A Framework to Generate Sets of Terms from Large Scale Medical Vocabularies for Natural Language Processing

Salah Aït-Mokhtar Caroline Hagège Pajolma Rupi
Xerox Research Centre Europe*
Firstname.Lastname@xrce.xerox.com

Abstract

In this paper we present our ongoing work on integrating large-scale terminological information into NLP tools. We focus on the problem of selecting and generating a set of suitable terms from the resources, based on deletion, modification and addition rules. We propose a general framework in which the raw data of the resources are first loaded into a knowledge base (KB). The selection and generation rules are then defined in a declarative way using query templates in the query language of the KB system. We illustrate the use of this framework to select and generate term sets from a UMLS dataset.

1 Introduction

Information extraction from free medical text using Natural Language Processing (NLP) is currently an important field considering the huge and still growing amount of unstructured textual documents in the medical domain (patient data, clinical trials and guidelines, medical literature). The ability to process automatically the information expressed in these documents can help to bridge gaps between patient information and clinical literature and it can be an asset for a wide range of applications in medical Information Extraction (IE) and NLP. The first step for effective processing of medical free text is the recognition of medical terms[1] appearing in these documents. For better interoperability, this annotation should be as compatible as possible with reference vocabularies and ontologies of the domain. Nonetheless, the terms from such resources require filtering and transformation before they are integrated into annotation tools.

In this paper we present our work on the process of selecting the set of terms to be integrated in the NLP component. We will briefly review the state of the art, and then describe the framework we developed for declarative and easily-maintainable selection and adaptation of term sets, based on a knowledge base (KB) system and query language. We will illustrate the use of this framework to select and generate term sets from an initial UMLS dataset.

2 Related Work

Several tools for annotating terms in medical text are currently available. MetaMap (Aronson and Lang (2010)) uses the Metathesaurus information in the Unified Medical Language System (UMLS) (Bodenreider et al. (2004) and Bodenreider (2007)) in order to automatically determine the medical concepts referred to in text. MetaMap relies on the SPECIALIST Lexicon of UMLS (NLM (2009); Browne et al. (2000)), a general English lexicon which includes both medical terms and general domain lexical entries. cTAKES (Savova et al. (2010)) performs information extraction from clinical narratives. It consists

*This work has been done in the context of the European FP7-ICT EURECA project: http://eurecaproject.eu/.

[1]Throughout this paper, we use the word "term" to refer to any character string that denotes a medical concept or entity, except in the figure depicting the UMLS ontology model (figure 1) where "Term" has the UMLS term class meaning.

of sequential components contributing to an incremental and cumulative annotation dataset. MedKAT (Medical Knowledge Analysis Tool)[2] is an NLP tool dedicated to the medical/pathology domain. It consists of a set of modules which annotate unstructured data such as pathology reports, clinical notes, discharge summaries and medical literature. These annotation tools use UMLS, BioPortal[3] or in-house built vocabularies as sources for the building of medical lexicons. Harkema et al. (2004) proposed a large scale terminology system for storing terms, which are compiled into finite-state lexicons to perform term lookup in texts. For that goal, they set up a relational database where terms from different origins are kept. This approach has the advantage of centralizing all the terminological information one may want to use for different applications. The terms stored in the database are then extracted and compiled into finite state lexicons. However, the work did not cover the issue of filtering and transforming the original set of terms before their inclusion into the NLP component.

Because of the size and the variety of existing medical vocabularies and medical texts, the ability to select and adapt the terminological information can help improve effective NLP tools, as reported by Demner-Fushman et al. (2010). Hettne et al. (2010) and Wu et al. (2012) have also shown that term filtering operations are useful in building an adequate medical lexicon. Hettne et al. (2010) conducted experiments for the building of a medical lexicon using UMLS Metathesaurus. They use term suppression and term rewriting techniques to filter out or discard terms which are considered irrelevant. As a result, a new, more consolidated medical lexicon is produced for medical concept annotation. In Wu et al. (2012) the UMLS Metathesaurus terms characteristics are exploited for discovering which of them are generalizable across data sources. After a corpus study, they came out with a set of filtering rules that significantly reduced the size of the original Metathesaurus lexicon.

In order to implement these selections and adaptations of term sets from existing medical vocabularies in a declarative and easily-maintainable way, we propose a framework based on an ontological representation and on knowledge-base query templates that define selection and adaptation rules.

3 A general framework for generating medical lexical resources

In the general framework we propose, an ontological schema is defined to capture the information contained in the terminological resources, according to which the raw data of resources are imported and loaded into an efficient knowledge base (KB) system in the form of entities and relations (RDF-like triples/quads). Depending on the requirements of the foreseen NLP-based application, the user defines the set of terms to generate from the terminological KB by writing a set of query templates in the query language of the KB system. Each query template can be tagged as a **deletion**, **modification** or **addition**.

