Published in final edited form as:

Int J Hum Comput Stud. 2010 June ; 68(6): 370–385. doi:10.1016/j.ijhcs.2009.08.002.

Easing semantically enriched information retrieval—An interactive semi-automatic annotation system for medical documents

Theresia Gschwandtner^{a,*}, Katharina Kaiser^a, Patrick Martini^a, and Silvia Miksch^{a,b}

^aInstitute of Software Technology and Interactive Systems, Vienna University of Technology, Favoritenstraße 9-11/188, 1040 Vienna, Austria

^bDepartment of Information and Knowledge Engineering, Danube University Krems, Dr.-Karl-Dorrek-Straße 30, 3500 Krems, Austria

Abstract

Mapping medical concepts from a terminology system to the concepts in the narrative text of a medical document is necessary to provide semantically accurate information for further processing steps. The MetaMap Transfer (MMTx) program is a semantic annotation system that generates a rough mapping of concepts from the Unified Medical Language System (UMLS) Metathesaurus to free medical text, but this mapping still contains erroneous and ambiguous bits of information. Since manually correcting the mapping is an extremely cumbersome and time-consuming task, we have developed the MapFace editor.

The editor provides a convenient way of navigating the annotated information gained from the MMTx output, and enables users to correct this information on both a conceptual and a syntactical level, and thus it greatly facilitates the handling of the MMTx program. Additionally, the editor provides enhanced visualization features to support the correct interpretation of medical concepts within the text. We paid special attention to ensure that the MapFace editor is an intuitive and convenient tool to work with. Therefore, we recently conducted a usability study in order to create a well founded background serving as a starting point for further improvement of the editor's usability.

Keywords

Semantic annotation; Graphical editor; UMLS

1. Introduction

One of the most crucial challenges in modern medical care is the handling of huge amounts of information. Making medical text computer-interpretable is an important step towards the effective processing of medical information. The following two examples demonstrate this need.

Analysis of patient records

Checking medical patient records against given findings, comparing parameters of different records, or statistically evaluating specific disease patterns are common tasks of medical research and clinical care. In particular, when a huge amount of information is to be captured and analyzed, the automation of these tasks is crucial for effective processing. Medical concepts within these patient records have to become easily extractable and comparable, in order to allow for queries such as how many patients above 60 years received a specific medication and how they recovered. Permitting to query concepts rather than keywords requires the previous identification, disambiguation, and tagging of medical concepts within these patient records.

Making medical knowledge computer-executable

On the other hand, the generation of computer-executable knowledge, such as contained in clinical practice guidelines (CPGs) (Field and Lohr, 1990)—systematically developed instructions and recommendations on appropriate treatment of patients in order to improve the quality of health care by providing the best available scientific evidence—is a very challenging task. For a detailed description of different approaches we refer to Peleg et al. (2003) and to Wang et al. (2002). However, the process of translating a textual CPG into a computer-executable model is an extremely time-consuming and cumbersome task, whereas medical experts and knowledge engineers have to collaborate intensively. Unfortunately, there is a great risk for knowledge engineers—who in general are not familiar with the details of medical concepts—to misinterpret the complex medical text while modeling the CPG. Thus, it appears that above anything else special attention has to be paid to ensuring the accurate interpretation of the text. Consequently, the disambiguation of the medical concepts included in the text of the document is imperative for correctly interpreting the text while further processing it into a computer-executable model.

As a first step towards the effective processing of medical text, the text needs to be translated into a computer-executable form. Thus, preprocessing the text by semantically annotating the medical concepts included is an important prerequisite to ensure an accurate interpretation of text. The semantic annotation of the narrative text of medical documents disambiguates the medical concepts included in the text and allows for semantically enriched information retrieval. Consequently, semantic annotations not only support the correct interpretation of the text, but also pave the way for more sophisticated analysis of the textual content such as statistical evaluations.

Multiple systems exist which generate a mapping from free biomedical text to medical terminologies, such as MicroMeSH (Elkin et al., 1988), Metaphrase (Tuttle et al., 1998), MetaMap (Aronson, 2001), PhraseX (Srinivasan et al., 2002), or KnowledgeMap (Denny et al., 2003). For a detailed description of annotation systems we refer to Reeve (2007).

We use the MetaMap Transfer (MMTx) program (Aronson, 2001) which maps concepts from the Unified Medical Language System (UMLS) Metathesaurus (Schyler et al., 1993)—the largest and well-accepted thesaurus in the biomedical domain—to corresponding concept chunks in the text.

1.1. The problem

It goes without saying that the complete reliability of the annotations is crucial in medical care—an extremely sensitive discipline. However, due to the ambiguity of free text, the automatic creation of an unambiguous mapping of UMLS concepts to medical concepts in the text of a document cannot be correctly accomplished by means of a semantic annotation system alone. The reliability of the MMTx results is not granted: on the one hand, MMTx

cannot always determine an appropriate or distinct concept for a text chunk and, on the other hand, MMTx sometimes provides the wrong syntactical information, which causes errors in the concept assignment. This, in turn affects the reliability of the annotated semantic information.

Thus, it is of paramount importance for medical experts to control these results and, if necessary, to correct them. However, controlling and modifying the output of the MMTx program is hard to handle for physicians, since the use of MMTx is command line based, and there is no graphical user interface. It requires programming knowledge to utilize the results of the MMTx program.

1.2. An interactive editor to support these tasks

There are two significant aspects if one wants to derive the greatest possible advantage from the information gained from semantic annotations of documents:

- guaranteeing the quality of the provided information, and
- providing easy access to all available information in combination with visualizing the given information in a comprehensive form.

We have developed an interactive editor to cope with these two aspects: the MapFace editor. It greatly facilitates the handling of the MMTx program, and, being easily extendable, it has the potential to become the one tool to deal with all kinds of semantic information necessary to ensure the accurate interpretation of a medical text. In the near future, the MapFace editor is to be extended by additional components, such as a component to detect coreferences in the text or a component to detect negated phrases (Gindl et al., 2008). These components take advantage of diverse information extraction methods to automatically perform these tasks, but again medical expertise is required to control the results.

Thus, the MapFace editor is designed to visualize different kinds of semantic information, to make them navigable and modifiable, but also to advise the user of those cases, for which a sufficient result could not be computed automatically, and last but not least to visualize important coherences to support a better understanding of the medical text. Using the MMTx program and the UMLS, which is a comprehensive biomedical thesaurus, MapFace can be used to semantically annotate medical documents of any medical domain.

Usability—We have evaluated the usability of the editor (see Section 6) to identify possible shortcomings of the design. This evaluation has yielded a number of suggestions of how to further improve MapFace; we will adapt the editor accordingly, thus ensuring an intuitive and convenient way of working with it.

