Knowledge Author: Creating Domain Content for NLP Information Extraction

William Scuba[1], Melissa Tharp[1], Eugene Tseytlin, Yang Liu[2], Frank A. Drews[3], Wendy W. Chapman[1]

[1]Department of Biomedical Informatics, University of Utah, Salt Lake City, UT 84108, USA

[2]University of California, San Diego, San Diego, CA 92093, USA

[3]Department of Psychology, University of Utah, Salt Lake City, UT 84108, USA

wendy.chapman@utah.edu

Abstract

Knowledge Author (KA) is a web-based software tool that supports users in developing domain content necessary for Natural Language Processing (NLP) applications, as well as a tool to provide recommendations for additional domain content. To accurately extract information from text, NLP tools require a semantic schema describing variables and their lexical variants and associated modifiers. Knowledge Author uses a set of semantic categories based on the secondary use clinical element models (SUCEM) and the Common Type System (CTS) to allow the user to quickly create and modify domain related variables. Additional information such as synonyms are obtained using Unified Medical Language System (UMLS) Metathesaurus terminology lookup and presented to the user in an easy to use fashion. KA can be found at blulab.chpc.utah.edu/KA/ .

1 Introduction

A wide range of NLP pipelines and components specific to clinical text have been developed since the early 1990s, facilitating the analysis of emergency department notes, radiology reports, and other free-text clinical documents using lexical, syntactic, and semantic information (Friedman 2000). Within the clinical domain, NLP systems based on these pipelines and components have been implemented in a variety of contexts, such as pharmaco-vigilance, case finding and patient screening, summarization of narrative patient information, and quality measurement (Li, Chase et al. 2008, Pakhomov S, Bjornsen S et al. 2008, Wang, Hripcsak et al. 2009, Al-Haddad MA, Friedlin J et al. 2010, Aramaki, Miura et al. 2010, Chiang, Lin et al. 2010, Van Vleck TT and Elhadad N 2010).

However, with all of its usefulness, it remains time consuming to develop the domain content necessary for NLP tools to produce the desired results. Previously existing ontology creation tools such as Protégé (Protégé) are feature rich, but they can be time consuming to learn, and are designed to serve a broad range of uses (e.g., from modeling archaeological information to modeling geographic data). Knowledge Author (KA) focuses solely on the clinical NLP domain, which allows for a more tailored set of features. The goal of KA is to allow domain experts such as medical clinicians to efficiently develop domain-specific content that can be used in clinical NLP systems.

2 Design and System Description

The overall design goal of KA is to allow the user to quickly create a semantic schema, which is the target extraction template for an NLP system. KA allows the user to create domain related variables and assign a semantic category to the variable. In KA, a variable is defined as a lexical concept found in a standardized terminology along with any relevant modifiers associated with that concept. For instance, a user interested in extracting the variable "fever" from text can create the variable and associate it to the UMLS concept "C0015967 fever" and assign it a semantic category of Problem. In addition, a user could create more complex variables such as "white blood cell count >=12,000". To accomplish this in KA, the user would create the variable and associate the UMLS concept "C0015967 white blood cell count" to the variable and assign it a semantic category of "Lab/Test/Measurement". KA will then prompt the user to enrich the varia-

ble with additional information such as synonyms, semantic categories, and semantic and linguistic modifiers (Figure 1). Only information that is relevant to the particular semantic type of category the user is creating is presented, such as the lab/test/measurement value ">= 12000" in the example variable above.

KA makes variable creation more efficient by allowing a user to look up and choose a concept from the UMLS Metathesaurus and importing any relevant information such as synonyms, concept definition, and semantic category into the KA interface. The output of KA is a schema indicating what information the user wants to extract from clinical documents.

To test the validity of a newly created variable, KA allows the user to search a default corpus of de-identified medical records for phrases that would potentially be retrieved for the new variable. The search mechanism is a simple Apache Lucene reverse index query (Apache Lucene) that uses the variable name and synonyms to provide potentially useful diagnostics to the user through examples of what the NLP system would extract. For example, in looking through the example sentences, a user could see that there is a problem with the synonym of a variable and alert them to the need to edit the variable's synonym list.

2.1 Workflow

A typical user workflow to extract information from narrative reports would start with creating a new domain variable. KA currently supports two different types of variables – 'person' and 'non-person'. Upon creation, the 'person' type prompts the user to provide information such as birth date, death date, race, age and gender to facilitate creation of complex variables like "African American females above 65 years of age." The non-person type gives the user access to a range of information as described below.

