Journal of Biomedical Informatics 46 (2013) 882–893

Contents lists available at SciVerse ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

Semantator: Semantic annotator for converting biomedical text to linked data

Cui Tao [a,b,*], Dezhao Song [b,c], Deepak Sharma [b], Christopher G. Chute [b]

[a] School of Biomedical Informatics, University of Texas Health Science Center at Houston, Houston, TX 77030, United States
[b] Division of Biomedical Statistics and Informatics, Mayo Clinic, Rochester, MN 55905, United States
[c] Department of Computer Science and Engineering, Lehigh University, Bethlehem, PA 18017, United States

ARTICLE INFO

Article history:
Received 20 October 2012
Accepted 6 July 2013
Available online 15 July 2013

Keywords:
Clinical narratives
Semantic Annotation
Semantic web
Semantator
Protege plugin

ABSTRACT

More than 80% of biomedical data is embedded in plain text. The unstructured nature of these text-based documents makes it challenging to easily browse and query the data of interest in them. One approach to facilitate browsing and querying biomedical text is to convert the plain text to a linked web of data, i.e., converting data originally in free text to structured formats with defined meta-level semantics. In this paper, we introduce Semantator (Semantic Annotator), a semantic-web-based environment for annotating data of interest in biomedical documents, browsing and querying the annotated data, and interactively refining annotation results if needed. Through Semantator, information of interest can be either annotated manually or semi-automatically using plug-in information extraction tools. The annotated results will be stored in RDF and can be queried using the SPARQL query language. In addition, semantic reasoners can be directly applied to the annotated data for consistency checking and knowledge inference. Semantator has been released online and was used by the biomedical ontology community who provided positive feedbacks. Our evaluation results indicated that (1) Semantator can perform the annotation functionalities as designed; (2) Semantator can be adopted in real applications in clinical and transactional research; and (3) the annotated results using Semantator can be easily used in Semantic-web-based reasoning tools for further inference.

1. Introduction

As recent surveys indicated, more than 80% of patients seek health information on the Internet [3]; more than 70% of physicians regularly search online for medical or professional updates [19]. Approximately 80% of health care data, as well as the ever-growing data online, however, consist of unstructured narratives [14,18]. Efficiently querying and browsing data embedded in these biomedical documents is an important and challenging task. The unstructured nature of these text-based documents brings to light an inherent problem: locked within these documents lies an extraordinary amount of key biomedical knowledge and clinical data, which can hardly be leveraged without intensive manual work. Traditional search engines such as Google can return users the potential documents of interest based on keywords. Users still have to, however, read through the returned documents until the information of interest is located. In addition, search engines usually return hundreds of thousands of links, many of which are not relevant to users' search.

One approach to facilitate browsing and querying biomedical text is to convert the plain text into an annotated web of data, i.e., to convert data originally in free text into structured formats with defined meta-level semantics. Manual annotation may not be realistic due to the large volume of text that needs to be processed. Fully automatic approaches for semantic annotation do not always give satisfying results. Semi-automatic data annotation is, therefore, an attractive alternative. Semi-automatic annotation supports information from biomedical text to be automatically extracted and annotated with manual on refining the annotations.

To support semi-automatic annotation, we developed Semantator. Semantator is a user-friendly, semantic-web-oriented environment for annotating data of interest in biomedical documents with respect to domain ontologies. Domain ontologies have been used in information technology to provide semantic definitions of a particular domain, which enable automated agents to perform queries intelligently and infer new knowledge. An ontology includes a set of classes and their relationships (e.g., class hierarchies and predicates). Semantator provides an environment to link data embedded in text to ontology concepts by using semantic annotation. Information of interest from a document can be annotated as an instance of an ontology class to obtain all the semantic definition of that class. In addition, relations between instances can be cre-

* Corresponding author.
E-mail address: cui.tao@uth.tmc.edu (C. Tao).

1532-0464/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jbi.2013.07.003

ated using the predicates (properties) defined in the ontology. The annotation results are saved in the Resource Description Framework (RDF) [21] format, which provides a standard way for data sharing and exchange and enables querying and browsing the data using the SPARQL query language [24]. In addition, Semantator also provides an interface where users can compare annotations done by different curators or annotation tools, leverage semantic web technologies for inferences, and detect conflicts in annotations.

More specifically, Semantator is implemented as a Protégé [2] plug-in, which allows users to view the original documents, the ontology used for annotation, and the annotation results in the same environment. Semantator provides two modes: (1) manual annotation and (2) semi-automatic annotation. In the manual annotation mode, an expert can choose an annotation schema (a domain ontology), open a document to be annotated, highlight different pieces of information to be annotated, and then mark which ontology concepts the information belongs to. For each highlighted piece of data, the system will generate class instances according to the annotation and display different class instances in different colors. Relationships between instances can also be created using the properties defined in the domain ontology. For the semi-automatic annotation mode, Semantator provides an Application Programming Interface (API), which provides the option to connect the Semantator annotation environment to state-of-the-art or customized information extraction or semantic annotation tools. Human curators can review the automatic annotation results in the Semantator environment and modify them as needed.

