Abstract

Although medical language processing (MLP) has achieved some success, the actual use and dissemination of data extracted from free text by MLP systems is still very limited. We claim that the adoption of an “enriched-document” paradigm (or “document-centered” view) can help to address this issue. We present this paradigm and explain how it can be implemented, then discuss its expected benefits both for end-users and MLP researchers.

1 Introduction

Some medical language processing (MLP) systems have achieved a reasonable level of performance and success from a technical point of view, the most well-known one being unarguably the Linguistic String Project Medical Language Processor (LSP/MLP [1]). However, the actual use and dissemination of data extracted from free text by MLP systems is still very limited.

In our opinion, the most crucial factor which impedes a wider use of these results by health care professionals involves the expected quality of the resulting data: one obviously cannot rely on extracted data as a substitute for the original text as long as this data is not produced by a near 100% safe procedure. Free text processing has not reached such a level of performance — neither can one expect humans to.

On another plan, MLP results are also not disseminated enough to the research community. Some conditions for this to happen are indeed political (individual and organisational willingness) and infrastructural (availability of network and servers). But there is also an issue of representation framework for such interchange to take place. At present, the kinds of results that are produced by different MLP systems are often so distant from one another that their levels of representation of medical information are simply not congruent.

We present in this paper a paradigm which can help to address these two problems. In this paradigm, we consider the analyses of a text as enrichments to this text: the resulting document integrates the original text, “annotated” by the data produced by MLP systems. Extracted data no longer replace the original text, which can be checked by the information user, alleviating the need for 100% reliable data. Besides, such an approach can facilitate the identification of data exchanged between MLP researchers.

This view of text management is not new. It is widespread in the natural language processing (NLP) and document processing communities, and is becoming increasingly known in medical informatics too. Our purpose is to transpose it to the context of medical language processing.

We first recall some background material, and then present the principles of the proposed model. We illustrate the model on several examples, and show how the resulting enriched documents can be exploited. We discuss the advantages that such a “document-centered” view of analyzed medical language should bring both to final users (health care professionals) and to MLP researchers. We also identify some limitations, and propose further directions of research.

2 Background

We present here the background which supports the paradigm proposed in this paper: a schematic reconstruction of MLP tasks, the emergence of a trend towards a document-centered patient record, the usefulness of annotated corpora, and some basics about document markup (see figure 1).
2.1 A reconstruction of medical language processing tasks

MLP systems aim at extracting “medical data” from free text. The prototypical application is diagnosis encoding (e.g., [2]), where the desired output is a diagnosis code in an existing classification (e.g., the International Classification of Diseases [3]). This is a classification, or categorization, task: given a string of words and an a priori set of classes, determine the class which should be assigned to this string. An extended version of categorization can assign several classes (or indices, or keywords) to an input string, for instance, to associate a set of SNOMED codes to an input sentence [5]. Some “ancillary” procedures in MLP are also categorization tasks: e.g., determining the part-of-speech of each word, or (more interestingly) determining the broad semantic category of each word or expression. The latter task plays a central role in [1], and [6] shows that its status can be promoted from ancillary MLP to an actual, end-user service. Although categorization may be the only task expected from an MLP system (e.g., in [2]), another important task is segmentation: given a string of words, identify relevant segments in this string. These segments correspond to meaningful units, for instance the expression of a diagnosis — which one might further want to categorize. Segmentation, in this context, allows to focus on (and categorize) specific parts of a text, rather than considering it as an indistinct whole. An application of segmentation could be to identify in a text all instances of diagnoses. This task is different from the above categorization of a string which is already known to be a diagnosis. “Ancillary” segmentation tasks include morpho-syntactic segmentation (into words, sentences, phrases and other syntactic constituents) as well as the identification of semantic units such as diagnoses, acts, or anatomic localizations.

Segmentation may be structured: segments may be nested in one another. For instance, a diagnostic expression may mention a specific anatomical location: we then have an anatomic location segment inside a diagnosis segment. The basic model of main-stream syntax is that of constituent structure, which is indeed a structured segmentation. Sager’s MLP system [1] can essentially be described as a structured segmentation and categorization system. Categorization and segmentation correspond to the paradigmatic and syntagmatic division in linguistics. It is therefore not a surprise that the data extracted from a natural language text can often be represented as a categorization of text segments. This accounts for a large part of MLP systems (e.g., [1, 7]), and does address a substantial part of the needs for data extraction from medical text.