2.2 UMLS Terminology Lookup

The next step in the workflow involves mapping the variable to a concept in the UMLS Metathesaurus (Figure 2). The user simply types in a concept and then clicks on the Terminology Lookup button. This will provide a popup window with a list of possible UMLS concepts to choose from. The list is generated using the UTS API 2.0 (UTS API 2.0). This mapping imports into the domain schema the synonyms, definition, and semantic type for the variable. A manual review process allows the user to remove or modify any of the imported data.

2.3 Corpus Search

Once a variable is created, KA allows the user to perform a corpus search to test the validity of the variable name and synonyms. The MTSamples (MTSamples) corpus of de-identified medical records is available for the user to search their variables against. The data use agreement for this corpus allows it to be used in KA.

The corpus search tool returns a list of sentences that contain the variable name. Statistics about how many sentences were returned relative to the corpus size are also displayed. This helps the user gauge the usefulness of the name and synonyms and potentially results in elimination of some errors before deploying the

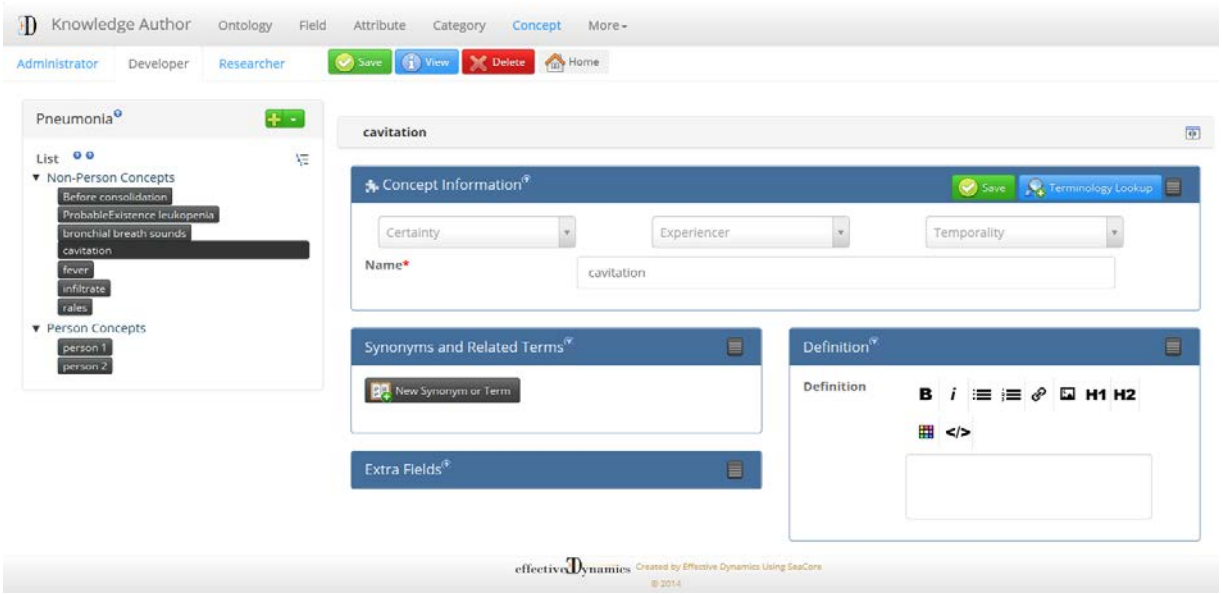

Figure 1: Concept creation interface

domain schema in an NLP tool. For example, one of the synonyms for 'malignant melanoma' in the UMLS is 'skin.' Including that synonym would create a large number of false positives that could be picked up using the corpus search tool.

2.4 Linguistic Modifiers

Category	Modifiers
Certainty	Definite Existence, Definite Negated Existence, Probable Existence, Probable Negated Existence
Experiencer	Patient, Family Member, Donor Family Member, Donor Other Member, Other Member
Temporality	Before, Before-Overlap, Overlap, After

Table 1: Linguistic modifiers available to the user

Linguistic modifiers allow the user to narrow down their variable to reduce irrelevant matches by specifying linguistic information found in the surrounding text where the variable is found. KA allows the user to specify the temporality (whether the concept occurs in the past, present, or future), certainty (whether the concept is asserted, negated, or hedged), and experiencer (whether the patient or someone else experiences the concept). For example, "colon cancer" could be modified to "possible family history of colon cancer". Table 1 lists the possible linguistic modifiers the user can choose from.

