder a sensitivity of 95.3% and a specificity of 97.7% [28]. Both negation detection algorithms described above and our Automated Problem List system use regular expressions for part of the document processing. Regular expressions are well described in a variety of publications like Jeffrey Friedl's book [29].

3. Materials and methods

3.1. Automated Problem List system

Our Automated Problem List system [30] uses NLP technologies to extract potential medical problems from free-text medical documents. This system is made of two main components: a background application and the problem list management application. The background application does all the text processing and analysis, and stores extracted medical problems in the central clinical database. These problems can then be accessed by the problem list management application integrated in our Electronic Health Record. Here, we describe an evaluation of the background application responsible for processing the medical documents and detecting problems. The ultimate success of our efforts to assist in maintaining the problem list is dependent on the accuracy of this application.

The initial version of the background application was designed to function with a limited list of 80 different medical problems. These were a group of diagnoses that were selected based on their frequency of use in a field of evaluation focused in the area of cardiovascular medicine and surgery. Some of the problems were high level diagnoses, like *Arrhythmia* or *Ischemic heart disease*; some were more specific diagnoses like *Mitral stenosis* or *Left bundle branch block*; and finally some problems could be considered observations, like *Wheeze* or *Pain*.

3.2. Documents processing and analysis

3.2.1. Documents pre-processing

As described in Fig. 1 and in details in another publication [30], the background application begins the processing of documents by detecting sections and sentences, using regular expressions and a set of rules. Before passing sentences to the NLP module, some disambiguation was required. Our NLP module uses MMTx (version 2.3.C). MMTx was originally developed to analyze MEDLINE abstracts, but now accommodates any type of text. Acronyms seem to be less common in paper abstracts than in clinical documents, and are the principal source of ambiguity for our system. Examples of acronyms ambiguous to MMTx are "Mr." (detected as mitral regurgitation), "M.D." (mental depression), "PA" (pernicious anemia), etc. To detect these ambiguous acronyms, the output of the NLP analysis of 40 randomly selected documents (from the same study population cited above) was examined by the first author. Disambiguation then consisted in replacing these acronyms with their full name. After preprocessing for disambiguation, sentences with contextual information (like headers indicating document type and section type) are passed to the NLP module for analysis.

3.2.2. Medical problems detection

The NLP module actually works in two steps as shown in Fig. 1: a first step using MMTx to extract each potential

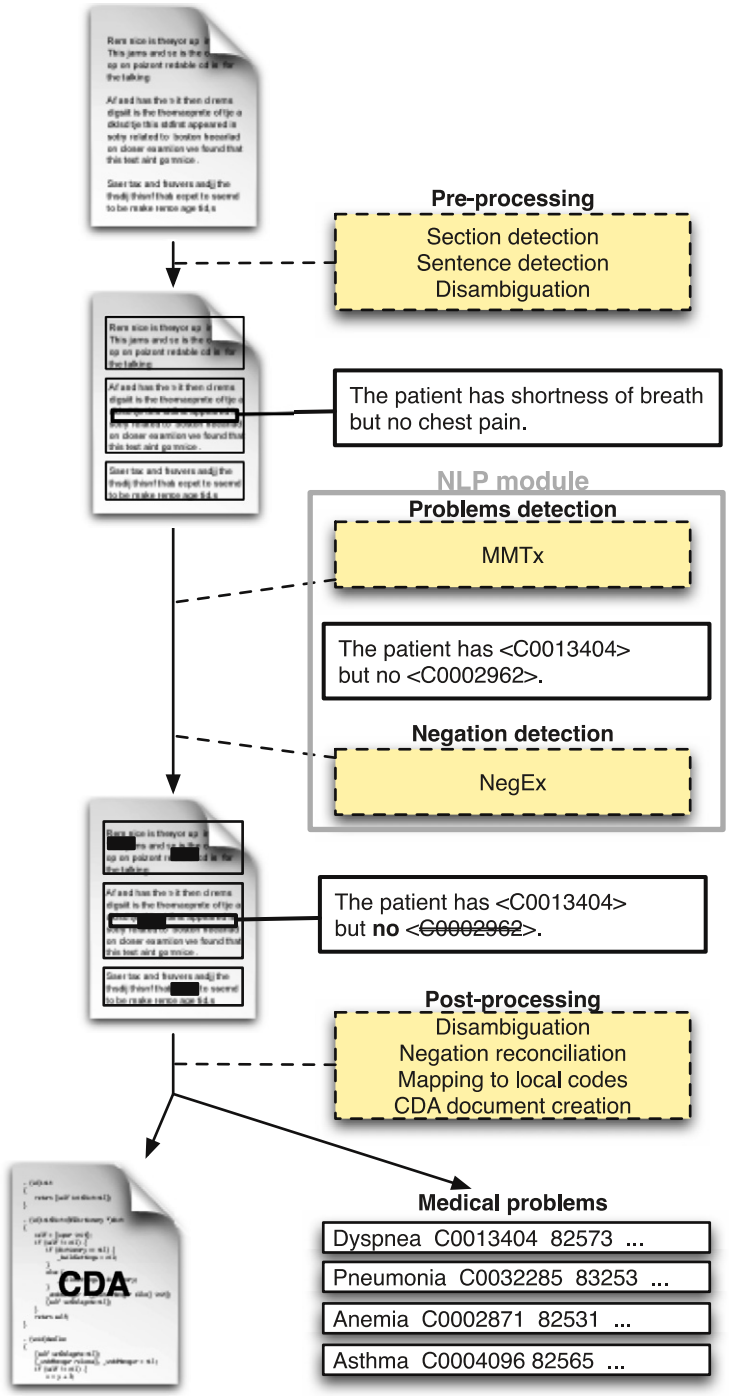

Fig. 1. Documents review process with an example sentence.