The Semantator has been released through our web site: http://informatics.mayo.edu/CNTRO/index.php/Download_Semantator. In our previous publication [23], we reported the basic functionalities of Semantator: preliminary implementation of the manual annotation mode; and semi-automatic annotation using the clinical Text Analysis and Knowledge Extraction Systems (cTAKES) [22] and the NCBO annotator [16] (Section 3). This manuscript extends our previous work by introducing two new major functionalities: (1) rule-based extraction capacity (Section 4) and (2) the annotation result comparison function (Section 5). We analyze and illustrate the benefits of using semantic web technologies on the Semantator annotated data (Section 6). We have also conducted a functionality evaluation (Section 7.1) and applied Semantator in a real clinical research application as a case evaluation (Section 7.2). The evaluation results indicate that Semantator can successfully conduct the annotation tasks as designed. We have received much positive feedback and suggestions from the community, based on what we have already improved and will continually improve the functionalities of the tool (Section 8).

2. Related work

2.1. Annotation systems

Andrews et al. [4] has reviewed a number of annotation systems and classified them into four categories: tag-based, attribute-based, relation-based, and ontology-based. The annotation systems within the first three categories allow minimal annotation model representation, and therefore can only enable a limited number of services that mainly focusing on basic browsing and searching functions. Knowtator [17], for example, is a attribute-based annotation environment that is well adopted by the clinical Natural Language Processing (NLP) community. Brat [1], as another example, is a web-based annotation tool for collaborative text annotation. Compared to the annotation systems in the first three categories, ontology-based annotation systems, such as Semantator, can provide semantic annotations that describe a resource with respect to a formal conceptual model. These systems allow semantic queries and reasoning. In addition to Semantator, there are other ontology-based annotation systems. Semantic-document [11] and GoNTogle [12], for example, support semantic annotation on documents with ontology classes. Compared to these systems, Semantator further supports instance relationship creation and provides reasoning capabilities. KIM [20] is a commercial software that supports manual, automatic, and semi-automatic annotation for both instances and relationships. KIM, however, does not allow users to use their own domain ontologies for annotations.

2.2. Information extraction and annotation algorithms

Automatic annotation systems rely on different information extraction and annotation algorithms. Existing algorithms can be generally categorized into pattern-based systems and machine-learning-based systems. Pattern-based systems, such as PANKOW [7] and Armadillo [6], try to locate named entities by using patterns that are either manually defined or semi-automatically induced. SemTag [9] and KIM [20] use pre-defined rules to locate the information of interest. Alternatively, systems such as S-CREAM [15] and MnM [27] use machine learning and NLP-based techniques to identify named entities. Although machine-learning-based approaches do not fully rely on manually defined rules, they are usually supervised algorithms, which require certain amount of training data that need human efforts.

For the biomedical domain, there are several well-acknowledged information extraction or annotation systems. MetaMap [5], for example, is a system to map biomedical text to UMLS Metathesaurus. The clinical Text Analysis and Knowledge Extraction System (cTAKES) [22] focuses on annotating clinical narratives to standard ontologies and terminologies such as SNOMED CT and RxNorm using NLP and machine learning based approaches. The NCBO annotator [16] is a web service that helps to match biomedical text with ontology terms from one or more ontologies hosted in BioPortal (http://bioportal.bioontology.org/). Semantator provides an API for users to plug in and play state-of-the-art automatic annotation tools to connect them with domain ontologies.

3. Basic semantic annotation functions

In this section, we describe the basic annotation functionalities of Semantator, including creating and removing ontology instances, managing instance relationships, and annotating relationships. We also introduce how different automatic annotation tools can be embedded in the Semantator environment.

3.1. Instance and relationship annotation

3.1.1. Creating and removing ontology instances

To create instances, a user can highlight a piece of text and select a class from the domain ontology as demonstrated in Fig. 1. By default, Semantator will save the highlighted string using *rdfs:label* to the newly created instance. Users can also add document fragments that describe instances of the same type into a "batch", and create them together. When deleting ontology instances, Semantator will first detect all instances for which this document fragment has been created, and users can then delete one or more of them as needed.

3.1.2. Managing instance relationships

The relationships between ontology instances are represented by properties in the ontology. For example, $\langle Event1, before, Event2\rangle$ means *Event1* happened before *Event2*. To create a relationship, a user will select the two instances (Fig. 2) and the corresponding