However, instead of assigning a text segment an atomic category drawn from an a priori set, one may choose to build for it a complex category. The hypothesis is that a finite set of a priori (pre-coordinated) categories is not sufficient for dealing with the open-endedness of natural language, and that one therefore needs to rely on a generative language for describing a potentially infinite set of (post-coordinated) categories. This is the assumption made by those who rely on some sort of compositional language [4, 8] for representing medical information, and in particular by the tenants of knowledge representation languages [9, 10]. Such a complex categorization is generally not reducible to atomic categorizations of structured segments, and must therefore be considered as a distinct class of MLP production.

The paradigm we present in this paper naturally caters for the first two types of NLP output: atomic categorization and (structured) segmentation, and thus can help to manage a very large part of the results of MLP systems. It can also be made, with some restrictions, to host the latter type (complex categorization).

2.2 The document-centered patient record

The traditional view of the electronic medical record (EMR) is that of a database which holds items of coded, or standardized, information. While this view has achieved some success, it does require a substantial effort of standardization over medical information. Moreover, a complete formalization of medical information is not theoretically reachable — this is precisely one of the reasons why natural language still has some appeal to medical practitioners.

An alternate view gives textual documents a first-class status. It is inspired both by an observation of non-computerized medical practice (e.g., [11, 12]) and by the
fast-growing document processing industry. In that view, the EMR is a collection of documents, which can be composed of text, images, etc. Textual documents here can, and generally do, have structure. They can be hierarchically divided into identified chunks, down to any suitable level, thus providing much of the same data structuring advantages as databases. But at the same time, not every piece of text has to be standardized: the balance between formal data and natural language text can be more finely tuned than with traditional database models, and they can merge together more tightly. The expected benefit is first a better account of available medical information, described as it can be expressed, rather than as it should be expressed to fit a predefined computer model. Assuming that a document-centered view of the medical record is closer to non-computerized practice than an item-oriented view, an additional benefit is a potentially better-suited interface for health care professionals to work on the patient record. Several attempts at experimenting with such a document-centered patient record are ongoing [13, 14, 15, 16, 17, 18].

The document-centered view aims at finding a suitable tradeoff between free text and formal data. By producing data from free text, MLP therefore plays a central role in a document-centered patient record and should benefit from it.

2.3 Annotated corpora as a precious resource

The 80’s have seen the development of increasingly larger on-line text corpora, and their dimension has now exploded with the advent of the World Wide Web. While large volumes of text are a very useful resource for studying language, for evaluating NLP tools and for training them [19], annotated corpora are an even more valuable asset for these same purposes. Annotated corpora link texts with their analyses: a typical, basic level of analysis may involve the identification of sentence boundaries or the part-of-speech of each word. More elaborate analyses associate syntactic parses with sentences or semantic categories with words. The resulting corpora are much more than text: they constitute sources of knowledge about language.

Typical work on annotated corpora includes:

- evaluating NLP systems: this is how the DARPA MUC competitions have been proceeding [20];
- training NLP systems: e.g., a part-of-speech tagger can tag accurately some 97 % of the words of a corpus provided it is given a large enough hand-annotated training corpus [21];
- building resources for NLP systems: this is actually a variant of the preceding point: a corpus where each word is marked for semantic category can be used to learn selectional restrictions or taxonomies [22].

The same idea lies behind some statistical approaches to MLP, which learn word to code associations on the basis of a manually provided training sample.

The paradigm we present is based on the same idea of annotating a text with its analyses: we believe this can help MLP researchers both for their own developments and by facilitating the exchange of enriched corpora.

2.4 Marking up text

Managing structure, categories, and annotations in a text can be realized by inserting in it tags which mark this structure, specify the categories, and identify the annotations. The publishing industry has pushed forth an initiative which led to the adoption of an ISO standard for this purpose: the Standard Generalized Markup Language (SGML [23, 24]). SGML has since then spread to a wide variety of industries and research disciplines. HTML (HyperText Markup Language), the language of the World Wide Web, is itself a fixed application of SGML.

SGML basically allows to define the nature and structure of the “tags” which can be inserted in a document. Tags delimit “elements” of a document, i.e., categorized segments, such as a section title, paragraph, diagnosis or person name. Tags may also associate “attributes” with elements, such as the level or number of a section title, the sex of a person or the standardized code for a diagnosis. The set of allowed tags in a class of documents and their authorized order of appearance and nesting are specified in a “Document Type Definition” (DTD), also written in SGML.