To start with, we give an outline of related work in Section 2. In Section 3 we describe the UMLS and the MetaMap program. We continue with a detailed description of the problems involved in the automatic semantic annotation of a document by means of the MMTx program in Section 4. In Section 5 we introduce the MapFace editor, its components, and its features. Subsequently, we describe the setting and state the results of the usability evaluation of the MapFace editor in Section 6. We outline the extensions and improvements we intend to implement in the near future in Section 7. Finally, we conclude the main results of our work in Section 8.

2. Related work

Annotating documents is a frequently used technique to enrich documents and make them machine-readable and - understandable. Within the realm of the Semantic Web (Berners-Lee

et al., 2001) several frameworks have been developed to manually or semi-automatically enrich documents with metadata. Many of these frameworks also provide a graphical user interface to not only visualize but also edit these annotations.

GATE (Cunningham et al., 2002) is a software toolkit originally developed at the University of Sheffield. It comprises an architecture, a free open source API, a framework, and a graphical development environment. Each processing and language resource can have its own associated visual resource. The user interface is divided into three visible parts. One of them contains a tree that shows the loaded instances of resources. The one below this is used for various purposes—such as to display document features and that the execution is in progress. The third and the largest part is the central one. It contains the document with annotations marked by colors and a vertical pane shows the list of annotations used in the document (see Fig. 1(a)).

MnM (Vargas-Vera et al., 2002) is a semantic annotation tool for extraction of knowledge structures from web pages through the use of simple user-defined knowledge extraction patterns. It contains three components: among a learning component and an information extraction component it contains an ontology-based mark-up component which allows the user to browse and to mark-up relevant pieces of information. The user interface for marking up documents is composed of two main areas: (1) a pane for showing the (annotated) document and (2) a pane for visualizing the ontology (see Fig. 1(b)).

Melita (Ciravegna et al., 2002) is an interactive annotation framework that implements a methodology for active annotation for the Semantic Web based on Information Extraction. Its user interface is very simplistic. It is composed of two main areas: (1) the ontology representing the annotations that can be inserted: a specific color is associated to each node in the ontology. (2) The document to be annotated: annotated text chunks are highlighted in color according to the node in the ontology (see Fig. 1(c)).

The Stanford parser grammatical relation browser (Bou, 2008) is a simple-to-use graphic interface to grammatical structure and relations of any text as parsed by the Stanford Parser (Levy and Manning, 2004) (see Fig. 1(d)).

These annotation systems are designed for specific purposes (e.g., annotating web-pages) but not for a special domain; although some can be customized by means of machine-learning or ontology-plugins. However, these tools work with significantly smaller and less complex ontologies than MapFace does. Moreover, they do not provide the necessary functionality to manipulate and navigate the results of the MMTx program.

The MMTx program and the UMLS provide Natural Language Processing (NLP) components and a comprehensive ontology especially tailored towards the very complex nature of the medical domain. Thus, we have developed an editor designed to effectively gaining, modifying, and visualizing the results of the MMTx program as well as accessing and visualizing information provided by the UMLS.

3. The UMLS and the MetaMap program

Preliminary, we specify basic principles that are essential for the understanding of this paper. The following subsections contain an outline of the components of the Unified Medical Language System and details of the MetaMap program.

3.1. The Unified Medical Language System

The Unified Medical Language System (Lindberg et al., 1993) has been developed by the National Library of Medicine (NLM), USA, within the UMLS R&D project, initiated in

1986. It is a controlled compendium of many vocabularies and classifications of the biomedical domain and also provides a mapping structure between them. The UMLS was created to facilitate the development of computer systems that process biomedical text by offering access to this knowledge basis. There are three main UMLS knowledge sources: the Metathesaurus (Schlyer et al., 1993), the Semantic Network (McCray, 1989), and the SPECIALIST Lexicon (Browne et al., 2000).

The Metathesaurus—The UMLS Metathesaurus (Schlyer et al., 1993) is the largest thesaurus in the biomedical domain, containing medical concepts from more than 100 vocabularies. It is built from numerous thesauri (e.g., Medical Subject Headings (MeSH)¹, Computer–Retrieval of Information on Scientific Projects (CRISP)-from the National Institute of Health (2006), classifications (e.g., International Classification of Diseases (ICD-9-CM)²), clinical coding systems (e.g., Systematized Nomenclature of Medicine (SNOMED CT) (Côté et al., 1993)), and lists of controlled terms used in various biomedical documents.

Consequently, it is not built to be a single standard vocabulary, but enables exchange of information between different clinical databases and systems, in accordance with contextual and inter-contextual relationships between these diverse coding systems and vocabularies.

The Metathesaurus is structured by medical concepts or meanings; all alternative names and views of the same concept from the different vocabularies are linked within a hierarchical context. Furthermore, useful relationships between these concepts are represented.

The Semantic Network—The Semantic Network (McCray, 1989) specifies the categorization of the concepts in the Metathesaurus to basic semantic types (McCray and Nelson, 1995), such as *antibiotic* or *pathologic function*, just to name two of the 135 semantic types of the Network. All concepts in the Metathesaurus are assigned to at least one semantic type. It also defines the set of useful relationships between these types and concepts (the current release of the Semantic Network contains 54 kinds of relationships).

The SPECIALIST Lexicon and the SPECIALIST NLP Tools—The SPECIALIST Lexicon (Browne et al., 2000) is an English lexicon that includes many terms of the biomedical domain. It contains syntactic, morphological, and orthographic information about each word or term in the lexicon.

The SPECIALIST Natural Language Processing (NLP) Tools have been developed by the Lexical Systems Group of the Lister Hill National Center for Biomedical Communications, to facilitate NLP by providing lexical variation and text analysis for application developers using the UMLS. Among others there are lexical tools to manage lexical variations, text tools to analyze plain text documents into words, terms, phrases, sentences and sections, and spelling tools to suggest correct spellings for misspelled words.

3.2. The MetaMap program

Based on the SPECIALIST NLP Tools NLM has developed the MetaMap program (Aronson, 2001). The MetaMap program initially tokenizes the text of a document into sections, sentences, phrases and concepts. Subsequently, it computes a set of best matching concept candidates from the Unified Medical Language System (UMLS) Metathesaurus (Schlyer et al., 1993) for corresponding concept chunks in the text. Thus, an important task

¹Medical Subject Headings, 2008. National Library of Medicine. Updated annually.

