OntoFast: Construct Ontology Rapidly

Abdul-Mateen Rajput[1], Marzio Pennisi[2], Santo Motta[2], and Francesco Pappalardo[3]

[1] Bonn-Aachen International Center for Information Technology [B-IT], University of Bonn, Germany
Abdul-Mateen.Rajput@uni-bonn.de
[2] Department of Mathematics and Computer Science, University of Catania, Italy
{mpennisi,motta}@dmi.unict.it
[3] Department of Drug Sciences, University of Catania, Italy
francesco.pappalardo@unict.it

Abstract. Ontology construction is a time consuming and labor intensive task. It may take many months to construct an ontology as according to standard practices each concept must have synonyms, domain specific definition, unique identifier and references. Current practices of ontology construction require manual data input to feed this data via programs such as Protégé etc. We designed a small application that speeds up the development of new ontologies. It provides an easy to use and convenient interface that allows to theoretically build an ontology within few days. The output of our program can be easily opened and then used into a standard ontology editor like Protégé. Availability: The software is freely available visiting this link: http://www.francescopappalardo.net/ontofast.zip.

Keywords: Ontology engineering, semantic web tools, ontology population.

1 Introduction

An ontology is a formal specification of a shared conceptualization [4]. Manual ontology construction is a time consuming process and it takes many months to construct an ontology from beginning. Application and objective of the ontology have to be defined and searching for the relevant concepts and metadata associated with each of the concept is a challenging task which usually takes much more time than anticipated. Due to the efforts and time consumption of constructing ontology, several approaches and applications have been developed. Some of them are automated and others are semi automated. Most of the automated ontology construction tools require technical expertise of computing and natural language processing making it difficult for people without computational background and it may take few months to couple of years to learn and master those tools. For example, as reported in a recent survey [6], learning and working with Protégé (which is not exactly an automated ontology construction tool though) is a time consuming task and it was found that six months experience was not sufficient to learn it. In addition, most of the automated tools work on a corpus to construct a hierarchical ontology. This means that the results could vary significantly on the basis of the content of the corpus. Some of these tools help to

P. Klinov and D. Mouromtsev (Eds.): KESW 2014, CCIS 468, pp. 237–241, 2014.

construct de-novo ontologies but cleaning the concepts which are not required and integrating associated metadata required in ontology are also cumbersome tasks. Some of the tools are presented as follows. ASIUM [3] (Acquisition of Semantic Knowledge Using Machine Learning Methods) acquires ontological knowledge from text given as an input. The system is based on conceptual and hierarchical clustering. Doddle II [10] is a system which can exploit the machine readable dictionary and text corpus to populate the domain specific ontology. KnowItAll [2] extracts facts from the web by using linguistic and statistics method and it is mainly designed for large scale information extraction. In addition, to the best of authors knowledge, there are few more programs such as MedSynDikate [5], OntoLearn [9], String-IE [8] and Text2Onto [1], but none of them provide an interface where a list of concepts can be given and associated metadata could be added in an automated way to construct an ontology.

Figure 1 shows the difference between a manual addition of a concept in an ontology and concept population via OntoFast. Adding a concept in an ontology roughly takes several minutes depending upon the size of metadata associated with it.

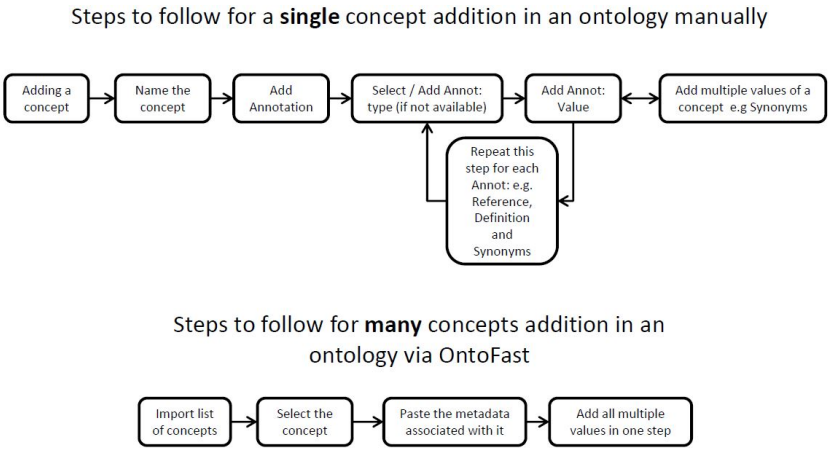

Fig. 1. The difference of steps between manual addition of a concept in an ontology and by using OntoFast to populate a new ontology