Large SGML DTDs have been proposed for a wide variety of industries and scientific disciplines. Of prominent interest for NLP is the Text Encoding Initiative [25], an international project which has led to a DTD encompassing a large part of the needs of the humanities, literary and linguistic research. In the medical informatics arena, the HL7-SGML group has recently been promoting the use of SGML for the medical record [26].

3 The enriched document model

In this section, we present the principles of the enriched document paradigm and sketch a possible implementation based on the SGML language.

3.1 MLP as a document enrichment process

We propose to handle the analyses which MLP can produce from a text as enrichments to this text. Once “data”
has been obtained, the original text is not “forgotten”. It remains there in the forefront to help to manage the derived data. It is still the reference that the reader of the patient record may want to consult when examining this patient’s data.

### 3.2 Indexing interpretations on the source text

The general picture of an enriched document is that of a text, some portions of which are associated with “analyses”, or “interpretations” (in the above-proposed terms, “categorizations”). A very simple principle is applied throughout the model: index these interpretations on the source text that they interpret (see figure 2). This gives a deserved importance to the segmentation aspect of NLP. The same principle was at the root of seed models in computational linguistics (the chart model for parsing [27]) and artificial intelligence (the blackboard model for problem-solving [28] and the ATMS model for belief maintenance [29]).

![Diagram showing indexing interpretations on the source text.](image)

**Figure 2:** Indexing interpretations on the source text.

Let us review some of the properties this indexing brings to the resulting enriched document.

- The default entry point into the document remains the original text. None of its structure or contents is lost in the new document, as might be the case in an item-oriented paradigm. Therefore, the basic structure, the basic reading of the document is that of the source text. The importance of the structure and physical aspect of documents in the patient record have been stressed by some authors (in particular, [11]): keeping these features should be helpful to the reader.

- Several interpretations of the text can be added monotonically to the same document. For instance, one MLP process could identify and categorize the overall structure of the text (reason for admission, antecedents, etc.) whereas another one could determine the SNOMED encodings of the relevant expressions in the text. This possibility of task division has implications both for MLP architectural issues and for research team collaboration.

  - An analysis of only some parts of a text can be handled. Moreover, the actual piece of text which gives rise to a specific analysis can be precisely identified. By applying categorization to a segmented text, one can make the difference between an analysis of a whole text and an analysis of a limited extract of this text. For instance, a diagnosis will not have the same meaning if it is presented as the diagnosis of a patient (attached to the whole report) or as a diagnosis found in the “reason for admission” section or in the “antecedents” section.

  - MLP-provided data can be useful as normalized search data. By indexing analyses on their source text, one can thus link back to the source text from the search data. As a consequence, since text and corresponding data are synchronized, one can have search processes operate on normalized data while letting the user view understandable text. This back link is also an essential feature to enable traceability and quality control of MLP results.

  - Since several such interpretations can be synchronized, search criteria operating on several levels of interpretation can be combined together, resulting in greater search power. For instance, one can search for a given diagnosis (segment indexed with a “diagnosis” category) in a specific section (segment indexed with a “section” category) of a discharge summary – and search or view the corresponding original text.

  - Finally, if an analysis produces a categorization, this categorization is inherited by the text segment indexed with it. Such a categorization can be made conspicuous to the reader through appropriate markup and rendering devices [6].

### 3.3 A simple implementation of the enriched text model

The main features of the enriched text model can be implemented with a few SGML constructs.

#### 3.3.1 Segmentation and categorization

A basic kind of analysis simply consists in categorizing text segments. Text segments can range from a (part of)
word to a whole paragraph, section or document. Categories may be morphosyntactic (part-of-speech, lemma, syntactic category) as well as semantic (broad semantic category, word sense, position in a thesaurus) or conceptual (position in an ontology). Another kind of segmentation concerns the general structure of the text (header, signature, divisions, paragraphs).

An analysis which results in the categorization of a text segment can be recorded as an encapsulation of this text segment within a start and end tags assigned to this category. For instance, the text on figure 3 is tagged with SGML elements denoting segmentation and categorization of structural units (div, head, p) and semantic units (name, date, diagnosis). A variant uses attribute values to specify categories. In the same example, a semantic categorization of divisions is encoded as a type attribute, and a precise semantic category for diagnoses is specified as an ICD-9CM attribute whose value is a position (a hierarchical code) in the ICD-9-CM classification.

Figure 4 shows an additional example where the beginning of a patient discharge summary is tagged with divisions (div, head), paragraphs (p), person names (name), health care locations (locactions, typelocs, nomlocs), dates (date), symptoms and diagnoses (etatpath), medical treatments (theramed, drug, poso) etc.