²Commission on Professional and Hospital Activities, 1978. International Classification of Diseases, Ninth Revision, with Clinical Modifications (ICD-9-CM). United States National Center for Health Statistics, Ann Arbor.

of the MetaMap program is the detection of phrases—a phrase is a group of words that functions as a single unit, which can contain several concepts—in biomedical text and subsequently, assigning them to corresponding concepts in the UMLS Metathesaurus. In doing so it goes through five steps (Aronson, 2001):

1. Parsing arbitrary text into simple noun phrases by using the SPECIALIST minimal commitment parser.
2. Generating variants for each phrase, i.e., all its spelling variants, acronyms, abbreviations, synonyms, inflectional and derivational variants and meaningful combinations of these, using the SPECIALIST Lexicon and a supplementary database of synonyms.
3. Retrieving a candidate set of all Metathesaurus concepts containing at least one of the variants.
4. Evaluating each candidate against the input text by computing the mapping strength of the candidate using a linguistically principled evaluation function and then arraying the candidates according to their mapping strength.
5. Combining candidates from disjoint parts of the noun phrase, recomputing the mapping strength for the combined candidates and forming a set of best matching Metathesaurus concepts for the original phrase.

In contrast to most other systems, MetaMap not only supports exact matches between a text token and the UMLS concept, but also considers spelling variants, abbreviations, and synonyms, for instance, variants of the term “ocular” are “eye”, “eyes”, “optic”, “ophthalmic”, “ophthalmia”, “oculus”, and “oculi”. For each variant its distance score from the original generator term is computed. Moreover, MetaMap supports partial matches between a term and the UMLS concept, i.e., one or more tokens match a UMLS concept only partially. Additionally, it ranks the candidates found by combining four different measures derived from comparing the terms of the UMLS concept with the terms of the source phrase.

The MMTx program is a Java-implementation of the MetaMap program.

4. Problem description and scenario

As a first step towards the semantic annotation of a medical document, the plain text of the document is mapped to corresponding concepts from medical terminologies and medical concepts included in the text are annotated accordingly. Thus, the unstructured medical text is enriched by important meta-information provided by the terminology system. This requires

1. access to a thesaurus of medical concepts (providing meta-information), and
2. generating a mapping of these thesaurus concepts to corresponding text chunks of the document.

The MMTx program creates a mapping of the text by tokenizing it into noun phrases and assigning concepts from the UMLS Metathesaurus to corresponding concept chunks of these phrases. An example of the original output of the MMTx program for the phrase “the most effective preventer drug” is given in Fig. 2.

It is evident that it is not easy to assess and perhaps to correct the information contained in this text list—and this is just the output for one phrase. Moreover, it is not always possible to achieve a correct tokenization of the text as well as an unambiguous mapping of UMLS concepts and semantic types to text chunks. The following scenario demonstrates the

necessity to be able to modify and correct the results of the MMTx program with the help of the MapFace editor.

4.1. Scenario

When processing the sentence “For patients above five years with mild asthma inhaled steroids are the most effective preventer drug.”, the results of the MMTx program yield the following erroneous and ambiguous bits of information (see Figs. 3 and 4):

1. The tokenization into phrase chunks is erroneous, i.e., MMTx considers the text “with mild asthma inhaled steroids” to be one phrase, whereas it consists of two phrases “with mild asthma” and “inhaled steroids”.
2. The UMLS concept *year* matching the text chunk “years” is equally associated with the semantic type *pharmacologic substance* and the semantic type *idea or concept*. Consequently, the user has to determine the appropriate semantic type.
3. No distinct UMLS concept could be determined for two concept chunks of the text; again the user has to choose the correct UMLS concept for these text chunks:

The UMLS concept *abuse of steroids*, which is associated with the semantic type *mental or behavioral dysfunction*, and the UMLS concept *steroids*, which is associated with the semantic type *steroid*, both match the text chunk “steroids”.

The UMLS concept *prevents*, which is associated with the semantic type *functional concept*, and the UMLS concept *PREVENT*, which is associated with the semantic type *pharmacologic substance*, both equally match the text chunk “preventer”.

4. The concept chunk “five” (annotated with the UMLS concept *five* and the semantic type *quantitative concept*) and the concept chunk “years” (annotated with the UMLS concept *year* and either the semantic type *temporal concept* or the semantic type *idea or concept*) should rather be one concept chunk annotated with the UMLS concept *age* and the semantic type *organism attribute*.

Thus, it is still necessary for physicians to control these results and—if necessary—to modify them. Consequently, there is a strong demand for a simple editor to facilitate that task for experts in medical science. This led us to develop an editor with a graphical user interface, which enables physicians to solve this task without requiring special skills in programming.

4.2. Requirements

The problems as described in the previous section and the shortcomings of existing approaches led us to meet the following challenges for the design of MapFace:

- Enabling medical experts without special programming skills to work with the MMTx program.
- Providing means for annotating the text at two levels—a phrase level and a concept level.
- Providing means to easily navigate and control the output of the MMTx program.
- Highlight ambiguous concept matches to support the immediate identification of needs for intervention.
- Providing means to easily modify the affiliation of UMLS concepts to text chunks (e.g., means for searching the UMLS Metathesaurus for better suited concepts).

- Supporting the understanding of relations among semantic types or among UMLS concepts in the text.
- Retaining the information gained from the MMTx program as structured data objects.

5. The MapFace editor

The analysis of requirements, i.e., kinds of information needs and required features, resulted in two significant means which have to be realized in order to take greatest possible advantage from the automatically created annotation of a medical document. On the one hand, a convenient way of navigating, controlling, and correcting the annotation has to be provided; on the other hand, an intelligible visualization of the comprehensive information, i.e., the tokenization of the text into concept chunks and phrase chunks, the semantic information, and relations between text tokens, is required to support a better understanding of the annotated text. With respect to these aspects we have developed the MapFace editor.

5.1. The semantic structure behind MapFace

In order to enable the navigation, control, and correction of the automatically created annotations within the medical document, we used a data structure that is mainly adopted from MMTx (see Fig. 5). By means of this data structure we are able to represent the necessary information and to perform the required procedures on it.

A document contains a set of sections. Each paragraph in a document corresponds to a section. Each of these sections consists of a set of sentences. A sentence is composed of a set of words and a group of words forming a single unit functions as a phrase. A phrase consists of at least one word. In order to search for UMLS concepts MMTx parses the phrase string and searches for concept candidates. Depending on the terminology sources used by the UMLS and the MMTx, respectively, more than one UMLS concept can be applicable for a certain term or term combination. The resulting set of UMLS concepts is called concept candidates.

Each UMLS concept is assigned to at least one semantic type, which gives us a better understanding of the concept's meaning. Thus, knowing and especially visualizing the semantic type of a concept can facilitate the proper selection of a concept's candidate. But we have also sensed that assigning a phrase a semantic meaning can support the understanding of the text and furthermore the automatic processing of the text. On this account we can also assign a semantic type to a phrase.

