Woolery: Toward Semantic Annotation of Structured Documents with FrameNet

Chloe Eggleston
College of Information and Computer Sciences
University of Massachusetts Amherst
Amherst, USA
ceggleston@umass.edu

Jeremy Abramson
Information Sciences Institute
University of Southern California
Los Angeles, USA
abramson@isi.edu

***Abstract*—In this paper, we introduce *Woolery*, a preliminary design and implementation of an intelligent annotator interface for structured documents (i.e. JSON). This annotator infers a mapping between the semantics of the keys in the document and an appropriate FrameNet structure, which can then be exported for downstream natural language processing tasks.**

***Index Terms*—FrameNet, NLP, annotation, lexical databases, JSON, ontology alignment, computational semantics**

I. Introduction

FrameNet is a lexical database used in the NLP community to label semantic frames: recurring linguistic structures that organize concepts and events into a defined schema [1]. The semantics of FrameNet, thus, are useful for many NLP tasks, such as Semantic Role Labelling [2], machine translation [3], and rule-based Natural Language Generation systems [4]. However, FrameNet relies on a limited number of annotations, which are only defined on unstructured text, and it is not fully mapped to related lexical databases such as WordNet [5].

A semantic frame in FrameNet is composed of *Frame Elements* (FE), which are contextual arguments that only allow for certain classes of words to be inserted into them. FrameNet also provides *lexical units* (LU): specific words and their parts of speech that have been labeled as potential arguments for a given frame, along with the given FE(s) that they align to and the sources they were annotated from. Woolery provides a semi-automated platform for a user — who may not be familiar with linguistic databases — to align the keys of a JSON document to the LUs they represent.

II. Approaches and Preliminary Results

To facilitate annotation, Woolery recurses down the tree structure of a JSON document, building a list of keys. It then tokenizes these keys into their constituent words and stems each of those words. For example, the key-value pair: `{"dateCreated": 1577880000}` would be tokenized into `date` and `created` and stemmed into `date` and `creat`. These stemmed words are then fed into a regex search of all LUs, providing a list of candidates that best match the semantic meaning of the key. For the preceding example, the tokenized and stemmed terms have a number of LU candidates: *date.n*, *date.v*, *consolidate.n* (etc.) for `date`, and *create.v*, *creation.n* for `creat`. The user examines the list — along with additional linguistic and contextual information from FrameNet about the specific frame and LU choice — and selects the appropriate annotation. The mapping is added to an output file for downstream consumption by a NLU/NLG tool.

Source	Direct	Hyp$_1$	Hyp$_3$	Hyp$_n$
Semlink 2 [8]	1872	10478	10836	10879
MapNet [6]	3663	14004	28481	36694
WordFrameNet [7]	6471	24017	49514	70009
Synset Coverage	$\frac{8046}{117659}$	$\frac{27570}{117659}$	$\frac{52290}{117659}$	$\frac{71937}{117659}$

Direct LU Mapping for a Synset
Hyp$_1$ LU Mapping within a Synset's Hypernyms
Hyp$_3$ LU Mapping within three Hypernym Expansions
Hyp$_n$ LU Mapping within all possible Hypernym expansions

TABLE I

While this search is often enough to surface the intended semantics of the JSON key, FrameNet may not have a match for a given term. To mitigate this, Woolery can access WordNet synonyms and hypernyms (or "umbrella terms") and map these more broad terms back to the appropriate FrameNet LU.

Woolery uses WordNet's Morphy interface and its synset search to produce a list of candidates. The user is provided with a list of synset categories called lexnames, such as *noun.event*, *noun.location*, *verb.communication*, or *verb.creation*. After selecting a category, the user is presented with the related synsets, as well as other related linguistic information from WordNet, and can make an appropriate selection. However, this selection is in the context of WordNet, not FrameNet, and must be mapped back to the appropriate LU.