medical problem and a second step to infer the state of each of those problems. Since we are only interested in 80 different medical problems, and not in the whole UMLS Metathesaurus content (UMLS version 2004AA contains over 1 million concepts [31]), we created a subset of the Metathesaurus adapted to our system. The selection process resulted in the reduction to about 0.25% of the original data set, from more than a million to about 2500 concepts. This reduction made the NLP module more than three times faster, and also increased its recall but reduced its precision. This issue was related to the absence of numerous concepts in our data subset, concepts used by MMTx for its own disambiguation. To prepare this version of MMTx with reduced scope, we used MetamorphoSys [32], an application provided with the UMLS. MetamorphoSys allows filtering the Metathesaurus, based on source vocabularies, semantic types, and other filters. We selected all "level 0" vocabularies (only the UMLS license is needed) and SNOMED, and included only the semantic types of the medical problems we were extracting. To subset the

set of recognized concepts further, we loaded the resulting MRCON, MRSO, and MRSTY tables into a MySQL database for subsequent processing. A mapping table was manually built to link the 80 selected concepts with all related subconcepts (e.g., *Right Bundle Branch Block* was mapped to *Incomplete Right Bundle Branch Block*, *Complete Right Bundle Branch Block*, and *Other or unspecified Right Bundle Branch Block*). This table was built by adding all UMLS concepts with a CHD (Child), SIB (Sibling), RN (Narrower), and some concepts with RL (Similar) and RQ (Related) relationships with the 80 concepts of our list of medical problems. These relationships are defined in the UMLS Semantic Network. Manual review was then done by the first author to remove irrelevant concepts (e.g., for *Thrombocytopenia*, a sibling like *Thrombocytosis* was removed, and a narrower concept like *Thrombocythemia, hemorrhagic* was also removed). This manually built table was used to select relevant concepts in the UMLS tables cited above. The final step was the creation of the MMTx data files. A tool called MMTx Data File Builder [33] is provided with MMTx and allows the creation of these files from UMLS Metathesaurus subsets or custom-made data sets. MMTx Data File Builder was used with strict model filtering.

3.2.3. Negation detection

As mentioned earlier, MMTx lacks negation detection. For example, "diabetes" is detected in the sentence "The patient is known for diabetes mellitus, treated with insulin injections," but also in the sentence "No diabetes is reported in the patient's history." We therefore had to add the ability to recognize negation. We adapted an algorithm described above and called NegEx, and implemented it in Java. We used the improved version of NegEx, called NegEx 2 [34], with some negation terms added.

3.2.4. Documents post-processing

This last step starts with some disambiguation. For example, when *heart arrest* was detected in a sentence, and *cardioplegia* or *cardioplegic* was present in the same sentence, then the detected *heart arrest* was considered absent (i.e., not a real problem but a voluntary and controlled action to allow heart surgery). Negation reconciliation as described in [30] followed, with mapping to local codes. MMTx extracts UMLS concepts from text, but our Electronic Health Record uses another terminology (Health Data Dictionary [35]). Mapping UMLS and our local terminology was therefore required. Our local terminology already provides mapping with UMLS concepts, but this mapping is incomplete (some of our 80 targeted problems lacked this mapping). We therefore had to add the local code corresponding to each UMLS concept in the mapping table mentioned above. The extracted medical problems and a transformed XML version of the document were then stored in a local database. The transformed version of the document follows the CDA (HL7 Clinical Document Architecture) [36] standard and uses a previously described information model [37]. This CDA version of the document is needed for the application used by physicians to update and manage the electronic problem list.

3.3. Study design

3.3.1. Evaluation goals

Five questions were answered in this evaluation. Can MMTx be improved for our specific use? This first question was answered by comparing the two versions of MMTx (default data set and custom data subset). The other four questions were related to review methodologies. Is there a difference when reviewing documents with an electronic application or on paper? Does the performance of reviewers change when the number of problems to detect changes? Can techniques like NLP or reading a same document twice improve reviewers' performance? How do NLP and human reviewers compare when analyzing the same documents?

3.3.2. Reference standard creation

To evaluate the accuracy of the NLP tools created to detect problems, a reference standard was created with a chart review. We randomly selected 160 clinical documents from a study population sampled using the following criteria: adult inpatients in a cardiovascular unit of the LDS Hospital (Salt Lake City, Utah) during the year 2002, with lengths of stay of at least 48 h, and with at least one discharge diagnosis in the list of the 80 selected diagnosis. The clinical documents included were from a variety of different types, as listed in Table 1. Our system processed each document twice: once with the default data set (data set provided with MMTx) and once with our customized concept subset.