For example, if a user wants to add a concept with n synonyms then he/she has to repeat the same steps n times. Further, if he/she wants to add references then the same practice has to be done. The same goes with the definition and any other annotation and all this is only for one concept. If you are considering to construct an ontology with many concepts which are having dozens of synonyms (as in biological domain) then it would probably take weeks to months to do the simple task of populating an ontology which is not attractive for domain experts. With the help of OntoFast the same task can be performed within couple of minutes while constructing a new ontology. It works only for new ontology because we assumed that massive population of concepts only needed when an ontology is started to be constructed. The tool's output is an OWL/XML file in which the concepts are stored as Classes.

2 Use

The application provides an easy to use interface where an ontology can be constructed and populate very quickly (see Figure 1). Different options allow users to embed a definition, synonyms and references of the ontology via interface. Here we describe our approach to build an ontology very quickly.

Importing of concepts is very easy, since the list of concepts can be imported by clicking on the load new Txt button (Figure 2). All the concepts of a prospective ontology can be given in a form of list in text file (.txt). Fields in the text file should be separated by carriage return commands. The application reads each new line as a new concept and generates the list of concepts that is visualized in the declarations text box (Figure 2). The associated metadata can be then added by selecting a concept in the list. Selected concepts will be highlighted in purple and yellow. Just after importing the list of declarations, the application asks to choose the output xml file to be then used in Protégé or in any similar application. From this moment the user will not need to take care of manually saving the output xml file, since the application will execute automatically saving every time a different concept is selected, and on exit.

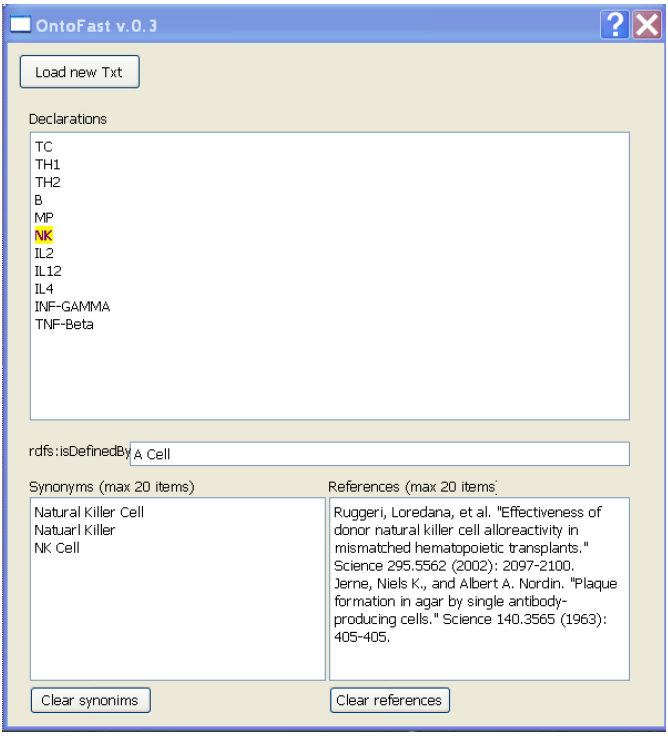

Fig. 2. OntoFast interface. 1: Load button for loading a list of declarations from a txt file. 2: Declarations list which shows concepts loaded by a txt file. A concept can be selected by clicking on it. The selected concept will be highlighted in purple and yellow. 3, 4 and 5: Fields for defining basic properties of the selected declaration. Synonyms and References fields allow to type more than one value.

Metadata can be easily associated with the imported concepts by selecting a concept and providing in the relative fields the associated details. The main attributes required for the ontology were definition, synonyms and references thus there are different text boxes given to incorporate the same. As the goal was to speed up the initial step in the generation of new ontologies, each of the boxes can accommodate copy/paste to quickly populate the ontology. In addition, more than one synonym and reference can be given in different lines. Finally, the hierarchy of the ontology can be arranged later on by user in Protégé, since such an operation can be already carried very quickly in it.