The UMLS also provides relationships between concepts and semantic types. Our data structure is also able to present concept relations and semantic relations between concepts as well as semantic relations between phrases.

5.2. The GUI

We have designed a graphical user interface based on the interfaces of established annotation tools, such as GATE (Cunningham et al., 2002). In adaption to specific requirements, i.e., the modification and visualization of the results of the MMTx program as well as the visualization of information gained from the UMLS, we have implemented features to visualize and modify the annotation at two levels—at a concept level as well as at a phrase level. In addition, we provide means to display and visualize relations between annotations of text tokens.

When the user starts the MapFace editor, a window with a menubar, a toolbar, and three different panes appears (see Fig. 6): the *editor pane*, the *candidates pane*, and the *annotation scheme pane*.

The editor pane—The main window of the graphical user interface of MapFace is the *editor pane*. It displays the text of the medical document. This is where the user selects the text that is to be processed by means of the MMTx program. In addition, in the *editor pane* the user can select the concepts or phrases of processed text by double-clicking the text chunk of the concept or phrase, where-upon a list of best matching UMLS concept candidates is displayed in the *candidates view*.

The candidates pane—The *candidates pane* is at the bottom of the user interface (see Fig. 6). It contains three different views, the *candidates view*, the *concept relations view*, and the *semantic relations view*.

The main view of the *candidates pane* is the *candidates view*. It provides information about the best matching UMLS concept candidates for a given text chunk and possibilities for editing. Therefore, the *candidates view* displays a list of matching UMLS concept candidates detected by the MMTx program for a selected concept chunk or phrase chunk in the editor, together with their semantic type, semantic collection, and semantic group. The lines of the table are highlighted according to the color coding of the corresponding semantic collections (see Section 5.3.1).

The *concept relations view* provides additional information for a UMLS concept candidate selected in the *candidates view*. It displays a list of all relations between the concept candidate selected in the *candidates view* and the UMLS concepts affiliated to concept chunks in the same section of the text. When selecting a relation from the list, the two concept chunks concerned are highlighted in the editor (see Section 5.3.4).

The *semantic relations view* provides a list of all relations between the semantic type of the UMLS concept candidate selected in the *candidates view* and the semantic types assigned to text chunks in the same section of the text. The two concept chunks or phrase chunks concerned are highlighted in the editor when a relation from the list is selected (see Section 5.3.4).

The annotation scheme pane—The *annotation scheme pane* is at the right of the user interface (see Fig. 6). The basic idea was to create an editor supporting different kinds of annotated information which is relevant for the processing of medical documents. This is why the *annotation scheme pane* contains different views for different annotation schemes.

Each view displays a list of subjects relevant for a specific annotation context. There are two annotation schemes implemented at present: the *semantic types annotation scheme* (see Fig. 7) and the *XML elements annotation scheme*. The former is concerned with the annotation of UMLS concepts and their semantic types as described in this paper, and the latter simply deals with the visualization of all XML elements occurring in the underlying XML document.

In order to be easily extendable, the editor's architecture permits the implementation of plugins. This, in turn allows the application of additional annotation schemes, for instance, annotation schemes for co-reference or negation detection.

5.3. Features

The following subsections explain how the semantic information is visualized (see Section 5.3.1), which information is annotated (see Section 5.3.2), how the user modifies the annotation (see Section 5.3.3), and what features are provided both to support the correct affiliation of equally matching meta-information to text chunks and to support a better understanding of the medical concepts (see Section 5.3.4).

5.3.1. Visualizing the annotation—The view of the *semantic types annotation scheme* contains a comprehensive list of semantic types of UMLS concepts, grouped by semantic collections (Chen et al., 2002) (for a different way of partitioning the UMLS semantic network see McCray et al. (2001)). Since there are 135 different semantic types, the grouping into 28 semantic collections facilitates the inspection of the semantic types.

In order to visualize the semantic information of annotated text chunks, we decided for a color-code. In contrast to the GATE (Cunningham et al., 2002) annotation tool, we went for the association of semantic information with unique and well distinguishable colors. The huge amount of semantic types available in the UMLS makes the association of each semantic type with a unique color impossible. Thus, we decided to color-code the 28 semantic collections (see Fig. 7). To this end, we started with 12 well distinguishable colors by Ware (2004) and derived 28 different colors. This color-code is used to highlight the associated text chunks accordingly.

For instance, the background color of the phrase chunk

[the most effective preventer drug]

indicates the semantic type assigned to this phrase chunk—in this case it is the color associated with the semantic type *pharmacologic substance* (phrase chunks are delimited by brackets in the MapFace editor).

In case of an ambiguous mapping to more than one UMLS concept, the text chunk in the editor is marked by a gray background, which serves as a reminder that the user should manually choose the correct UMLS concept from the list of equally matching candidates and assign it to this chunk. It is possible to turn off the highlighting of annotated text chunks, which leaves only the ambiguously mapped text chunks marked by a gray background. Thus, the user can identify the needs for intervention at a glance (see Fig. 8).

5.3.2. Annotated information—The annotation of the text is accomplished by inserting XML tags into the document. These tags deal with the tokenization of the text into sections, sentences, phrases, and concepts as well as with the semantic information about assigned UMLS concepts.

Information about medical concepts within the text: A concept chunk is annotated with the best fitting UMLS concept together with its semantic type (McCray and Nelson, 1995), its semantic collection (Chen et al., 2002), and its semantic group (McCray et al., 2001). For instance, the text chunk “patients” is annotated with the UMLS concept *patient*, which is associated with the semantic type *patient or disabled group* (see Fig. 3). This semantic type belongs to the semantic group *living beings*, as well as to the semantic collection *group*. In unambiguous cases the concept chunks are automatically annotated with the appropriate UMLS concept and the associated information.

Information assigned to phrase chunks in the text: Whenever a distinct semantic type for a phrase chunk could be determined, the phrase chunk is annotated automatically. To

manually annotate a phrase chunk, the user chooses one of the semantic types assigned to the concept chunks within this phrase (see Fig. 9).

5.3.3. Editing the annotation—If the user selects a text chunk in the editor, the annotated information is displayed in the *candidates pane*. This pane displays information about the assigned UMLS concept or—in case of ambiguity—a list of possible matching UMLS concept candidates. Furthermore, possibilities to modify the annotation at a semantic level are provided.

Choose a distinct UMLS concept candidate: In case of an ambiguous mapping of a concept chunk to more than one UMLS concept, a list of matching UMLS concept candidates is displayed in the *candidates pane*, from which the user has to choose the appropriate concept candidate and assign it to the concept chunk (see Fig. 10).