The rules that govern which elements and attributes can be used and how they can nest are specified in an SGML document type definition (DTD). Figure 5 shows an extract of the DTD enforced in the document of figure 4. Each legal element is declared (telement declaration) with its allowed attributes (attlist declaration). The content of an element can be simple text (#PCDATA) or other SGML elements.

3.3.2 More complex analyses

More complex analyses need to be represented explicitly. The TEI has defined several notations for managing such analyses (see, e.g., [30]).

The simplest one consists in inserting the analysis as text enclosed within specific start and end tags, at the place in the text where it belongs (interlinear analysis). Assuming for instance that we want to include the conceptual representation of each sentence as derived by an MLP system such as MENEITAS [10] or RECIT [31], we can segment the text into sentences (element s), each of which includes this conceptual representation (in a cr element — see figure 6). The corresponding DTD must indeed cater for these elements.

In variant encodings, such analyses are grouped together in a special part of the document (for instance, in the end) and SGML reference links tie together a text segment and its analysis. This has the advantage of keeping the “main” part of the document “clean”. It is even possible to keep the analyses outside the source document, so that it remains untouched (and therefore may have a read-only status, a safe way to implement data integrity), using external references to positions in this document. HyTime links [32], in particular, allow to do this.

A limitation of the proposed encoding for “complex” analyses is that they are stored as plain text rather than encoded as markup (elements and attributes). As a consequence, the SGML processing mechanisms built in SGML tools do not help much for processing (e.g., searching) these representations.

4 Uses of enriched documents

4.1 Overview of uses

We envision two broad classes of users for enriched medical documents: health care professionals and medical informatics researchers (see figure 7). We define the first class as being interested primarily in working on patient data. The second class is concerned with methods and tools for processing medical language and knowledge.

An enriched document may be used in (at least) three ways. One may simply want to consult the document — the enrichments can help reading or navigating through the document. One can obtain information or data from the document — for instance, extract patient information to be sent to another person or system. Finally, one may want to store data into the document — including writing the original text. The last two operations may be combined, for instance when applying NLP tools to a text and
Monsieur DURAND, âgé de 88 ans, est hospitalisé le 18/1/1997 pour un diagnostic ICD-9CM '413.9': angor spontané.

The first two operations on an enriched document are extremely simple to implement with SGML-aware software. For instance, using an SGML browser (e.g., Panorama PRO, [33]), one can define a style sheet which specifies how each kind of SGML element should be rendered. Figure 8 shows a Panorama display of the document of figure 4, where section heads are in boldface, symptoms and diagnoses are in italics, examination results are indented, and person and location names are replaced with an icon.

With this same SGML tool, one can define "navigators", user-tailored tables of contents which are presented in a companion window and synchronized with the main text window. Figure 9 shows a navigator based on section heads and symptoms and diagnoses. This navigator can be used both as a synopsis to get a quick feel for the patient's history and, indeed, as a navigation device: a
click on one of the items in this window scrolls the main window to the actual occurrence of this item in the source text. With the appropriate tools (here, the Panorama Pro SGML browser), these helps come at a marginal cost once the text is tagged.

The third feature (hyperlinks) generally requires that the desired links exist and encodes them as appropriate tags in the document. These links could be precomputed, e.g., by NLP software, or be user-defined.

### 4.3 Searching a collection of enriched documents

The basic search mechanism in an item-oriented database deals with formal data. Searching natural language texts needs to resort to different methods, namely, content-oriented search, also called full-text search. Recent software allows to apply full-text search to structured text, taking into account its segmentation and categorization features. So for instance, one can search for occurrences of the word “kidney” in diagnosis elements that are nested in div elements with attribute value type=’reason for admission’.

One can at will choose to search on text, on tags (and attributes), or on a combination of both: encoding MLP analyses as SGML-tagged segments in enriched text enables the application of off-the-shelf software to perform powerful searches on both the original text and the data extracted from it.

As hinted above, such tools know how to manipulate SGML markup and plain text. However, confronted with, for instance, the conceptual graph of figure 6, the best they can do is to consider it as text. As a result, the full formal meaning of the conceptual graph is lost. Nevertheless, concept and relation names can be searched as if they were keywords, which can still be useful (as a fallback) depending on the kind of queries which are processed. A better account of such elaborate representations requires some investment beyond a simple use of off-the-shelf SGML software.

### 4.4 Extracting information

An enriched document is an elaborate compound from which one may wish to extract only a selected subset. This may be useful, for instance, in a workflow context to produce a targeted report to be sent to a correspondent. This is also convenient to build different views of the same document for consultation, as suggested above.