Two independent physicians reviewed each electronic document using a web-based review application. When the two reviewers disagreed, a third physician determined the presence or absence of the disputed medical problem. A medical problem was considered present if mentioned in the text as probable or certain in the present or the past (e.g., "the patient has asthma"; "past history positive for asthma"; "pulmonary edema is probable"), and considered

Table 1
Types of clinical documents in the test set

Type of document	Instances
Death summaries	3
Diagnostic procedure reports	33
Pathology reports	10
Radiology reports	60
Emergency Department reports	13
Consultation reports	6
Surgery reports	9
History and Physicals	15
Progress notes	2
Discharge summaries	9
Total	160

absent if negated in the text or not mentioned at all (e.g., "this test excluded diabetes…"; "he denies any asthma"). To improve inter-rater reliability, reviewers were trained and tested on selected sample cases before the formal review, and were provided a set of standardized instructions. We also used a medical record review technique called *explicit review*. This approach involves directing the reviewers to look at specific concepts (our list of selected medical problems) on which judgment is to be based [38]. *Explicit review* is associated with higher inter-rater reliability than *implicit review*, where reviewers use only their knowledge or beliefs to make judgments. The focusing process described was achieved by displaying the document to review beside a list of the 80 targeted medical problems (Fig. 2). Reviewers checked the medical problems they considered present in the document and submitted this pre-review. To improve the quality of the review, a NLP-assisted methodology was used as shown in Fig. 3. Results of the pre-review were compared in real-time with the results of the NLP module, which had already been run on each document. Reviewers were then asked whether they wanted to keep a problem that was not found by NLP, or add a problem that had been found by NLP only. To help the reviewer understand how the system had identified a specific concept, the document was displayed with the sentences containing the recognized problems highlighted in red (Fig. 4). After eventually selecting the problems they wanted to keep or add, reviewers finally submitted the refined review.

3.3.3. Measurements

Eight standard measures were used to describe the accuracy of this Natural Language Processing system. The first two are the most common: precision (equivalent to positive predictive value; Eq. (1)) and recall (equivalent to sensitivity or true positive rate; Eq. (2)). In our case, the concepts were elements of the set of medical problems chosen for this project. Another typical value combining precision and recall—the F-measure (Eq. (3))—was also calculated. In calculating the F-measure, a β value of 1 gives equal weight to precision and recall, and a value higher than 1 gives more weight to the recall. Other measures were also calculated, like overgeneration (equals 1-precision; Eq. (4)), undergeneration (equals 1-recall; Eq. (5)), error (Eq. (6)), accuracy (Eq. (7)), and fallout (equivalent to false positive rate here; Eq. (8)). To calculate these values, problems were counted and categorized as true positive (TP; concept present in the document and found by NLP), false positive (FP; concept found by NLP but absent from the document), false negative (FN;

Fig. 2. Screenshot of the web-based review application, before submitting the pre-review.

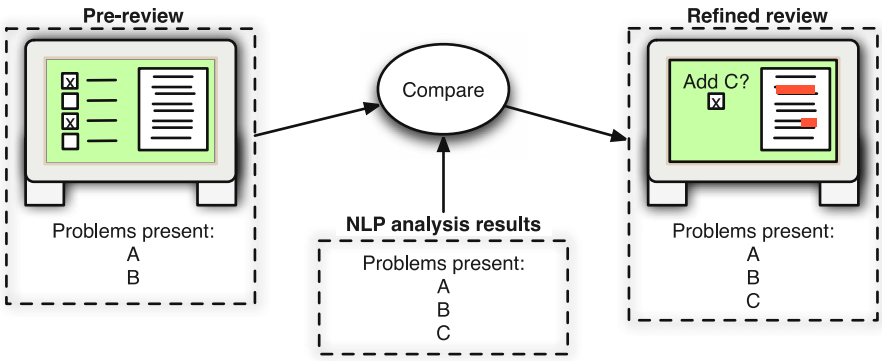

Fig. 3. NLP-assisted review methodology.

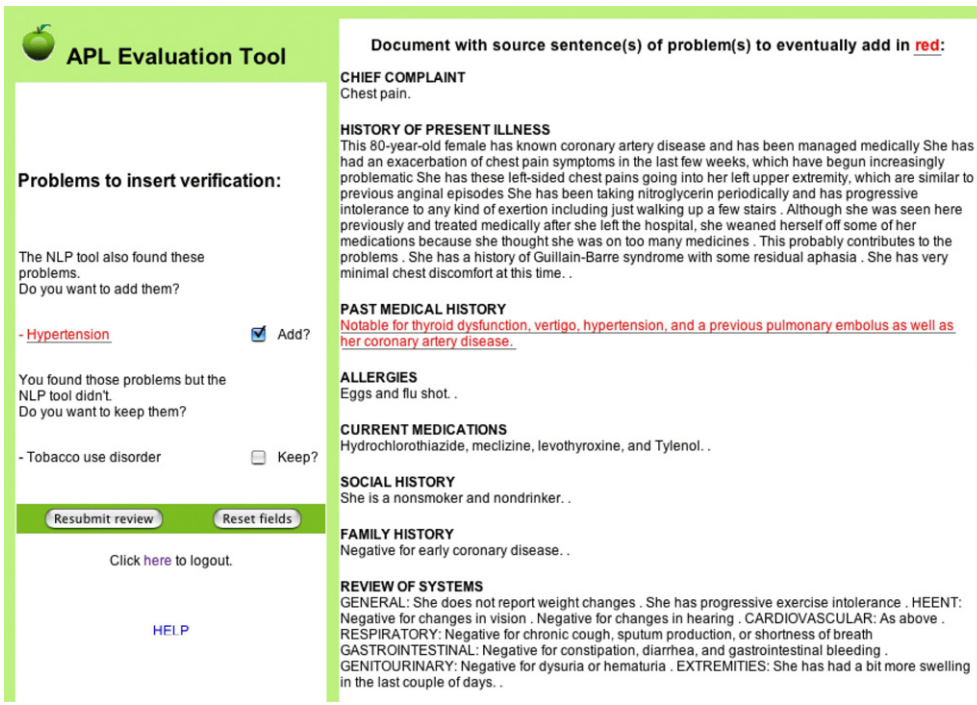

Fig. 4. Screenshot of the web-based review application, before submitting the refined review (the text originally displayed in red is underlined in this grayscale figure). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.)