Look for additional UMLS concept candidates: If the appropriate candidate does not appear in the candidates list, the user can search the Metathesaurus for additional UMLS concepts, by entering an alternative expression for the concept. Subsequently, the entered expression is processed by means of the MMTx program and the matching UMLS concepts are then added to the candidates list of the concept chunk.

Determine semantic types of phrase chunks: In order to correctly define the semantic meaning of a phrase, the user selects a semantic type associated with one of the concepts within this phrase and assigns it to the phrase chunk.

For instance, the phrase “the most effective preventer drug” contains the concept chunks (with associated semantic types): “most” (*quantitative concept*), “effective” (*qualitative concept*), “preventer” (*functional concept*), and “drug” (*pharmacologic substance*) (see Fig. 4); hence, the appropriate semantic type for this phrase chunk is *pharmacologic substance*.

Delete the annotation: The annotated information can be deleted if it is not appropriate, i.e., tags which mark text chunks to be a concept chunk or a phrase chunk can be deleted as well as the annotated semantic information.

Furthermore, the editor provides features to easily modify the tokenization of the text into phrase chunks and concept chunks at a syntactic level.

Modify the bounds of annotated text chunks: If a phrase chunks or a concept chunk appears to be wrongly tokenized, MapFace provides possibilities to correct the boundaries of this token. Phrase chunks and concept chunks can be deleted, and in turn, new phrase and concept chunks can be created. This enables the user to modify the delimitations of these text chunks. For instance, the phrase chunk

[with mild asthma inhaled steroids]

has been wrongly tokenized by the MMTx program and is to be split up like this:

[with mild asthma] [inhaled steroids]

The splitting of this phrase chunk is accomplished by deleting the phrase chunk and subsequently creating two new phrase chunks. Associated semantic types are assigned automatically to new phrases—provided that a distinct semantic type could be determined. In the example above, the semantic type of the first phrase, “with mild asthma”, is *finding* and the semantic type of the second phrase, “inhaled steroids”, is *steroid*.

Proceeding in a similar way, two text chunks—two concept chunks or two phrase chunks—can be merged into one text chunk. For instance, the concept chunks

five years

By now, the concept chunk “five” is associated with the semantic type *quantitative concept* and the concept chunk “years” is associated with both semantic types *temporal concept* and *idea or concept*. Again, the user deletes the two concept chunks and subsequently creates one big concept chunk. In addition, the user searches the UMLS Metathesaurus for UMLS concepts matching the text “age”, which is entered by the user. The UMLS concepts *age* and *elderly* are then added to the candidates list of the newly created concept chunk. Finally, the user assigns the appropriate UMLS concept to the concept chunk.

five years

This concept chunk is now annotated with the UMLS concept *age* which is associated with the semantic type *organism attribute*. Besides, merging two phrase chunks can be accomplished in one step (which is quite convenient for this frequent type of action).

5.3.4. Supporting the correct interpretation—The MapFace editor was not only designed to edit the Metathesaurus information, but also to support the understanding of medical concepts within the text as well as important coherences between them by means of enhanced visualization features. To serve this need, the color-coding of semantic collections and correspondingly highlighting text chunks within the text of the document constitutes a key component. While processing the concepts of the text, the user consolidates the association of certain colors with referenced semantic meanings. This mental connection supports the prompt and correct interpretation of medical concepts within the text. The editor allows the user for highlighting all text chunks referring to one or more selected semantic types, which enables the immediate identification of semantically similar concepts. In addition, the association of colors with semantic meanings facilitates the selection of the correct UMLS concept from the list of possible candidates for a given concept chunk, or the selection of the appropriate semantic type for a phrase chunk. For additional support, the editor provides the following special features.

Relations between UMLS concepts: The UMLS Semantic Network defines relationships between semantic types and between UMLS concepts. To facilitate the selection of the correct UMLS concept candidate for a given concept chunk we provide a means which takes advantage of the information available for the context of the given concept chunk, i.e., UMLS concepts assigned to other text chunks in the same section of the text and relations of these UMLS concepts to the concept candidates of the given text chunk. By selecting a UMLS concept in the *candidates pane* and subsequently activating the *concept relations view*, a list of relations of this UMLS concept to other UMLS concepts is displayed. On selection of a specific relation, the two concept chunks concerned are highlighted in the *editor pane*. By examining the relations to other text chunks, additional knowledge is gained, which in turn promotes the correct interpretation of the medical concepts. This means filters the relations relevant for the given section of the text out of huge amounts of information available in the UMLS.

Relations between semantic types: In a similar way, MapFace checks for relations between a given semantic type and semantic types assigned to other text chunks in the same section, when activating the *semantic relations view*. Again, the two text chunks of each relation are highlighted in the *editor pane* (see Fig. 11). This feature not only concerns concept chunks but also phrase chunks, for both kinds of text chunks are annotated with semantic types. By exploiting these relations, support for interpretation and decision can be derived. Moreover,

means to access all relations among semantic types available in the UMLS may be of additional value for the correct interpretation of the semantic type. However, providing access to the huge amount of available information about relations among UMLS concepts may lead to an overload on information and thus, it may affect the correct interpretation of the UMLS concept adversely.

Automatic decrease of semantic types list: The list of semantic type candidates for a phrase chunk can be automatically decreased to the most likely ones by taking advantage of the information about the relations of each candidate to the semantic types assigned to text chunks in the same sentence.

Summary: With the MapFace editor we provide the following features:

- A graphical user interface for the MMTx program.
- Automatic annotation of the document text.
- Navigation and visualization of the MMTx output.
- Modification of the tokenization of the text into phrase chunks and concept chunks.
- Possibilities to control, complete, and correct the affiliation of UMLS concepts to concept chunks.
- Possibilities to assign semantic types to phrase chunks.
- Advising the user of needs for corrections.
- Access to existing relations between UMLS concepts assigned to concept chunks of the text.
- Access to existing relations between semantic types assigned to concept chunks or phrase chunks of the text.
- Support of a better understanding of medical concepts and coherences within the text.

6. Usability evaluation

In order to assure the usability of the MapFace editor, a usability evaluation was conducted to gather constructive feedback about its design. We went for a heuristic evaluation approach according to Nielsen (1994) and Nielsen and Molich (1990). Advantages of this inspection method include the application of established usability principles, its intuitiveness, and its ability for effective identification of major and minor usability problems while keeping the costs for required time and equipment to a minimum (Holzinger, 2005).

The heuristic evaluation is focused on the detection of usability problems of the user interface design with respect to 10 given categories (e.g., user control and freedom, aesthetic design, information structuring, and consistency of the terminology and of the interaction mechanisms), thus, medical domain knowledge is not required to evaluate the usability of the user interface. It is commonly assumed that 3–5 expert evaluators (i.e., experts in usability testing) are sufficient for a heuristic usability evaluation (Holzinger, 2005).

