GazOntology: A Tool for Building GATE Gazetteer Lists from Ontologies

Daniel R. Schlegel, Rose Fontana, and Adrian Naaktgeboren

Department of Computer Science, SUNY Oswego, Oswego NY, USA

Abstract

The integration of ontologies with natural language processing tools is necessary for the use of said ontologies in named entity recognition tasks. We present a tool to convert (parts of) ontologies, including synonyms, to a convenient list form for use with the GATE text processing system.

Introduction

One of the impediments to using ontologies in natural language processing (NLP) tasks is the lack of tools for automatically annotating text with ontological terms. The most frequently used tool for this task is MetaMap which uses the UMLS Metathesaurus to combine several vocabularies, terminologies, and ontologies. Unfortunately, it does not contain a large number of ontologies, and the methodology for merging terms from various sources can lead to mismatched terms. Some groups have made use of other "terminology servers" to handle matching free-text text with terms in some terminology, but these are often custom designed and limited in scope.

In NLP parlance, the task of annotating text with ontological terms would be a named entity recognition (NER) task. Existing NLP tools have a variety of mechanisms for NER, supporting matching strings of text against lists, databases, rules, and so on. Less frequently are ontologies supported.

Until somewhat recently, inclusion of large sets of synonyms in ontologies was somewhat rare, but are exactly what is required for using ontologies as an NER resource. As noted by Spasic, most of the labels we use for ontological terms are not the kind of linguistic units we use in language, but rather "... they have more in common with documentation thesaurus descriptors, facet labels or index terms from a controlled vocabulary than with terminological terms" (Spasic et al. 2005).

The General Architecture for Text Engineering (GATE) (Cunningham et al. 2002) is a tool for designing pipelines of NLP tools and running them on text documents, often for the purpose of information extraction. GATE supports a significant number of plugins for many NLP tasks and integrates with pipelines such as Stanford CoreNLP, OpenNLP, and UIMA. Importantly for us, GATE supports a suite of NER tools, including the (poorly named) gazetteer which performs string matching against lists. Of course, list-based NER is inherently limited and must be combined with more sophisticated methods, which GATE also supports.

We present a tool which allows automatic generation of GATE gazetteer lists from ontology hierarchies.[1]

GATE Named Entity Recognition

GATE processes text using a pipeline of processing resources (PRs). These PRs produce annotations, attaching feature-value pairs to spans of text. Among other things, these pairs can include simple properties such as the length of a span of text, linguistic properties such as the part of speech of a word, and the semantic category denoted by a span of text derived from NERs.

GATE includes mechanisms for both list-based and rule-based NER, often used together. Lists may contain complete named entities or words (keys) which given context can indicate that a named entity begins with or ends with the key (*e.g.*, "Hospital" in a hospital name or "Jr." in a person's name). Rule-Based NER allows identification of named entities through regular expressions over annotations using the Java Annotation Patterns Engine (JAPE). These rules allow for recognition of complete entities for which keys were noted in the list-based NER. Rules may be used for simple post-coordination tasks based on linguistic patterns as well as the already identified named entities. They also provide an opportunity for a first pass at disambiguation and the removal of over-matches given the context available in word orderings.

The tool we have developed may be used to generate gazetteer lists for use with the list-based NER. A GATE gazetteer list resides in a file in which every line begins with a textual string which is to be identified in the free text document being annotated. It then contains a set of zero or more feature-value pairs, where a feature provides a name for some property of the matched string, and the value provides a value for that property. The format of these files is provided below in EBNF format.

```
gazFile -> {row}
row -> string {feature-value}
feature-value -> ":" feature "=" value
```

The feature-value pairs become part of annotations on the textual span which is matched in the document being annotated. Entire GATE gazetteer lists may each also (optionally) be given a major type and minor type, which are included in

[1] The tool discussed in this paper is available at: https://github.com/oswegonlu/GazOntology.

the text annotations after a match. Including the major and minor types in the text annotations allows for higher level categorization of a collection of terms. Rule-based NER can make use these annotations in performing matching. Writing generalized rules makes particular use of the major/minor types.