concept present in the document but not found by NLP), or true negative (TN; concept absent from the document and not found by NLP):

$$\text{precision} = \text{TP}/(\text{TP} + \text{FP}), \quad (1)$$

$$\text{recall} = \text{TP}/(\text{TP} + \text{FN}), \quad (2)$$

$$F\text{-measure} = ((\beta^2 + 1)PR)/((\beta^2 P) + R), \quad (3)$$

$$\text{overgeneration} = \text{FP}/(\text{TP} + \text{FP}), \quad (4)$$

$$\text{undergeneration} = \text{FN}/(\text{TP} + \text{FN}), \quad (5)$$

$$\text{error} = (\text{FN} + \text{FP})/(\text{TP} + \text{FN} + \text{FP}), \quad (6)$$

$$\text{accuracy} = (\text{TP} + \text{TN})/(\text{TP} + \text{FN} + \text{FP} + \text{TN}), \quad (7)$$

$$\text{fallout} = \text{FP}/(\text{FP} + \text{TN}). \quad (8)$$

Agreement between reviewers was measured using Cohen's κ and Finn's R. The latter was more relevant because of a strong concentration on marginals in our case (numerous TN). Statistical analysis was based on the Mann–Whitney U statistic for non-normality reasons.

3.3.4. Review methodologies evaluation

A subset of 80 documents from our 160 documents test set was used. The different configurations tested (number of documents analyzed and medical problems to detect) are listed in Table 2. In all configurations, each document was reviewed by two reviewers and a third if they disagreed. Each reviewer read a certain document only once. To approximate the effect of reading a same document twice, the two reviews of each document were combined. For the paper-based review, reviewers received a printed checklist of the 80 target medical problems attached to each document to review. To evaluate the effect of the num-

Table 2
Review configurations

Review method	Set of documents analyzed	Documents reviewed by each reviewer
Web-based application, looking for 80 problems	160 documents	40
Web-based application, looking for 20 problems	80 documents	20
Web-based application, looking for 40 problems	80 documents	20
Paper-based, looking for 80 problems	80 documents	20

ber of problems to detect, the reviewers used the web-based review application with the same subset of 80 documents, looking for only 20 or 40 problems of the 80 targeted problems. To compare NLP and human reviewers, we compared the analysis of these 80 documents by the version of MMTx using the custom data subset, and the review by humans of the same documents printed on paper.

4. Results

4.1. Documents review

Eight different physician reviewers participated in the review process to create the reference standard. Five were board-certified physicians and three were residents with two or more years of training. With the web-based application described above, reviewers spent between 93 and 189 s per document. Reviewers' overall agreement was almost perfect, with a Cohen's κ of 0.9 and a Finn's R of 0.985 when building the reference standard. This latter value is more representative of the actual agreement, since the agreement table was strongly skewed, with far more true negatives than true positives. Agreement was also almost perfect in the other review configurations, with no statistically significant difference. For the paper-based review and the review with only 40 problems, Finn's R was 0.978, and for the review with 20 problems, Finn's R was 0.989. When analyzing agreement for the individual medical problems separately, few medical problems had a Finn's R below the 95% confidence interval, and the lowest agreement was found with *Angina* (Finn's R of 0.9) and *Pain* (Finn's R 0.875), related to the common and ambiguous "chest pain" phrase, possibly indicating *angina* but also *pain*. We feel that the results of the physician review give us a reasonable "gold standard" against which to compare the NLP application.

4.2. NLP accuracy

For the output of the NLP system, recall, precision, undergeneration, overgeneration, error, accuracy, and fallout were measured and mean and 0.95 confidence intervals were computed. The F-measure was calculated with a β value of 1 (same weight for recall and precision), and a β value of 2, to give more importance to the recall, the most important feature for our system (Tables 3 and 4). Indeed, our aim is to detect as many of the medical problems that are described in free-text documents as possible. Recall should be more important than precision in this context.

The recall and precision results are represented on a recall–precision graph with mean values and 95% confidence intervals (Fig. 5). The true positive rate (equivalent to recall here)–false positive rate (equivalent to fallout here) graph (Fig. 6)—also known as ROC graph (Receiver Operating Characteristic)—gives a result very similar to the recall–precision graph (Fig. 5).

Statistical analysis of the NLP results showed that the recall was significantly higher (two-tailed $p < 0.0001$) for the customized subset than the default data set. Precision and fallout were not significantly different (precision two-tailed $p = 0.0797$; fallout two-tailed $p = 0.0665$).