6.1. The setting

We conducted a study with four usability experts, who were asked to solve typical tasks by means of the MapFace editor. They took note of every usability problem they encountered

during the session and rated the severity of the problem on a scale from 1 to 5, where 5 represents the highest severity.

At the time of the study, the evaluators were not familiar with the MapFace editor and its underlying tools in order to illustrate the behavior of a non-trained user. To ensure unbiased evaluations, separate testing sessions were performed with each evaluator, an observer being present at each testing session to answer questions about the domain and to give hints when the evaluator was clearly in trouble—but only after commenting the usability problem.

Each evaluator went through the interface of the MapFace editor two times. The first round was aimed at getting a feeling for the flow and the general scope of the user interface. In the second round the evaluator was supposed to focus on interactive interface elements with respect to a given list of Nielsen's established usability principles (see Table 2).

Each evaluator was asked to solve a given list of typical tasks by means of the MapFace editor and to enter the found problems on a list with reference to the violated usability principles. Additionally, the evaluators rated the severity of the problem (1 representing the lowest severity and 5 the highest). The evaluators were asked to solve the following tasks:

1. Processing the text (or parts of the text) of a document with the MMTx program.
2. Modifying the tokenization of the processed text into phrase chunks, if necessary.
3. Modifying the tokenization of the processed text into concept chunks, if necessary.
4. Controlling whether each concept chunk is annotated with the correct UMLS concept.
5. Modifying the annotated information of concept chunks, if necessary.
6. Assigning a semantic type to phrase chunks.

6.2. Usability problems

The total number of usability problems found is 32; their distribution according to their average severity—in case more than one evaluator encountered and rated the problem—is given in Table 1.

In Table 2 we list the usability principles according to Nielsen (1994) and the number of problems related to each principle. Most problems encountered (i.e., 25%) violate the principle “flexibility and efficiency of use”, for instance, the *necessity to frequently change selection tools*. So far, the MapFace editor provides two selection tools for different purposes: one tool to select already annotated text chunks in the editor and another tool to select unprocessed text of the document (e.g., to process the text by means of the MMTx program). However, while correcting the output of the MMTx program the user has to switch the active selection tool frequently, which is cumbersome and slows down the workflow.

Other problems that violate this principle are for instance, a *lack of shortcuts for frequently used actions* and *some features should be combined in a single button*.

The second largest group of usability problems (18.75%) relate to the principle “visibility of system status”, including problems of *insufficient explanations of some operations*, *little visibility of the results of selected operations*, and *insufficient progress information*.

Another frequently mentioned problem is the *necessity to switch between concepts mode and phrases mode*. The annotation of text chunks is carried out at two levels, i.e., at a

concept level and at a phrase level. Hence, the editor provides two modes, the concepts mode and the phrases mode. Correcting the output of MMTx calls for a basic idea about what presents a concept and what presents a phrase, and additionally, it requires frequent switching between these two modes. However, the advantages and shortcomings of merging these two modes into one, which would allow the user to simultaneously deal with both, concept chunks and phrase chunks, have to be carefully examined; it might well lead to a confusing representation of matters. In addition, more sophisticated ways of selection would have to be realized, which in turn could adversely affect the usability of the editor. Nevertheless, this problem wants additional consideration in order to find a solution that ensures a convenient workflow (see Section 7.1).

In addition, we categorized the usability problems according to the effort it takes to fix them.

1. 43.75% of all found problems take only little effort to be fixed, for instance,
 - Some information text should be changed.
 - Some features should be active in the first place.
2. 34.37% of all found problems take a moderate effort to be fixed, for instance,
 - Shortcuts should be provided for frequently used functions.
 - Some features might be combined in a single button.
3. 21.88% of all found problems take more effort to be fixed. The two most interesting problems of this category are described in detail below.

7. Further work

As mentioned before, the MapFace editor is to be extended by a component that detects negated phrases (Gindl et al., 2008) as well as by a component to detect coreferences within a medical text. By means of information extraction methods, negated phrases and co-references will be identified and subsequently highlighted in the editor. Again, unreliable or ambiguous results will be indicated, and features to modify the annotation will be provided.

Besides these extensions of the editor, we will eliminate the problems encountered during the usability inspection, as described in the following subsection. In addition, we give some examples of the further processing of annotated medical texts in Section 7.2.

7.1. Planned improvements

With respect to the outcome of the usability evaluation, we plan to realize the following improvements to eliminate the problems encountered.

Category (1)—Problems of this category take little effort to be fixed. Changes such as improving the explanations of some operations, removing unnecessary dialogs, and adding some explanatory text to the FAQ document can immediately be implemented in order to eliminate these problems.

Category (2)—Problems of the category (2) take a moderate effort to be fixed. However, changes of the user interface which are necessary to solve these problems can easily be accomplished, for instance, improving the visibility of the results of selected operations, implementing progress information for longer lasting operations, implementing shortcuts for frequently used operations, as well as merging some frequently combined sequences of operations into one operation step.

Category (3)—Problems of category (3) will either take some extra effort to be fixed, or they require more general changes in the design of the user interface. Examples:

- Implementing means to split up a given phrase chunk or concept chunk in one step by placing the text cursor at the desired split point and subsequently clicking a button.
- Enabling and disabling buttons in accordance with the selection of appropriate/inappropriate text chunks to perform the operation, for instance, enabling the “merge phrases” button only if two adjacent phrase chunks are selected in the editor.
- Eliminating the phrases mode and instead making phrases accessible through the shift key can solve the necessity of frequently switching between concepts mode and phrases mode.
- Eliminating the text selection cursor and combining both functions—selecting an annotated text chunk by double-clicking and selecting text by dragging the cursor—in one tool, such as an arrow cursor. This would do away with frequently having to change the selection tool in the toolbar or menubar.

Though the heuristic usability evaluation approach has been a good choice to identify shortcomings of the user interface at this stage of development, further usability tests have to be conducted to exhaustively evaluate the complete design. It is recommended to combine a heuristic evaluation with a usability test method, such as a cognitive walkthrough (Holzinger, 2005).

7.2. Subsequent processing of annotated documents

By means of the MapFace editor medical texts are thoroughly annotated, controlled, and corrected, which permits semantic annotation of high quality. These annotations allow for the immediate identification and automated retrieval of any given medical concept (e.g., symptoms, disease, diagnostic findings, and therapeutic procedures) within these texts, which in turn is mandatory for the effective computer processing of medical documents, for instance

- automatic generation of computer-interpretable clinical practice guidelines on the basis of textual guidelines;
- automatic retrieval of medical patient records (e.g., scanning the records for given findings or for disease patterns); as well as
- automatic indexing of medical documents (e.g., for document retrieval).