The basic mechanism for these purposes is the selection, in tagged texts, of specified segments, based on markup and content, and their rearrangement in the desired order to produce the target document. This mechanism can be effected by SGML transformations, for which a vari-

---

Figure 5: A document type definition (extract).
Monsieur DURAND, âgé de 88 ans, est hospitalisé le 18/1/1997 pour un diagnostic ICD9CM '413.9' angor spontané à répétition.

Figure 6: Embedding a conceptual representation.

(Windows 95 screen dump)

Figure 8: Rendering an enriched document.

4.5 Enriching a document

Enriched documents accommodate expansion in at least two contexts. First, health care professionals must be able to enter new information into the patient record (with the usual precautions to comply with authorizations and ensure data integrity). Beyond the mere creation of new documents, a reader can annotate existing documents [36]: the document-centered paradigm facilitates “active reading”.

Second, MLP programs can be applied to the current state of a document and perform interpretation tasks as proposed above. The basic flow of control consists in selecting input segments, passing them to the MLP program, and then inserting the results back into the enriched document structure. A typical processing chain could be: select all occurrences of diagnoses (diagnosis elements, identified in a previous pass); pass them to an encoding program, which produces an ICD code for each of them; store the resulting ICD codes into an ICD9CM attribute of the diagnosis element. The Multext project [34] has defined a general NLP architecture where components work on an SGML encapsulation of natural language data. Note that MLP programs need not all be SGML-aware: Multext also proposes alternate encodings and transcoders so that usual NLP components can be plugged into such a processing chain.

Such an architecture facilitates the reuse of existing MLP components: its strength relies on the definition of a set of analysis levels (input or output of NLP components) and the corresponding encodings. Some general-purpose levels have been identified and some encodings proposed by NLP projects such as Multext or the TEI [25]. An obvious need for this kind of architecture to be really useful for MLP is to define relevant levels of analysis for medical texts. We return to this point later.

4.6 Exchanging enriched documents

The general NLP community has been exchanging data and programs for a while now: one can find repositories for lexicons, annotated corpora, parsers and other NLP resources in multiple places over the world (see for instance the Natural Language Software Registry [37] or the European Language Resources Association [38]) — although one must stress that resources are most developed for the English language.
The MLP community, in contrast, lacks a comparable level of sharing of specific resources. This point was put forward at the last meeting of Working Group 8 (Natural Language Processing) of the European Federation for Medical Informatics, which was held during the Medical Informatics Europe Conference in August 1996.

Enriched documents constitute annotated corpora which can be extremely useful to the MLP research community. They allow to compare more precisely different approaches, thanks to common test data, and would be a central piece of material in evaluations such as discussed by [39]. For instance, to work on automated diagnosis coding, a necessary material is a corpus of already coded diagnoses. A corpus of texts (e.g., patient discharge summaries) where diagnoses are identified and coded (i.e., segmented and categorized) with conventional SGML elements and attributes would be a valuable resource to share for this purpose. Annotated corpora can also be used to train coding programs. For instance, a large enough corpus where each word is tagged with a broad semantic category (e.g., Sager's 39 semantic categories, or the categories derived from SNOMED's 11 axes) could provide training material for a semantic tagger. The normalized document or corpus “header” provides room to describe the contents of and specific conventions applied in the annotated corpus.

We believe that the specification of a general set of MLP-produced data would help the MLP community focus on useful data to be exchanged. Furthermore, defining encodings for these data, in the form of SGML Document Type Definitions, would provide a more precise framework and would facilitate data exchange. Here again, the work of the TEI [25] is of interest: one of the main goals of the TEI was precisely to facilitate the exchange of annotated textual data between scholars. A valuable feature of the TEI scheme is the provision for each text or corpus of a header [40] where information about the document can be inserted: author, mode of construction and encoding, etc. This self-documentation is particularly useful when exchanging corpora between different teams.

5 Discussion

The enriched document paradigm presented here relies on an SGML encoding of medical text and data. As already mentioned earlier, SGML is making its way into the medical informatics community. We would like to stress two important specificities of our proposal. First, we are concerned with partially structured information: a mixture of text and data, where natural language itself can undergo some variable degree of structuring. This feature of traditional (paper-based) medical records is considered essential by authors such as [11]. This is different from approaches which assume a strict division between structured data and free text, as we understand that HL7 and its SGML initiative does. This is also different from approaches where structured data is accessed and displayed through HTML front-ends [41]. Second, we use SGML to tag the logical content of documents, rather than their external presentation. The actual presentation and layout of documents must be determined by separate style sheets — which can be important too, but must be designed at a different level. This contrasts with methods which directly encode the presentation of documents with specific SGML markup, for instance HTML.