4.3. Analysis of review methodologies

Recall and precision results are represented on a recall–precision graph with mean values and 95% confidence intervals (Fig. 7). When evaluating the difference between the reviews using the web-based review application and the paper-based review, the fallout was significantly lower for the web-based review without the help of the NLP analysis than for the paper-based one (two-tailed $p = 0.043$). Precision and recall were not significantly different (two-

Table 3
NLP module evaluation and web-based or paper-based review results with means and 95% confidence intervals

Measurements	Default data set	Custom subset	Web-based review	Paper-based review
Recall	0.74 (0.68–0.799)	0.896 (0.854–0.939)	0.788 (0.748–0.827)	0.796 (0.736–0.857)
Precision	0.756 (0.694–0.819)	0.691 (0.63–0.752)	0.912 (0.883–0.94)	0.856 (0.799–0.912)
F-measure ($\beta = 2$)	0.743	0.846	0.81	0.807
F-measure ($\beta = 1$)	0.748	0.780	0.845	0.825
Undergeneration	0.26 (0.2–0.319)	0.104 (0.061–0.146)	0.212 (0.173–0.251)	0.204 (0.143–0.264)
Overgeneration	0.244 (0.181–0.306)	0.309 (0.248–0.37)	0.088 (0.06–0.116)	0.144 (0.088–0.2)
Error	0.391 (0.33–0.452)	0.344 (0.284–0.404)	0.274 (0.233–0.315)	0.286 (0.222–0.351)
Accuracy	0.979 (0.975–0.983)	0.976 (0.972–0.981)	0.993 (0.992–0.994)	0.987 (0.984–0.99)
Fallout	0.017 (0.014–0.021)	0.013 (0.009–0.016)	0.001 (0.001–0.002)	0.004 (0.003–0.005)

Table 4
Results comparing configurations with 20, 40, or 80 problems to detect, and reading them twice or having the help of NLP, with means and 95% confidence intervals

Measurements	20 problems	20 problems read twice	20 problems with NLP	40 problems	40 problems read twice	40 problems with NLP	80 problems	80 problems read twice
Recall	0.872 (0.789–0.955)	0.926 (0.82–1)	0.918 (0.845–0.992)	0.803 (0.733–0.873)	0.908 (0.828–0.988)	0.921 (0.87–0.971)	0.788 (0.748–0.827)	0.913 (0.871–0.954)
Precision	0.904 (0.831–0.977)	0.839 (0.709–0.969)	0.924 (0.856–0.993)	0.855 (0.788–0.922)	0.761 (0.654–0.868)	0.868 (0.808–0.928)	0.912 (0.883–0.94)	0.859 (0.812–0.907)
F-measure ($\beta = 2$)	0.878	0.907	0.919	0.813	0.874	0.91	0.81	0.902
F-measure ($\beta = 1$)	0.888	0.88	0.921	0.828	0.828	0.894	0.845	0.888
Undergeneration	0.128 (0.044–0.211)	0.074 (0–0.18)	0.081 (0.008–0.155)	0.197 (0.127–0.267)	0.092 (0.012–0.172)	0.079 (0.028–0.13)	0.212 (0.173–0.251)	0.087 (0.046–0.129)
Overgeneration	0.096 (0.023–0.169)	0.161 (0.031–0.29)	0.075 (0.007–0.144)	0.145 (0.078–0.212)	0.239 (0.132–0.346)	0.132 (0.072–0.192)	0.088 (0.06–0.116)	0.141 (0.093–0.188)
Error	0.209 (0.113–0.305)	0.217 (0.072–0.361)	0.147 (0.057–0.237)	0.292 (0.213–0.37)	0.294 (0.186–0.402)	0.186 (0.118–0.253)	0.274 (0.233–0.315)	0.211 (0.156–0.266)
Accuracy	0.996 (0.994–0.998)	0.999 (0.998–1)	0.998 (0.996–0.999)	0.991 (0.988–0.993)	0.995 (0.994–0.997)	0.994 (0.992–0.996)	0.993 (0.992–0.994)	0.997 (0.996–0.997)
Fallout	0.001 (0–0.002)	0.001 (0–0.002)	0.001 (0–0.002)	0.003 (0.002–0.005)	0.003 (0.002–0.005)	0.004 (0.002–0.005)	0.001 (0.001–0.002)	0.002 (0.001–0.003)

tailed $p = 0.2327$ for the precision; two-tailed $p = 0.3433$ for the recall).

When NLP analysis was added to the first step of the review with the web-based application, the recall increased significantly (two-tailed $p = 0.0074$ when comparing the 40 problems review with and without NLP), but the precision and the fallout were not significantly different. Comparing the first step of the review with a second review of the same document (i.e., read once vs. read twice) showed a significantly higher recall when reading a document twice (two-tailed $p < 0.0001$ with the 80 problems review), but also a lower precision (two-tailed $p = 0.0297$ with the 80 problems review).

The same tendency was observed with the review for 40 or 20 problems, but confidence intervals were too large to show a significant difference. No significant difference in precision or fallout was found between the reviews looking for 80, 40, or only 20 problems. Recall was higher (two-tailed $p = 0.0017$) when looking for only 20 problems rather than 80 problems.

Finally, when comparing the performance of humans and NLP, we found a significantly higher precision (two-tailed $p < 0.0001$) and a significantly lower fallout (two-tailed $p < 0.0001$) with humans than with NLP, but the recall with humans was significantly lower than with NLP (two-tailed $p < 0.0001$).