Furthermore, the accurate semantic annotation of medical texts enables professionals other than medical experts to properly understand these texts, for instance knowledge engineers, who have to correctly translate medical recommendations into computer-executable statements.

8. Conclusion

The MapFace editor is an important and useful means to create, visualize, and edit the semantic annotation of a medical document. Using the MapFace editor enables medical experts to automatically annotate the document by means of the MMTx program, to easily navigate the annotated information, and to modify the annotation both at a semantic as well as at a syntactic level. To do so without the MapFace editor would be a very cumbersome task requiring not only some skill in programming, but also an enormous amount of working hours. The MapFace editor is intended to visualize all kinds of annotated information,

important for processing a medical document. Controlling and correcting the annotation by means of the MapFace editor ensures the quality of the outcome, which in turn is imperative for the validity of the outcome of any subsequent processing step. With respect to the results of the usability evaluation, we will ensure that the MapFace editor is not only a useful and time-saving means, but also an intuitive and convenient tool to work with.

Acknowledgments

The research leading to these results has received funding from “Fonds zur Förderung der wissenschaftlichen Forschung FWF” (Austrian Science Fund), Grant L290-N04 and from the European Community’s Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 216134.

References

- Aronson, AR. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. Proceedings of the 25th Annual American Medical Informatics Association Symposium (AMIA 2001); AMIA, Washington, DC. 2001; p. 17-21.
- Berners-Lee T, Hendler J, Lassila O. The Semantic Web: a new form of web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American*. 2001; 284(5):34-43. [PubMed: 11396337]
- Bou B. last accessed: November 13, 2008 Stanford parser grammatical relation browser. 2008 (<http://grammarbrowser.sourceforge.net/>), last updated: November 1, 2008
- Browne, AC.; McCray, AT.; Srinivasan, S. The Specialist Lexicon. Lister Hill National Center for Biomedical Communications, National Library of Medicine; Bethesda, MD: 2000. The SPECIALIST Lexicon Technical Report 6/2000
- Chen Z, Perl Y, Halper M, Geller J, Gu H. Partitioning the UMLS Semantic Network. *IEEE Transactions on Information Technology in Biomedicine (IEEE TITB)*. 2002; 6(2):102-108.
- Ciravegna, F.; Dingli, A.; Petrelli, D.; Wilks, Y. User-system cooperation in document annotation based on information extraction. In: Gomez-Perez, A.; Benjamins, VR., editors. Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02). Ontologies and the Semantic Web; Springer, London, UK. 2002; p. 122-137.
- Côté, RA.; Rothwell, DJ.; Palotay, JL.; Beckett, RS.; Brochu, L. The Systematized Nomenclature of Medicine. SNOMED International, College of American Pathologists; Northfield, IL: 1993.
- Cunningham, H.; Maynard, D.; Bontcheva, K.; Tablan, V. GATE: an architecture for development of robust HLT applications. In: Isabelle, P., editor. Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL’02); Association for Computational Linguistics, Morristown, NJ. 2002; p. 168-175.
- Denny JC, Smithers JD, Miller RA, Spickard A. Understanding medical school curriculum content using KnowledgeMap. *Journal of the American Medical Informatics Association*. 2003; 10(4):351-362. [PubMed: 12668688]
- Elkin, PL.; Cimino, JJ.; Lowe, H. Mapping to MeSH: the art of trapping MeSH equivalence from within narrative text. In: Greenes, RA., editor. Proceedings of the 12th Annual Symposium on Computer Applications in Medical Care (SCAMC 1988); IEEE Computer Society Press, Los Alamitos, CA. 1988; p. 185-190.
- Field, MJ.; Lohr, KN. Clinical Practice Guidelines: Directions for a new program. National Academy Press; Washington, DC: 1990.
- Gindl, S.; Kaiser, K.; Miksch, M. Syntactical negation detection in clinical practice guidelines. In: Andersen, SK.; Klein, GO.; Schulz, S.; Aarts, J.; Mazzoleni, MC., editors. Proceedings of the 21st International Congress of the European Federation of Medical Informatics MIE 2008, eHealth Beyond the Horizon—Get IT There; IOS Press, Amsterdam, The Netherlands. 2008; p. 187-192.
- Holzinger A. Usability engineering methods for software developers. *Communications of the ACM*. 2005; 48(1):71-74.
- Levy, R.; Manning, CD. Deep dependencies from context-free statistical parsers: correcting the surface dependency approximation. In: Scott, D., editor. Proceedings of the 42nd Annual Meeting of the

- Association for Computational Linguistics (ACL'04); Association for Computational Linguistics, Morristown, NJ. 2004; p. 327
- Lindberg DA, Humphreys BL, McCray AT. The Unified Medical Language System. *Methods of Information in Medicine*. 1993; 32(4):281–291. [PubMed: 8412823]
- McCray, AT. UMLS Semantic Network. *Proceedings of the 13th Annual Symposium on Computer Applications in Medical Care (SCAMC 1989)*; IEEE Computer Society Press, Washington, DC. 1989; p. 503-507.
- McCray, AT.; Burgun, A.; Bodenreider, O. Aggregating UMLS semantic types for reducing conceptual complexity. *Proceedings of the 10th World Congress on Medical Informatics, 84 Studies in Health Technology and Informatics (MEDINFO '01)*; IMIA, IOS Press, Amsterdam, The Netherlands. 2001; p. 216-220.
- McCray AT, Nelson SJ. The representation of meaning in the UMLS. *Methods of Information in Medicine*. 1995; 34(1–2):193–201. [PubMed: 9082131]
- National Institute of Health. last accessed: Okt 17, 2008 *Computer Retrieval of Information on Scientific Projects (CRISP)*. 2006 <http://crisp.cit.nih.gov>
- Nielsen, J. *Usability Inspection Methods*. Wiley; New York: 1994. Heuristic evaluation; p. 25-62.
- Nielsen, J.; Molich, R. Heuristic evaluation of user interfaces. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 1990)*; ACM Press, New York, NY. 1990; p. 249-256.
- Peleg M, Tu SW, Bury J, Ciccarese P, Fox J, Greenes RA, Hall R, Johnson PD, Jones N, Kumar A, Miksch S, Quaglini S, Seyfang A, Shortliffe EH, Stefanelli M. Comparing computer-interpretable guideline models: a case-study approach. *Journal of the American Medical Informatics Association*. 2003; 10(1):52–68. [PubMed: 12509357]
- Reeve, LH. *Semantic annotation and summarization of biomedical text*. Drexel University, College of Information Science and Technology; Philadelphia, PA: 2007. Ph.D. Thesis
- Schlyer PL, Hole WT, Tuttle MS, Sherertz DD. The UMLS Metathesaurus: representing different views of biomedical concepts. *Bulletin of the Medical Library Association*. 1993; 81(2):217–222. [PubMed: 8472007]
- Srinivasan, S.; Rindfleisch, TC.; Hole, WT.; Aronson, AR.; Mork, JG. Finding UMLS Metathesaurus concepts in MEDLINE. *Proceedings of the 26th Annual American Medical Informatics Association Symposium (AMIA '02)*; AMIA, Washington, DC. 2002; p. 727-731.
- Tuttle M, Olson N, Keck K, Cole W, Erlbaum M, Sherertz D, Chute C, Elkin PL, Atkin G, Kaihoi B, Safran C, Rind D, Law V. Metaphrase: an aid to the clinical conceptualization and formalization of patient problems in healthcare enterprises. *Methods of Information in Medicine*. 1998; 37(4–5): 373. [PubMed: 9865035]
- Vargas-Vera, M.; Motta, E.; Domingue, J.; Lanzoni, M.; Stutt, A.; Ciravegna, F. MnM: ontology driven tool for semantic markup. In: Handschuh, S.; Collier, N.; Dieng, R.; Staab, S., editors. *Proceedings of the ECAI 2002 Workshop on Semantic Authoring, Annotation & Knowledge Markup (SAAKM 2002)*; ECCAI, Lyon, France. 2002; p. 43-47.
- Wang D, Peleg M, Tu SW, Boxwala AA, Greenes RA, Patel VL, Shortliffe EH. Representation primitives, process models and patient data in computer-interpretable clinical practice guidelines: a literature review of guideline representation models. *International Journal of Medical Informatics*. 2002; 68(1–3):59–70. [PubMed: 12467791]
- Ware, C. *Information Visualization—Perception for design*. second ed.. Morgan Kaufmann; San Francisco, CA: 2004.