Deletion query templates contain a predefined unbound variable T that can be instantiated with a term from the KB: if the resulting query runs successfully, then the term should be deleted from the final output term set. Similarly, modification and addition query templates have two unbound variable T and NT: when variable T is instantiated with a term from the KB and the resulting query succeeds, variable NT is instantiated. The resulting values of NT are new terms that should replace the original term T in the case of modification queries, or should be added along with the original term to the output term set. The user provides the set of query templates as parameters to the term selection and generation engine. The engine iterates through all the terms of the KB: for each term, it instantiates the input variable T of each query template with the term. Deletion queries are tested first: if one of them succeeds for the current term, then the term is discarded from the output term set and the engine goes to the next KB term. If not, and if one of the modification queries succeeds for the term, then the engine adds the output value of the query (i.e. all possible values of variable NT) to the output term set. Finally, if one of the addition rule succeeds, then it adds both the original KB term and the output terms (i.e. all possible values of variable NT) to the output set. A new term is always assigned all the information of the original term from which it is produced (i.e. same concept(s), semantic type(s), etc.), together with a specific tag.

[2] http://ohnlp.sourceforge.net/MedKATp/

[3] http://bioportal.bioontology.org/

We use an in-house KB system, called SKB, to store all the data, and its query language to define the query templates. An example of a deletion rule is: discard any term that has more than 6 tokens (see section 4.3.2). It can be defined with the following query making use of a regular expression:

```
regex(T "(\S+\s){6,}\S+")
```

A more interesting case is the "semantic type" modification rule (see table 2, section 4.3.2), which removes any semantic type within parentheses inside the initial term: in the following query template, the $regex$ part captures the semantic type substring (captured group $\backslash 2$), checks that it's the name of a KB node n that represents a UMLS semantic type (i.e. has the "hasTreeNumber" property), and instantiates output variable NT by deleting that substring (and the parentheses) from the initial term T, and finally checks that the new term NT is not already assigned to the same concept c in the initial UMLS data:

```
regex(T ".*?( *\(([^\)]+)\) *).+") & n=@findEntity(\2) & umls:hasTreeNumber(n ?)
& NT=@replace(T " *\(([^\)]+)\)" "") & umls:hasName(c T) & ~umls:hasName(c NT)
```

By using such a KB storage and query language, the system we propose has the advantage of providing a clean, modular and declarative way to define, maintain and change the criteria of selecting and generating terms from large-scale terminological resources. There is no need to code the transformation and selection rules in a programming language. The only requirements are that: (a) the original terminological resource is loaded into a KB (this is done once), and (b) the query language of the KB system has to be powerful enough to make it possible to use regular expression matching and back references, and string related functions and operators (e.g. concatenation, length, replacement) inside the queries. As a matter of fact, we did not choose a relational DBMS and SQL to implement the system because some of the relevant selection and transformation rules cannot be implemented with single SQL queries.

4 An example: Extracting UMLS information for NLP tools

4.1 UMLS dataset

We use UMLS as the basis for the creation of medical lexicons. UMLS combines a variety of source vocabularies, ontologies and terminologies. Integration of over 100 sources generates a very large medical knowledge base. In our work, we use all the English terms of category 0 of the 2012AA release of UMLS (i.e. licence-free vocabularies). The subset consists of 46 different vocabularies and contains 3.97 million English terms referring to 1.9 million concepts.

4.2 Defining an ontology and loading UMLS data to the KB system

We want to provide declarative ways to specify criteria for terms and concepts that are relevant for the medical lexicon we aim to build. These criteria may include meta-information such as the source name or the category of the source, but also a selection by language, semantic types or linguistic characteristics of the term. We define an ontology model (see figure 1) that incorporates the **complete** knowledge about concepts, terms, semantic types and relations, contained in the UMLS Metathesaurus and Semantic Network. The aim of building this ontology is to have easily traversable structured information.

We developed a Perl program which parses the main UMLS files, extracts the information according to the ontology model and transforms it into a triples/quads that are loaded into the KB. We produced 139.7 million of triples for category 0 data (all languages. The data is then loaded into the KB, where it can be further explored before being compiled into finite-state lexicon.