2.5 Semantic Modifiers

Semantic modifiers are similar to linguistic modifiers with the exception that specific modifiers are associated with specific semantic categories – unlike linguistic modifiers that are associated with all non-person variables. The semantic category of the variable determines, which semantic modifiers can be assigned to the variable. There are currently eight possible semantic categories to choose from - Anatomical Site, Allergy/Intolerance, Encounter, Lab Test Measurement, Medication, Problem, Procedure Intervention, Social Risk Factor, and Vital Sign. Each category contains a number of possible semantic modifiers which were based on the secondary use clinical element models (Tao et al. 2013) and the Common Type System (Wu, et al. 2013), which are two commonly used models employed to represent information extracted by NLP systems in the biomedical NLP community. Each of the semantic modifiers has, in turn, a number of possible values associated with it. For example, for the semantic category 'Medication', the user would be able to choose from semantic attributes such as 'dosage' or 'delivery route.' The 'delivery route' semantic attribute has a number of possible values such as 'oral' or 'intravenous.' Semantic attributes support creation of variables such as "Ibuprofen 80 mg orally every four hours."

2.6 Input/Output File Format

KA saves the user's work to an internal database that is available to the user upon login. The data

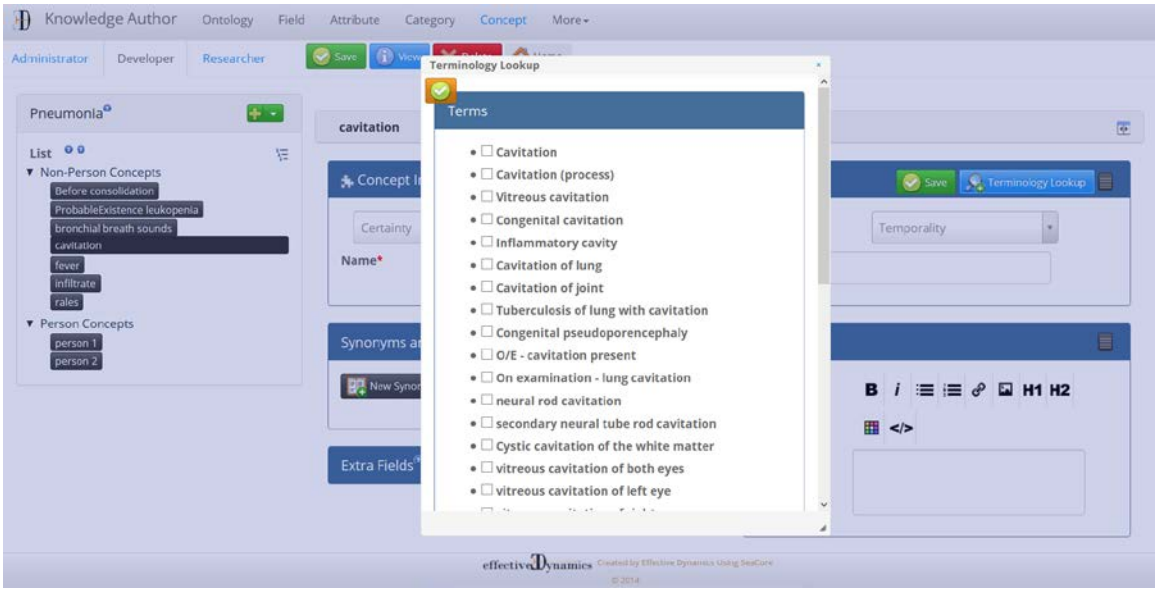

Figure 2: UMLS terminology lookup interface

Concept Name	Created in KA	Reason
infiltrate	yes	
consolidation	yes	
cavitation	yes	
fever	yes	
temperature > 100.4 deg F	partial	NT
leukopenia	yes	
white blood cell count < 4000 WBC/mm3	partial	NT
leukocytosis	yes	
white blood cell count >= 12000 WBC/mm3	partial	NT
altered mental status in adults >= 70 years old	yes	
new or change in purulent sputum	yes	
increased respiratory secretions	yes	
increased suctioning requirements	yes	
new or worsening cough	partial	LON
new or worsening dyspnea	partial	LON
new or worsening tachpena	partial	LON
respiratory rate > 25 bpm	partial	NT
rales	yes	
bronchial breath sounds	yes	
worsening gas exchange	yes	
increased oxygen therapy	yes	
increased respiratory suctioning	yes	
increased ventilator care	yes	
Oxygen saturation (PaO2/FiO2 <= 240)	partial	NT

Table 2: List of concepts used in the pneumonia use case. Whether or not the concept was able to be fully created using KA, and the reason if not is also listed.
LON - Logical operator between modifiers not supported in KA at the time the use case was completed.
NT - Numeric Threshold not supported in KA at the time the use case was completed.

can also be exported from the database to an OWL (OWL) file.

2.7 Software Architecture

KA is a web-based platform written in Java 7 on top of a MySQL database. The SeaCore (SeaCore) framework is used to facilitate the web development. The UMLS terminology is accessed through use of the Java based UTS API 2.0 which queries the UMLS metathesaurus service. The corpus search feature uses Apache Solr 4.7 which is a web service wrapper to the Apache Lucene Java search library (Apache Solr).

