Milestone: Augmenting Interactive Semantic Parsing with Word Embeddings

Brendon Boldt

Nov/13/2017

Abstract

Semantic parsing provides a powerful way to demonstrate an understanding of a natural language utterance since it constructs easily executable formal representations from a utterance. One issue with semantic parsing, though, is that it treats words merely as tokens and has no mechanism for inferring the semantics of an unknown word. Using vector representations of words in semantic parsing provides an effective solution to this problem since it allows the semantic parser to infer the meaning of new words by comparing their similarity to known words. Furthermore, vector word embeddings can be learned at a large scale since it is an unsupervised algorithm.

1 Introduction

One of the major subfields of natural language processing is natural language *understanding* which carries the connotation that the system processing the language is able to find meaning in an utterance rather than simply searching for keyword occurences or otherwise. One such method of demonstrating an "understanding" of a natural language is by having the system in question produce a concrete action or piece of information which has semantic matching that of the utterance.

Semantic parsing, on a general level, is an effective way of demonstrating an understanding of natural language because it maps utterances to formal representations (e.g., lambda-DCS, SQL) based on a formal grammar. Semantic parsing has been applied to tasks such as answering questions, interpreting email-related commands, and building virtual block structures. [1] [2] [4].

Like most formal grammars, semantic parsing grammars usually treat each word as either a unique or lemmatized[1] token. In this model, each word has no meaning until it is explicitly assigned one by the semantics of the grammar.

This inflexibility is important in the contexts of applications like programming languages where exactness is expected and required, but this trait is not so helpful when parsing natural language. For example, a semantic parser might interpret "select red box" while not understanding "choose red box" which, for the large part, mean the same thing. Understanding natural language requires flexibility in that words with similar meaning should have *some* degree of interchangeability as opposed to none which is offered by the typical token-based approach.

It would be better, then, to represent words in a way that would group words with similar meanings, and word embeddings, that is, representing words as vectors gives us a very efficient way to do this. One of the primary advantages of augmenting semantic parsing with word embeddings as opposed to dictionary synonyms is that learning the embeddings is a unsupervised task which allows it to be done at scale much more easily than hand-picking sets of words which are semantically similar enough to a given token.

[1]Lemmatized tokens account for word inflections; for example, "walks", "walked", and "walking" would all match "walk" when lemmatized.

2 Methodology

For the purposes of this paper, we ill be using the following grammar which is a simplified and slightly altered version of the of the grammar used in Voxelurn [4].

$\langle Root \rangle$ ::= $\langle Action \rangle$

$\langle Action \rangle$ ::= $\langle ActionVerb \rangle$ $\langle Object \rangle$

$\langle ActionVerb \rangle$ ::= '`add`' | '`remove`' | '`select`'

$\langle Object \rangle$ ::= $\langle ObjectItem \rangle$ | $\langle Color \rangle$ $\langle ObjectItem \rangle$

$\langle ObjectItem \rangle$ ::= '`ball`' | '`box`'

$\langle Color \rangle$ ::= '`light`' | '`dark`'

Using a simple grammar such as the one above allows us to isolate a few grammar rules to test the feasibility of using word embeddings to augment semantic parsing.

Normally, parsing uses a standard bottom-up approach where the terminal tokens are matched with the corresponding tokens in the utterance; if one of the tokens in the utterance does not match, the parse will fail. In the augmented semantic parser, an unknown token is treated as a wildcard so that the parse succeeds. The unknown token is then compared with the tokens in the grammar using the word embeddings to determine which grammatical token to use.

We are using the GloVe word vector model as a source for word embeddings [3]. To measure the degree of similarity between two words, we are using the consine similarity of two corresponding vectors .

3 Experiment

The experiement will consist of taking valid sentences in the grammar and replacing individual words with synonyms and determing whether the augmented semantic parser correctly infers the correct parse given the unknow token. For example, take the utterance "build box"; here we have an $\langle ActionVerb \rangle$ replaced with the word "build". The parser would make a correct inference if it maps "build" to the correct grammatical $\langle ActionVerb \rangle$ which would be "add" in this case (and not "remove" or "select") since "build" and "add" are loose synonyms. For this experiement, WordNet will be used to generate the synonyms.

Further experimentation would involve using human subjects (e.g., using Amazon Mechanical Turk) to directly test whether the robustness added by the word embeddings is effective or not. The effectiveness would be determined by the ability of the semantic parser to address the natural variations in language as used by humans rather than just the strict synonym replacement demonstrated in this paper.

4 Discussion

5 Conclusion

6 Progress

I am currently in the process of modifying the SEMPRE semantic parser to parse an unknown tokens as a wildcard so that the candidate tokens can be compared against the unkown token using the word embedding. I have currently written the code that does the word embedding comparison based on the GloVe pre-trained word embeddings. I plan to use WordNet to generate the majority of synonyms that will be used to test the word embedding semantic parser; specifically, the Java WordNet interface provided here: `https://rednoise.org/rita/reference/RiWordNet.php` will be used.

References

[1] Amos Azaria, Jayant Krishnamurthy, and Tom Mitchell. Instructable intelligent personal agent, 2016.

[2] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1533–1544. ACL, 2013.

[3] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[4] Sida I. Wang, Samuel Ginn, Percy Liang, and Christopher D. Manning. Naturalizing a programming language via interactive learning. *CoRR*, abs/1704.06956, 2017.