4.3 Transformation of the set of medical terms

We exclude terms from the initial dataset on the basis of semantic types. We also take advantage of previously published work (Hettne et al. (2010); Wu et al. (2012)) to select the most useful cleaning

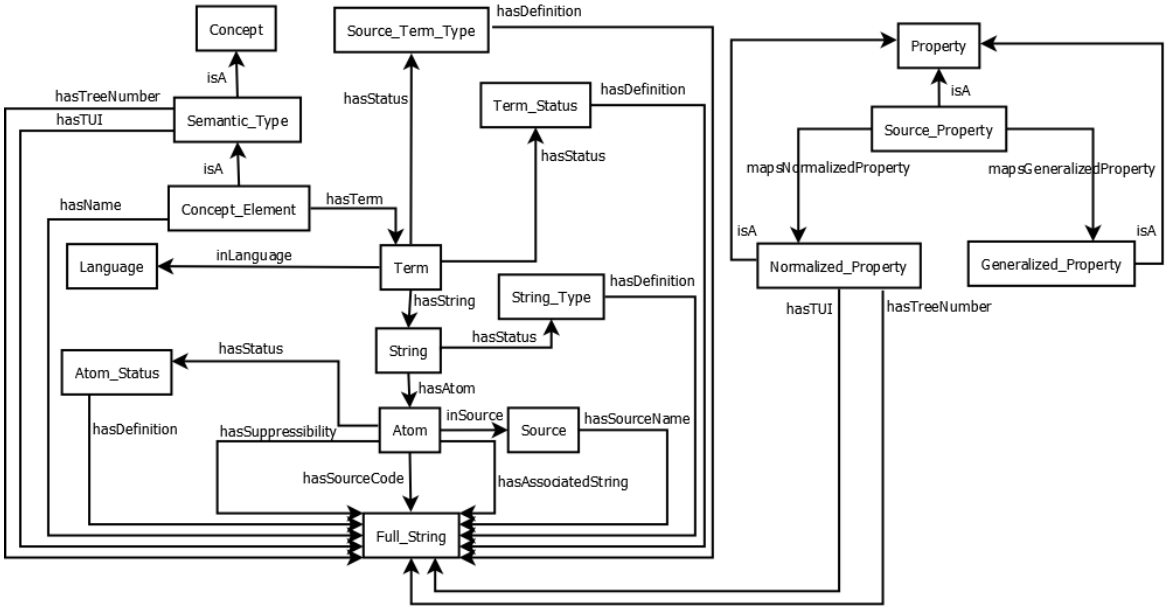

Figure 1: Ontology model

and transformation rules, and implement them with a few changes motivated by linguistic observation and experiments with the data. We compile the resulting terms with their UMLS semantic types into a finite-state transducer (FST), which is then unioned with the general lexical FST of the NLP components.

4.3.1 Cleaning by semantic types

Because we integrate the medical terms into a larger NLP system, we do not keep general domain information (i.e. information not specific to the medical domain). We discard terms belonging exclusively to UMLS concepts that have generic semantic types, e.g. semantic types corresponding to classical named entity types (like Organization and Geographic Area): the linguistic tools we rely on include a named entity recognition system that already captures this information in a more systematic way.

4.3.2 Transformation rules from the state of the art

We implemented term filtering rules described in Hettne et al. (2010) and Wu et al. (2012), using KB query templates described in 3. There are three main types of rules: deletion, addition and modification rules. The effect of each rule on the initial dataset is presented in tables 1, 2 and 3.

Table 1: Impact of deletion rules: number of deleted terms per rule

Rule	# deleted	Ex. of deleted term
Short token	893	*"9394"*
Dosages	388,019	*"Ampicillin 10g injection"*
At-sign	249,381	*"Medical and Surgical @ Muscles @ Transfer"*
EC	195	*"EC 1.1.1.62"*
Any classification	3,948	*"Unclassified ultrastructural features"*
Any underspecification	1,302	*"Unspecified amblyopia"*
Miscellaneous	9,625	*"Other causes of encephalitis"*
Special characters	45	*"[M]Brenner tumors (morphologic abnormality)"*
Maximum number of words	847,136	*Any term with 7 or more tokens*
Maximum number of characters	787,788	*Any term with more than 55 characters*

Table 2: Impact of modification rules: number of affected terms and number of resulting terms per rule

Rule	# matches	# resulting terms	Ex. impacted term	Ex. new term
Angular brackets	2,666	1,620	*"Bacteria <prokaryote>"*	*"Bacteria"*
Semantic type	419	411	*"Insulin (human) extract"*	*"Insulin extract"*

Table 3: Impact of addition rules: number of matching terms and number of new terms per rule