GazOntology

GazOntology is a tool for automatically generating gazetteer lists from ontologies. It is capable of extracting the term labels and synonyms for all or a subset of terms in an ontology and using them to generate gazetteer lists. The tool aims to provide flexibility, allowing the user to select whether they wish to include imported ontologies, whether they wish to include only terms with a certain ontology prefix (e.g., GO, or PRO) or all terms, and whether to include the entire ontology or just those terms subsumed by a provided class expression. The current version of the tool supports synonyms defined in the ways prescribed by the Relation Ontology (RO). This allows for three relations connecting terms to their synonyms: `hasExactSynonym` for exact synonyms, and `hasRelatedSynonym` and `hasBroadSynonym` for less precise synonymy. The user of our tool may select which of these relations to include.

The generated lists include a feature-value pair which links the textual string to the IRI of the term, as in the following small excerpt from the Units of Measurement Ontology (UO):

```
kilovolt:IRI=UO_0000248
kV:IRI=UO_0000248
kg:IRI=UO_0000009
kilogram:IRI=UO_0000009
```

The tool also optionally outputs a line of text to be added to the gazetteer's index of lists which contains the name of the output file, the major type, and the minor type. The major type is by default set to the top term in a hierarchy if a class expression is provided. There is currently no default minor type.

This tool supports NLP pipeline development in that it is easy to add synonyms which link to IRIs without modifying the ontologies themselves. Maintaining custom versions of ontologies can become burdensome as the source ontology evolves. The workflow we have adopted involves placing these additional synonyms in a separate gazetteer list so that updated versions of the ontology can be used to regenerate the synonym list without disrupting our manually added synonyms.

Discussion and Future Work

Attempting to create complete sets of synonyms for complex terms is a battle that can never be completely won. Ontologically, plural terms are different from their singular counterparts, and yet groups of items of a kind may not be in an ontology even though they often are mentioned in text. English allows different word orderings to denote the same entity, and sometimes these can be spatially distributed throughout a sentence. Other languages which allow free word ordering have this problem compounded further as term names become more complex. It is difficult to find the balance between lists of synonyms and rules which might build more complex terms from simpler terms. It is the opinion of the authors that lists such as these are useful mostly in identifying the atomic entities or concepts, and that they should be composed in some way to arrive at the more complex terms (likely involving a mapping between relations in the ontology and semantic relations from the language). This greatly complicates the NER task and perhaps illustrates that NLP cannot truly be a pipeline, but is more a web of interacting pieces.

All of this said, it is likely useful to generate some types of derivational variants for terms and synonyms, as is done by MetaMap (Aronson 2001). Future work should focus on this area and allow users to select which types of variants they are interested in. Word order variants are likely outside the scope of this work.

GATE includes a suite of ontology tools which allow for viewing, editing and tagging text with ontology terms. Unfortunately, the tagging tool only works on the class names (not any properties such as synonyms), and does not seem to support getting the term labels, instead using IRIs often used as class names in large ontologies. The advantage to this tighter integration is that the JAPE-related PRs are ontology aware, allowing for hierarchical compatibility checks between text and rules. Unfortunately its limitations are such that it is not usable for most tasks. While we see the benefit in modifying these existing tools to support a wider range of ontologies, we also see an advantage in creating something which abstracts away from the ontology so that ontology updates are less disruptive. Future work should aim to balance these concerns.

We are still experimenting with use cases and ways this tool might be generalized for use by the larger community. Our use case involves downstream processing which will make use of the IRI features on annotated text and will then do further reasoning using this connection into the underlying ontologies, but others may have different considerations. The tool has been published under an open source license, and we welcome feedback and contributions.

Acknowledgements

This work was supported by the National Library Of Medicine of the National Institutes of Health under Award Number R15LM013030. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

References

Aronson, A. R. 2001. Metamap variant generation.

Cunningham, H.; Maynard, D.; Bontcheva, K.; and Tablan, V. 2002. GATE: A framework and graphical development environment for robust NLP tools and applications. In *ACL02*.

Spasic, I.; Ananiadou, S.; McNaught, J.; and Kumar, A. 2005. Text mining and ontologies in biomedicine: making sense of raw text. *Briefings in bioinformatics* 6(3):239–251.