3 Use Case

To test the functionality of KA, we created a pneumonia use case file using KA and compared it with an OWL file created manually using Protégé 4.1 of the same domain. The same knowledge engineer created both files and only timed how long it took to create the variables and related lexical information (i.e. time it took to gather and to learn the use case and the time it took to build the proper structures in Protégé to hold the information was not included in the time comparison). Using KA, the variables could be created in a fraction of the time it took to manually create the same variables in the manual OWL file and did not require expertise in OWL. This major time difference was mainly due to the fact that within KA, most of the lexical information (i.e. synonyms, definition, semantic category) was automatically imported when a variable is mapped to a UMLS concept, which had to be manually added to the Protégé file. It should be noted that KA is not designed to be a replacement for Protégé as it has a different feature set, but rather to speed up and simplify the creation of domain content.

Both versions were able to create variables in the pneumonia schema (see Table 2). In KA, however, 33% of the concepts could not be represented fully because currently, KA lacks the ability to add numeric thresholds and the ability to use logical operators to combine values of modifiers (i.e., “new or worsening cough”). In its current version, a user would have to create two separate variables that share the same concept but have different modifiers to capture this information (i.e. “new cough” and “worsening cough”). However, the user would have to duplicate the Terminology lookup, synonyms and definition for each variable that shares the same underlying concept.

4 Conclusion and Future Development

We have presented a web-based tool for building a semantic schema of domain content that could be used in an NLP application. KA leverages three existing knowledge resources – the

SUCEMs, CTS, and the UMLS – to provide the user with relevant information for creation of domain-specific variables, which allows for rapid semantic schema creation by a user without formal training in ontology development or NLP. Future work includes adding constructs that will allow users to link variables together using relationships (i.e. "ibuprofen *treats* pain") and logical operators and usability testing with naïve users.

KA is the first part of a pipeline that will allow the user to create an NLP schema, annotate documents, process documents using various NLP tools, and analyze the results. We envision an end-to-end system that allows the user to rapidly build custom clinical text queries. The full pipeline is currently under development.

Acknowledgements

We would like to acknowledge Effective Dynamics for their excellent work programming the system. This work was supported by National Library of Medicine grant LM010964.

Reference

Al-Haddad MA, et al. (2010). "Natural Language Processing for the Development of a Clinical Registry: A Validation Study in Intraductal Papillary Mucinous Neoplasms." HPB **12**(10): 688-695.

Apache Solr. http://lucene.apache.org/solr/ .

Apache Lucene. http://lucene.apache.org/ .

Aramaki, E., et al. (2010). "Extraction of Adverse Drug Effects from Clinical Records." Studies in health technology and informatics **160**(1): 739-743.

Chiang, J.-H. H., et al. (2010). "Automated Evaluation of Electronic Discharge Notes to Assess Quality of Care for Cardiovascular Diseases Using Medical Language Extraction and Encoding System (MedLEE)." Journal of the American Medical Informatics Association **17**(3): 245-252.

Friedman, C. (2000). "A broad-coverage natural language processing system." Proc AMIA Symp: 270-274.

MTSample. http://www.mtsamples.com/ .

OWL. http://www.w3.org/TR/owl-features/ .

Pakhomov S, et al. (2008). "Quality Performance Measurement Using the Text of Electronic Medical Records." Medical Decision Making **28**(4): 462-470.

Protoge. http://protege.stanford.edu/ .

Rindflesch, T. C. and M. Fiszman (2003). "The interaction of domain knowledge and linguistic structure in natural language processing: interpreting hypernymic propositions in biomedical text." J Biomed Inform **36**(6): 462-477.

SeaCore. https://xp-dev.com/summary/203304 .

Tao, C, at al. (2013). "A semantic-web oriented representation of the clinical element model for secondary use of electronic health records data." Journal of the American Medical Informatics Association 1;20(3):554-62.

United Medical Language System (UMLS). http://www.nlm.nih.gov/research/umls/ .

UTS API 2.0. https://uts.nlm.nih.gov//home.html#apidocumentation

Van Vleck TT and Elhadad N (2010). Corpus-Based Problem Selection for EHR Note Summarization. AMIA Annu Symp Proc. Washington, DC**:** 817-821.

Wang, X., et al. (2009). "Active Computerized Pharmacovigilance using Natural Language Processing, Statistics, and Electronic Health Records: a Feasibility Study." Journal of the American Medical Informatics Association.

Wu, S., et al. (2013). "A common type system for clinical natural language processing." Journal of Biomedical Semantics 4:1