5. Discussion

This study showed that our system using MMTx with a custom data set and negation detection has good recall and satisfying precision, both at a level that fulfills our requirements for the NLP module of our Automated Problem List system in a clinical setting. These requirements were based on a consensus with some future clinical users of our system and was a recall of about 90% or higher and a precision of about 60% or higher. This study also showed that an MMTx-based NLP system with a customized subset had a higher recall than with the default MMTx data set. It showed that a paper-based review had higher fallout than a review using an electronic review application, but also that adding the help of NLP when doing reviews with the web-based application increased the recall. When comparing simple reviews with double readings of the same document, the recall was higher but the precision was lower when reading a same document twice. Reviews looking for fewer medical problems had a higher recall. And, finally, human reviewers had a lower recall, but higher precision than NLP.

5.1. Agreement between reviewers

The excellent inter-reviewers agreement allows a reference standard of good quality, therefore giving accurate results. Agreement was also high in all configurations of our review, without significant differences. This was made possible by the use of explicit review techniques, always

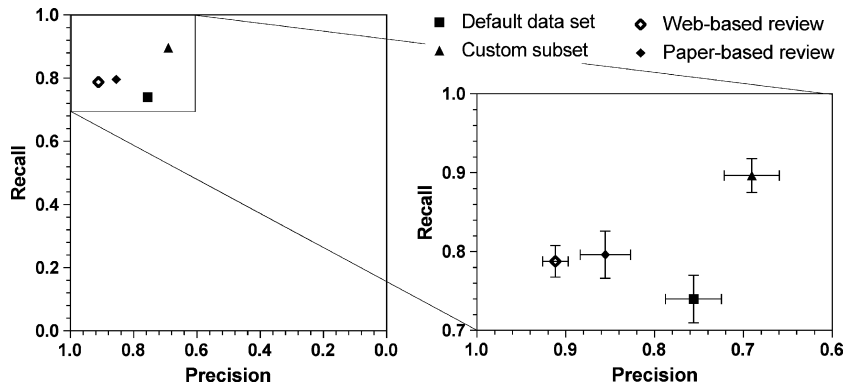

Fig. 5. Graphical display of recall and precision when detecting 80 medical problems, with mean values and 95% confidence intervals.

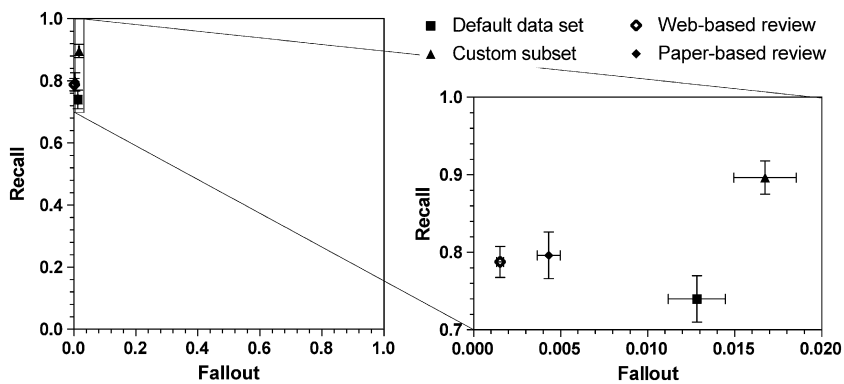

Fig. 6. Graphical display of recall and fallout when detecting 80 medical problems, with mean values and 95% confidence intervals.

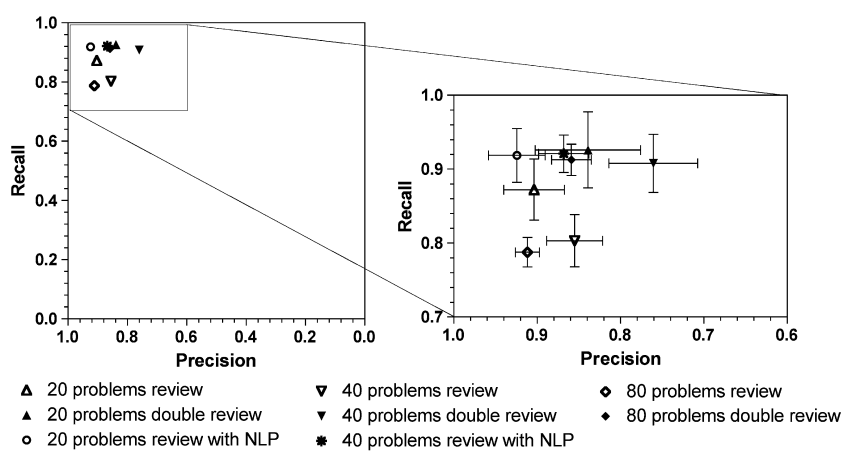

Fig. 7. Graphical display of recall and precision when reviewers detect 20, 40, or 80 medical problems, with mean values and 95% confidence intervals.

showing the list of problems to look for beside the document to review.

5.2. Results comparison with other similar studies

Our results compare favorably with another evaluation of MMTx, where a recall of 53% was reported, even if this result has to be considered cautiously because of small sample size and other reasons [39]. Our system only extracted a limited set of concepts, and all children of those concepts were matched to the parent ones, therefore improving the recall. Other NLP systems extracting UMLS concepts from free-text have reported evaluations. For example MetaMap demonstrated an exact-match recall of 52.8% and partial-match recall of 93.3%, and exact-match precision of 27.7% and partial-match precision of 55.2% [18]. This study evaluated detection of all biomedical concepts in title phrases. MedLEE has recently been evaluated when extracting UMLS concepts from medical text documents, achieving 83% recall and 89% precision [13].