Whereas MLP produces data from natural language text, it does not have to replace text with this data. A more conservative approach simply adds this data to the original text. As a consequence, depending on the use which is made of the resulting document, a 100% efficiency is not necessary. One can draw a parallel with information retrieval: 100% recall and precision are not necessary (neither are they available) to provide useful services to users. Answers to queries come in the form of actual texts or abstracts, which the user can read to make his/her final assessment. Since the user is “in the loop”, the program only serves as an assistant; the final thinking and decision belong to the user. The enriched document approach allows to follow a similar principle: although search and navigation can work on MLP-produced data, the user can always be presented with the synchronized, original text or text extract — together with the data if relevant.

Note that the consultation of an MLP-enriched document has something more than a “classical” (hyper)text. Navigation can rely not only on explicit text and links, but also on the attached, underlying MLP-produced interpretations. For instance, given the appropriate tools, one can navigate through the links induced by common semantic categories (e.g., go from one diagnosis to the next in a text) or search for text segments whose attached analysis at a given level satisfy some criterion (e.g., find all occurrences of conditions affecting the lower limits, based on a SNOMED encoding of the text). This opens up a whole range of dynamic navigation mechanisms.

MLP deals with tasks of various complexities. For instance, identifying segments of a given broad semantic cat-
egory (e.g., all occurrences of anatomical locations) is less difficult than providing a detailed conceptual representation for each sentence. Indeed, the results of MLP systems are generally less good as the difficulty of the task increases. The proposed enriched document architecture can allocate some room for the different levels of MLP results. The exploitation of the available data is then more or less efficient according to the available levels and the quality of MLP components. For instance, consultation helps (e.g., layout and rendering — see above) can be of varying quality depending on which segmentations and categorizations are available and how accurate they are. Here again, the approach is well suited to target services that can accomodate some degree of imprecision or incompleteness in their input data, and that give a major role to text. It allows to put to work in the short term the simpler MLP tools, and to introduce more sophisticated tools gradually, instead of having to wait until MLP is perfect before it can be used.

The underlying assumption throughout this paper is the possibility to specify a common set of general data types and their encodings for MLP-processed data: in other words, SGML document type definitions (DTDs) for enriched documents. This work still remains to be done, and is by no means an easy one. It is strongly linked to the definition of DTDs for medical language documents, which is itself closely related to the definition of the electronic medical record. Several tracks already go some way in this direction. In the NLP community, enterprises such as the Text Encoding Initiative or the Multext project have made steps towards DTDs for various kinds of texts and analyses. The TEI DTD, in particular, is a very wide-covrage, extremely parameterizable, document type definition which can be thus adapted to many needs. In the medical informatics community, efforts such as CEN TC251 or HL7 have considered the encoding of many kinds of medical data; both are considering the use of SGML to pursue their tasks.

The proposed architecture does not solve the issue of the incompatibility (or incomparability) of different representations of medical data. It can help to identify, though, that representations concern the same level of interpretation (e.g., different controlled vocabularies for coding diagnoses; or different knowledge representations for describing the meaning of a sentence), and provide notations to explicit the nature and extent of each representation; but it does not include means to go from one representation to another. Projects such as the UMLS [42] or GALEN [9] do address this issue. The enriched document paradigm can help, for its part, to anchor these different representations to natural language texts and to one another.

6 Conclusion

We have described a general paradigm for managing data produced by medical language processing systems and, more generally, medical data, from its textual form to a knowledge representation. The basic principle is to associate NLP-produced data to its source text, the natural implementation being based on SGML markup and tools. Although this does not constitute in itself a new NLP technique, we have shown that this principle can bring many benefits to MLP, both by fostering shorter term application of current MLP components in a health care setting, and by facilitating data interchange between MLP research teams.

The major need for this paradigm to be effective is the definition of an architecture and document type definitions for MLP-produced data. The definition of DTDs for medical documents is an independently motivated goal for the document-centered medical record [12, 18] or for a data-oriented electronic medical record [26]. We believe that these requirements should be joined within a common “medical document encoding initiative”.

7 Acknowledgments

We thank Vincent Brunie for useful discussions about SGML and SGML tools, and Dr. Thomas Similowski for providing patient record material for experimentation.

References