3 Results and Discussion

In this paper we presented an application which allows to construct ontologies quickly in order to speed-up the standard procedure of constructing ontologies and associating metadata. This actually takes time that goes from many months to couple of years and involves many people. For example, PLIO (Protein-Ligand Interaction Ontology) [http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3106195/] was constructed in 18 months, MSO (Multiple Sclerosis Ontology) was created in 1 year and GO (Gene Ontology) took many years and still being updated continuously. One of the main hurdles while constructing MSO was represented by the difficulty of introducing new concepts into the Protégé user interface. This task revealed to be time consuming and labor intensive. Since the ontology engineers are specialists in their domain and they may construct ontologies only for their specific needs, they are usually not experts in ontology construction work. This lack of practice often slows down the progression of the work and forces them to do repetitive task which can be automated easily. Our application solves this problem by providing an easy to use and convenient interface which facilitates quick ontology construction and save domain experts precious time. Since ontologies can be put with different hierarchies and different application scenarios which vary from person to person and from task to task, we were not actually interested in putting an hierarchical feature into it. In addition, the output of our program can be easily opened into standard ontology editors like Protégé and the hierarchy can be then changed by drag and drop to the users specific needs. Hierarchy relies on expert so domain expert can easily construct ontologies with the help of our tool instead of wasting time by manually populating the ontology. Our tool provides an easy to use interface to quickly populate and construct ontology instead of doing repetitive work of adding concepts one by one. More than one synonym and reference can be given in different lines making it more convenient for information retrieval systems to broaden the coverage of the ontology. Depending on the collection of metadata etc for biomedical domain, we showed that enrichment of ontology can be automated with the Knime workflow and UMLS [7]. In addition the same workflow can be used to query any public MySQL database with some minor changes. Ontologies are considered controlled vocabularies for knowledge representation. Our tool is a middle interface between dictionaries and ontologies and provides an easier way to transform one into another.

Acknowledgements. This work has been sponsored by Merck KGaA, a pharmaceutical and chemical company.

References

1. Cimiano, P., Völker, J.: Text2onto - a framework for ontology learning and data-driven change discovery. In: Montoyo, A., Muńoz, R., Métais, E. (eds.) NLDB 2005. LNCS, vol. 3513, pp. 227–238. Springer, Heidelberg (2005)
2. Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D.S., Yates, A.: Web-scale information extraction in knowitall (preliminary results). In: WWW 2004: Proceedings of the 13th International Conference on World Wide Web, pp. 100–110. ACM, New York (2004)
3. Faure, D., Nédellec, C.: Knowledge acquisition of predicate argument structures from technical texts using machine learning: The system ASIUM. In: Fensel, D., Studer, R. (eds.) EKAW 1999. LNCS (LNAI), vol. 1621, pp. 329–334. Springer, Heidelberg (1999)
4. Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge Acquisition 5(2), 199–220 (1993)
5. Hahn, U., Romacker, M., Schulz, S.: medsyndikate—a natural language system for the extraction of medical information from findings reports. International Journal of Medical Informatics 67(1-3), 63–74 (2002)
6. Khondoker, M.R., Mueller, P.: Comparing ontology development tools based on an online survey. In: Proceedings of the World Congress on Engineering (WCE 2010), London, U.K, vol. I (2010)
7. Rajput, A.M., Gurulingappa, H.: Semi-automatic approach for ontology enrichment using umls. Procedia Computer Science 23, 78–83 (2013)
8. Saric, J., Jensen, L.J., Ouzounova, R., Rojas, I., Bork, P.: Extraction of regulatory gene/protein networks from medline. Bioinformatics 22, 645–650 (2006)
9. Velardi, P., Navigli, R., Cucchiarelli, A., Neri, F.: Evaluation of OntoLearn, a methodology for automatic population of domain ontologies. In: Buitelaar, P., Cimiano, P., Magnini, B. (eds.) Ontology Learning from Text: Methods, Applications and Evaluation. IOS Press (2006)
10. Yamaguchi, T.: Acquiring conceptual relationships from domain-specific texts. In: Maedche, A., Staab, S., Nedellec, C., Hovy, E.H. (eds.) Workshop on Ontology Learning. CEUR Workshop Proceedings, vol. 38. CEUR-WS.org (2003)