5.3. Errors analysis

When analyzing the errors made by our system, we found a number of negation detection errors, usually in long sentences with a list of negated concepts (e.g., in "patient history negative for diabetes, myocardial infarct, pulmonary embolism, arrhythmia, and varicose veins," all concepts would be detected as absent except *varicose veins* that would be detected as present). This problem is related to the algorithm that uses a window of six words that would be negated before or after the negation term. Other errors were linked to the use of the custom subsets, wrongly detecting concepts when only one word of the concept term was present (e.g., detect *heart block* in "when walking 2 blocks") or ignoring contextual information. This lack of context problem is the most common error cited when MMTx failures were analyzed [39].

5.4. Evaluation results

The recall difference between the default data set and the custom subset, along with the gain in speed, made us select the version of our NLP module using MMTx with a custom subset for the subsequent clinical use of the Automated Problem List system. The difference in speed is related to the drastic reduction in size of the data set used by MMTx. This reduction of the concepts to extract by MMTx also significantly increased the recall, without significantly affecting the precision but with a little increase in fallout. An annoying side effect of this drastic size reduction was the higher rate of false positives when using the custom subset. It forced us to develop a disambiguation process before and after the use of the NLP module by our system. Without this disambiguation, precision would have been lower with the custom subset.

The comparison of the paper-based review and the electronic review showed that the web-based application allowed a better recall, with a similar precision. This difference was made possible by the two-step review process of the web-based application, comparing the pre-review with NLP results and proposing additions or removals if relevant. One could argue that this improvement was only due to reading the same document twice, but we showed that reading a document twice improved the recall but reduced the precision, when the help of NLP increased both recall and precision. Finally, the comparison of the review looking for 20, 40, or 80 problems showed that with more problems to look for, recall decreased. This result is consistent with the subjective conviction that a more complex task (more problems) results in more errors.

5.5. Study design

To reduce biases and improve the generalizability of this evaluation, we attempted to follow published criteria for effective evaluation of NLP systems [40]. Most criteria listed in the cited publication were respected, although, in our case, the developer of the system also designed and led its evaluation. To minimize this issue, documents were randomly selected after the system was frozen for evaluation, and reviewers did their task fully independently. Data collection was fully automated and reviewers were blinded, to avoid assessment biases. Measures were all standardized and highly automated. A recruitment bias was avoided by clearly defining the inclusion/exclusion criteria of patients to randomly select test documents from.

The sample size was sufficient to show significantly different recall between the default data set and the custom subsets used with MMTx, and therefore helped us select the right configuration to use with our system for clinical use, but a larger sample could have reduced confidence intervals and possibly demonstrated a significant difference in precision. It may also have shown a greater difference in recall between the reviews looking for different numbers of medical problems, and between the reviews with simple and with double reading of a document, when detecting only 20 or 40 problems.

6. Conclusion

We have developed tools to automate the problem list using NLP to extract potential medical problems from free-text documents in a patient's EMR. This system's goal is to improve the problem list's quality by increasing its completeness, accuracy, and timeliness. Here, we have described an evaluation of the NLP module and shown reasonable performance for clinical use of our system. We are currently engaged in clinical testing of a problem list management tool that included the NLP system described above. The effect of our system on the quality of the problem list will be evaluated, and we hope to find an increased proportion of correct medical problems, a reduced proportion of incorrect medical problems, and a reduced time between problem identification and addition to the problem list. These features will help to guarantee the quality of this central component in our problem-oriented Electronic Medical Record. Further analysis of the NLP module is planned, including a comparison to other NLP tools present in our laboratory (like the NLP application developed in our laboratory—MPLUS—and another tool based on regular expressions).

Acknowledgments

We thank Adam Wilcox, Roberto Rocha, and Alun Thomas for their suggestions that helped improve the web-based review application and design its evaluation. This work was supported by a Deseret Foundation Grant (Salt Lake City, Utah).

References

[1] Bayegan E, Tu S. The helpful patient record system: problem oriented and knowledge based. Proc AMIA Symp 2002:36–40.