As of now, there is no complete mapping from of WordNet to FrameNet, but it is a well-studied task in the lexical databases community [6], [7]. We used the following resources to build as complete of a mapping of WordNet 3.0 to FrameNet 1.7 as possible (see Table I). Note that these mappings are not strictly one-to-one; in the case of a synset with multiple lexical units, the user will need to select one themselves.

Semlink 2 provides mappings from another linguistic

This work was performed in a Research Experiences for Undergraduates program funded by National Science Foundation grant #1659886.

database called VerbNet [9] version 3.3, to FrameNet version 1.7. VerbNet includes references to lemmas in WordNet for each of its verb examples which were used to obtain synsets. Semlink provided the names of the frames, but not the LU(s), so for each Frame, we checked if the verb's exact name was a LU, and if not, we iterated through that verb's lemmas to get the LU Mappings.

Both MapNet and WordFrameNet provide mappings from WordNet 1.6 to FrameNet 1.3. The WordNet 1.6 synsets were mapped to WordNet 3.0 using prewritten mappings [10]. Synsets in WordNet 1.6 that map to multiple synsets in WordNet 3.0 were included as separate entries. For frames in FrameNet 1.3 that were not in FrameNet 1.7, we tested if there was a v1.7 frame with all the same lexical units as the mentioned v1.3 frame. This, along with human verification, allowed us to map some frames that were renamed between FrameNet versions. Entries where no frame match was found were dropped from the list. Finally, we used the word and part of speech to obtain the LU mappings within each given frame.

Lastly, in addition to the above semi-automated approaches, Woolery also provides an interface for the user to search through *all* frames in FrameNet, and select a lexical unit of a given frame that is the most similar to a given key.

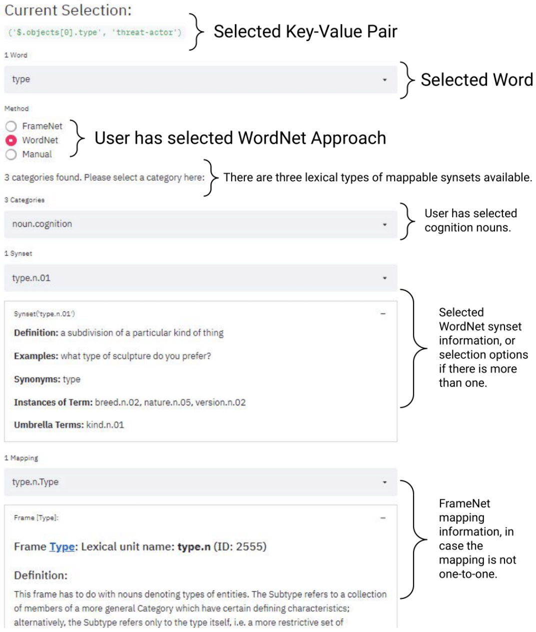
Currently, Woolery is implemented as a graphical user interface in Python (see Figure 1). The user is presented with a prompt to load a JSON file and with a list of extracted keys (not pictured). For each key, the user is given a choice of methods to obtain the appropriate LU mapping (as given above). As each mapping is chosen, the location of the key (via its *JSONpath*) and the selected LU's FrameNet ID are saved to disk.

III. Conclusions and Future Work

While preliminary results are promising with the proposed approach, there are a number of potential avenues for further refinement. First, Woolery could appeal to non-linguistic (i.e. semantic) ontologies such as ConceptNet or WikiData. These ontologies may provide additional or improved mappings versus the current approach via navigating their semantic class hierarchies, much in the same way as shown with WordNet synsets above. In addition, the system could consider multiple or nested keys as a way of increasing the semantic content of an annotation item. Lastly, with a corpus of sufficient breadth, graphical methods could be used on the *values* of a document, providing semantics not otherwise available by examining the key alone.