Fig. 1.
 Annotation editors. The user interfaces of related editors. (a) GATE (Cunningham et al., 2002). (b) MnM ontology-based markup component (Vargas-Vera et al., 2002). (c) Melita annotation component (Ciravegna et al., 2002). (d) Stanford parser browser (Bou, 2008).

Phrase: “the most effective preventer drug.”

Meta Candidates (13)

- 812 Drug (Pharmaceutical Preparations) [Pharmacologic Substance]
- 756 Pharmaceutical [Intellectual Product]
- 719 Pharmaceutics (Pharmacy) [Biomedical Occupation or Discipline]
- 719 Medicament [Pharmacologic Substance]
- 645 Most [Quantitative Concept]
- 645 Effective (Effect) [Qualitative Concept]
- 574 PREVENT [Pharmacologic Substance]
- 574 Effectiveness [Qualitative Concept]
- 574 Prevents [Functional Concept]
- 545 Prevention (Prophylactic treatment) [Therapeutic or Preventive Procedure]
- 545 Prevention [Therapeutic or Preventive Procedure]
- 545 Preventable [Intellectual Product]
- 530 Preventive (Preventive intent) [Qualitative Concept]

Meta Mapping (815)

- 645 Most [Quantitative Concept]
- 645 Effective (Effect) [Qualitative Concept]
- 574 Prevents [Functional Concept]
- 812 Drug (Pharmaceutical Preparations) [Pharmacologic Substance]

Meta Mapping (815)

- 645 Most [Quantitative Concept]
- 645 Effective (Effect) [Qualitative Concept]
- 574 PREVENT [Pharmacologic Substance]
- 812 Drug (Pharmaceutical Preparations) [Pharmacologic Substance]

Fig. 2.
The original output of MMTx. These are the results of the MMTx program for the phrase “the most effective preventer drug”.

Fig. 3.

The Mapping created by the MMTx program. MMTx tokenizes the input sentence into phrase chunks; subsequently, MMTx maps medical concepts from the UMLS Metathesaurus to text chunks. The encircled objects are wrong or ambiguous results of the MMTx program which have to be corrected by means of the MapFace editor (see Fig. 4).

Fig. 4.

Corrections accomplished by means of the MapFace editor. Compare Fig. 3: disambiguation of the mapping of UMLS concepts and semantic types to the text chunks “years”, “steroids”, and “preventer”. In addition, the tokenization of the phrase chunks “with mild asthma inhaled steroids” has been corrected and instead of the two concept chunks “five” and “years” we prefer a single concept chunk annotated with the UMLS concept age.

Fig. 5.
Simplified class diagram of the data structure used in MapFace.

Fig. 6.

The user interface of the MapFace editor. The user interface is divided into three panes: the *editor pane*, the *annotation scheme pane*, and the *candidates pane*.

Fig. 7.

The view of the *semantic types annotation scheme*. This view contains a list of semantic types grouped by semantic collections (Chen et al., 2002), whereas each semantic collection is associated with a unique color.

Fig. 8. Indicating ambiguous concept matches in the editor. In case there are two or more equally matching UMLS concept candidates for one concept chunk, the text chunk is marked by a gray background.

Fig. 9. Information assigned to phrase chunks. The user manually chooses one of the semantic types assigned to the concept chunks within the phrase and assigns it to the phrase chunk.

Candidate	Semantic Type	Semantic Collection	Semantic Group
Abuse of steroids	Mental or Behavioral Dysfunction	Pathologic Function	Disorders
Steroids	Steroid	Chemical	Chemicals & Drugs

Fig. 10.

The *candidates view*. The *candidates view* displays a list of equally matching UMLS concept candidates for each concept chunk that could not be mapped unambiguously. The displayed UMLS concept candidates match the text chunk “steroids” (see Fig. 8).

Fig. 11. Relations between two semantic types. The *semantic relations* view displays a list of relations of the semantic type of a selected text chunk to semantic types assigned to other text chunks in the same section of the text. On selection of a specific relation, the corresponding text chunks are highlighted in the *editor pane*. By now, the concept chunk “Ovarian cancer” is not annotated, but the semantic type *neoplastic process* is a possible candidate. The examination of relations of all candidates for the concept chunk “Ovarian cancer” to other concept chunks in the same section facilitates the correct decision.

Table 1

Usability problems

Average severity	1	2	3	4	5
Problems	2	6	17	4	3

The distribution of usability problems according to their average severity (5 representing the highest severity).

Table 2

Usability principles (Nielsen, 1994; Nielsen and Molich, 1990).

Usability principle	Number of problems
Visibility of system status	6
Match between system and the real world	3
User control and freedom	1
Consistency and standards	4
Error prevention	2
Recognition rather than recall	3
Flexibility and efficiency of use	8
Aesthetic and minimalist design	2
Help users recognize, diagnose, and recover from errors	2
Help and documentation	1

The distribution of usability problems according to the violated usability principle.