[2] Van Ginneken AM. The computerized patient record: balancing effort and benefit. Int J Med Inform 2002;65(2):97–119.
[3] Scherpbier HJ, Abrams RS, Roth DH, Hail JJ. A simple approach to physician entry of patient problem list. Proc Annu Symp Comput Appl Med Care 1994:206–10.
[4] Wasserman H, Wang J. An applied evaluation of SNOMED CT as a clinical vocabulary for the computerized diagnosis and problem list. Proc AMIA Symp 2003:699–703.
[5] Unified Medical Language System (UMLS®). Available at <http://www.nlm.nih.gov/research/umls/>.
[6] Payne T, Martin DR. How useful is the UMLS metathesaurus in developing a controlled vocabulary for an automated problem list? Proc Annu Symp Comput Appl Med Care 1993:705–9.
[7] Goldberg H, Goldsmith D, Law V, Keck K, Tuttle M, Safran C. An evaluation of UMLS as a controlled terminology for the Problem List Toolkit. Medinfo 1998;9(Pt 1):609–12.
[8] Zelingher J, Rind DM, Caraballo E, Tuttle M, Olson N, Safran C. Categorization of free-text problem lists: an effective method of capturing clinical data. Proc Annu Symp Comput Appl Med Care 1995:416–20.
[9] Wang SJ, Bates DW, Chueh HC, Karson AS, Maviglia SM, et al. Automated coded ambulatory problem lists: evaluation of a vocabulary and a data entry tool. Int. J. Med. Inform. 2003;72(1–3):17–28.
[10] Chi E, Lyman M, Sager N, Friedman C. Database of computer-structured narrative: methods of computing complex relations. SCAMC 85; 1985; 221–226.
[11] Sager N, Friedman C, Chi E. The analysis and processing of clinical narrative. Medinfo 1986:1101–5.
[12] Friedman C, Johnson SB, Forman B, Starren J. Architectural requirements for a multipurpose natural language processor in the clinical environment. Proc Annu Symp Comput Appl Med Care 1995:347–51.
[13] Friedman C, Shagina L, Lussier Y, Hripcsak G. Automated encoding of clinical documents based on Natural Language Processing. J Am Med Inform Assoc 2004.
[14] Aronson AR, Bodenreider O, Chang HF, Humphrey SM, Mork JG, Nelson SJ, et al. The NLM indexing initiative. Proc AMIA Symp 2000: 17–21.
[15] Aronson AR. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. Proc AMIA Symp 2001:17–21.
[16] Zou Q, Chu WW, Morioka C, Leazer GH, Kangarloo H. IndexFinder: a method of extracting key concepts from clinical texts for indexing. Proc AMIA Symp 2003:763–7.
[17] Denny JC, Smithers JD, Miller RA, Spickard 3rd A. "Understanding" medical school curriculum content using KnowledgeMap. J Am Med Inform Assoc 2003;10(4):351–62.
[18] Pratt W, Yetisgen-Yildiz M. A study of biomedical concept identification: MetaMap vs. People. Proc AMIA Symp 2003:529–33.
[19] Aronson AR. Query expansion using the UMLS Metathesaurus. Proc AMIA Symp 1997:485–9.
[20] Wright LW. Hierarchical concept indexing of full-text documents in the unified medical language system information sources map. J Am Soc Inf Sci 1998;50(6):514–23.
[21] Pratt W, Wasserman H. QueryCat: automatic categorization of MEDLINE Queries. Proc AMIA Symp 2000:655–9.
[22] Weeber M, Klein H, Aronson AR, Mork JG, de Jong-van den Berg LT, Vos R. Text-based discovery in biomedicine: the architecture of the DAD-system. Proc AMIA Symp 2000:903–7.
[23] Sneiderman CA, Rindflesch TC, Bean CA. Identification of anatomical terminology in medical text. Proc AMIA Symp 1998:428–32.
[24] Rindflesch TC, Hunter L, Aronson AR. Mining molecular binding terminology from biomedical text. Proc AMIA Symp 1999:127–31.
[25] Brennan PF, Aronson AR. Towards linking patients and clinical information: detecting UMLS concepts in e-mail. J Biomed Inform 2003;36(4–5):334–41.
[26] Shadow G, McDonald C. Extracting structured information from free text pathology reports. Proc AMIA Symp 2003:584–8.
[27] Chapman WW, Bridewell W, Hanbury P, Cooper GF, Buchanan BG. A simple algorithm for identifying negated findings and diseases in discharge summaries. J Biomed Inform 2001;34(5):301–10.
[28] Mutalik PG, Deshpande A, Nadkarni PM. Use of general-purpose negation detection to augment concept indexing of medical documents: a quantitative study using the UMLS. J Am Med Inform Assoc 2001;8(6):598–609.
[29] Friedl JEF. Mastering regular expressions. Cambridge: O'Reilly; 1997.
[30] Meystre SM, Haug PJ. Automation of a problem list using Natural Language Processing. BMC Med Inform Decision Making 2005;5:30. Available at <http://www.biomedcentral.com/1472-6947/5/30/>.
[31] UMLS Metathesaurus Fact sheet. Available at <http://www.nlm.nih.gov/pubs/factsheets/umlsmeta.html/>.
[32] MetamorphoSys. Available at <http://www.nlm.nih.gov/research/umls/meta6.html/>.
[33] MMTx Data File Builder. Available at <http://mmtx.nlm.nih.gov/DataFileBuilder.pdf/>.
[34] Chapman WW. NegEx 2. Available at <http://web.cbmi.pitt.edu/chapman/NegEx.html/>.
[35] Rocha RA, Huff SM, Haug PJ, Warner HR. Designing a controlled medical vocabulary server: the VOSER project. Comput Biomed Res 1994;27(6):472–507.
[36] Dolin RH, Alschuler L, Beebe C, Biron PV, Boyer SL, Essin D, et al. The HL7 clinical document architecture. J Am Med Inform Assoc 2001;8(6):552–69.
[37] Meystre S, Haug PJ. Medical problem and document model for natural language understanding. Proc AMIA Symp 2003:455–9.
[38] Ashton CM, Kuykendall DH, Johnson ML, Wray NP. An empirical assessment of the validity of explicit and implicit process-of-care criteria for quality assessment. Med Care 1999;37(8):798–808.
[39] Divita G, Tse T, Roth L. Failure analysis of MetaMap Transfer (MMTx). Medinfo 2004:763–7.
[40] Friedman C, Hripcsak G. Evaluating natural language processors in the clinical domain. Methods Inf Med 1998;37(4–5):334–44.