Determining the semantics of key/value pairs in structured documents can be a time consuming task. FrameNet provides a framework for classifying semantics into discrete and quantifiable structures. However, mapping documents to it requires either automated approaches — which are often inaccurate or need large corpora of unstructured texts — or a user with intimate knowledge of FrameNet internals. Intelligent annotation with Woolery helps mitigate the deficiencies of fully automated annotation approaches, without requiring linguistic expertise on behalf of the user.

Fig. 1. Woolery's WordNet Annotator Interface



Current Selection:

('$.objects[0].type', 'threat-actor')

Selected Key-Value Pair

1 Word

type

Selected Word

Method

○ FrameNet
● WordNet
○ Manual

User has selected WordNet Approach

3 categories found. Please select a category here:

There are three lexical types of mappable synsets available.

3 Categories

noun.cognition

User has selected cognition nouns.

1 Synset

type.n.01

Synset('type.n.01')

Definition: a subdivision of a particular kind of thing

Examples: what type of sculpture do you prefer?

Synonyms: type

Instances of Term: breed.n.02, nature.n.05, version.n.02

Umbrella Terms: kind.n.01

Selected WordNet synset information, or selection options if there is more than one.

1 Mapping

type.n.Type

Frame [Type]:

Frame Type: Lexical unit name: **type.n** (ID: 2555)

Definition:

This frame has to do with nouns denoting types of entities. The Subtype refers to a collection of members of a more general Category which have certain defining characteristics; alternatively, the Subtype refers only to the type itself, i.e. a more restrictive set of

FrameNet mapping information, in case the mapping is not one-to-one.

References

[1] C. F. Baker, C. J. Fillmore, and J. B. Lowe, "The Berkeley FrameNet project," in *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*. Montreal, Quebec, Canada: Association for Computational Linguistics, Aug. 1998, pp. 86–90. [Online]. Available: https://aclanthology.org/P98-1013

[2] D. Das, D. Chen, A. F. Martins, N. Schneider, and N. A. Smith, "Frame-semantic parsing," *Computational linguistics*, vol. 40, no. 1, pp. 9–56, 2014.

[3] S. Peron-Corrêa, A. Diniz, M. Lara, E. Matos, and T. Torrent, "Framenet-based automatic suggestion of translation equivalents," in *Computational Processing of the Portuguese Language*, J. Silva, R. Ribeiro, P. Quaresma, A. Adami, and A. Branco, Eds. Cham: Springer International Publishing, 2016, pp. 347–352.

[4] D. Dannélls and N. Gruzitis, "Controlled natural language generation from a multilingual framenet-based grammar," in *Controlled Natural Language*, B. Davis, K. Kaljurand, and T. Kuhn, Eds. Cham: Springer International Publishing, 2014, pp. 155–166.

[5] C. Fellbaum, *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

[6] S. Tonelli and D. Pighin, "New features for FrameNet - WordNet mapping," in *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*. Boulder, Colorado: Association for Computational Linguistics, Jun. 2009, pp. 219–227. [Online]. Available: https://aclanthology.org/W09-1127

[7] E. Laparra, G. Rigau, and M. Cuadros, "Exploring the integration of wordnet and framenet," in *Proceedings of the 5th Global WordNet Conference (GWC 2010), Mumbai, India*, 2010.

[8] K. Stowe *et al.*, "Semlink 2.0: Chasing lexical resources," in *Proceedings of the 14th International Conference on Computational Semantics (IWCS)*. Groningen, The Netherlands (online): Association for Computational Linguistics, June 2021, pp. 222–227. [Online]. Available: https://www.aclweb.org/anthology/2021.iwcs-1.21

[9] K. K. Schuler, "Verbnet: A broad-coverage, comprehensive verb lexicon," Ph.D. dissertation, University of Pennsylvania, 2006. [Online]. Available: http://verbs.colorado.edu/ kipper/Papers/dissertation.pdf

[10] J. Daudé, L. Padró, and G. Rigau, "Mapping WordNets using structural information," in *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*. Hong Kong: Association for Computational Linguistics, Oct. 2000, pp. 504–511. [Online]. Available: https://aclanthology.org/P00-1064