Rule	#matches	#new	Ex. matching term	Ex. output term(s)
Syntax inversion	494,518	482,236	*"GIST, malignant"*	*"malignant GIST"*
Possessives	7,263	7,263	*"Addison's Disease"*	*"Addison Disease"*
Short/long form	32,351	32,129	*"AD - Alzheimer's disease"*	*"AD", "Alzheimer's disease"*

4.3.3 Discussion of the transformation rules

Some of the filtering criteria adopted from Hettne et al. (2010) and Wu et al. (2012) have been slightly modified. We changed the ***Short token*** original rule to avoid filtering out relevant terms containing only one letter, like "*B*" (i.e. *Boron*, concept C0006030), or "*C*" (i.e. *Catechin*, concept C0007404). The ***Short form/long form*** (acronym/full term) original rule proposed by Hettne et al. (2010) used the algorithm in Schwartz and Hearst (2003). However, we found many cases not covered by the proposed algorithm, because the acronyms are not always built strictly from the first upper-case letters of the tokens of the terms tokens: e.g. "Cardiovascular Disease (CVD)", "Von Hippel-Lindau Syndrome (VHL)". Besides, there are UMLS terms in which the short form is at the beginning of the term. For example: "AD - Alzheimer's disease", "ALS - Amyotrophic lateral sclerosis", "BDR - Background diabetic retinopathy". We adapted the **short form/long form** rule accordingly.

5 Conclusion

We presented a framework for selecting and modifying large amounts of medical terms and integrating them into NLP lexicons[4]. Terms are first extracted from existing medical vocabularies present in the UMLS and stored into a knowledge base which preserves the original information associated to these terms (ontological and relational information). The way we store this information is in line with current trends of the semantic web and linked data. We took advantage of the powerful query language of a KB system in order to define filtering, suppression and transformation operations on the original terms. The most important characteristic of the approach is that this is performed in a declarative way, even for operations such as term modification. Consequently, the creation of new medical vocabularies for different NLP applications is easier than with programming-based methods. Finally, finite-state transducers containing these extracted and modified terms are first created and then combined with general purpose lexicons. The next stage will be to use and evaluate NLP tools relying on these lexicons.

References

Aronson, A. R. and F.-M. Lang (2010). An overview of MetaMap: historical perspective and recent advances. *JAMIA 17*(3), 229–236.

Bodenreider, O. (2007). The Unified Medical Language System (UMLS) and the Semantic Web. "`http://www.nettab.org/2007/slides/Tutorial_Bodenreider.pdf`".

[4]The system is licensable to clinical NLP community members.

Bodenreider, O., J. Willis, and W. Hole (2004). The unified medical language system. http://www.nlm.nih.gov/research/umls/presentations/2004-medinfo_tut.pdf.

Browne, A. C., A. T. McCray, and S. Srinivasan (2000). *The SPECIALIST LEXICON*. Lister Hill National Center for Biomedical Communications, National Library of Medicine.

Demner-Fushman, D., J. G. Mork, S. E. Shooshan, and A. R. Aronson (2010, August). UMLS content views appropriate for NLP processing of the biomedical literature vs. clinical text. *Journal of Biomedical Informatics 43*(4), 587–594.

Harkema, H., R. Gaizauskas, M. Hepple, A. Roberts, I. Roberts, N. Davis, and Y. Guo (2004, May). A large scale terminology resource for biomedical text processing. In *Proceedings of the NAACL/HLT 2004 Workshop on Linking Biological Literature, Ontologies and Databases: Tools for Users*, Boston.

Hettne, K., E. van Mulligen, M. Schuemie, B. Schijvenaars, and J. Kors (2010). Rewriting and suppressing UMLS terms for improved biomedical term identification. *Journal of Biomedical Semantics 1*(1), 5.

NLM (2009). SPECIALIST Lexicon and Lexical Tools. National Library of Medicine, National Institutes of Health, Bethesda, MD 20894, USA. http://www.ncbi.nlm.nih.gov/books/NBK9680/.

Savova, G. K., J. J. Masanz, P. V. Ogren, J. Zheng, S. Sohn, K. C. Kipper-Schuler, and C. G. Chute (2010). Mayo clinical text analysis and knowledge extraction system (cTAKES): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association 17*(5), 507–513.

Schwartz, A. S. and M. A. Hearst (2003). A simple algorithm for identifying abbreviation definitions in biomedical text. In *Proceedings of the Pacific Symposium on Biocomputing*, pp. 451–462.

Wu, S. T.-I., H. Liu, D. Li, C. Tao, M. A. Musen, C. G. Chute, and N. H. Shah (2012). Unified Medical Language System term occurrences in clinical notes: a large-scale corpus analysis. *Journal of the American Medical Informatics Association 19*(e1), e149–e156.