EN.601.769 Assignment 1: Semantic Role Labeling

Instructors: Benjamin Van Durme, Kyle Rawlins
TA: Seth Ebner

Due: March 5, 2021

1 Introduction

We've seen that semantic roles are useful for reasoning about and describing events. We've also seen that the inventory of categorical semantic roles can easily proliferate to handle a number of edge cases, which motivated the theory of fine-grained proto-role properties [1].

In this assignment, you will create datasets and build models for predicting semantic roles. You will convert annotations for semantic proto-roles into annotations for coarse semantic roles, then train a set of binary classifiers to predict the semantic roles of arguments with respect to predicates. You will also explore aspects of the data and models and propose extensions to your approach.

2 Decomp Toolkit

The Decomp Toolkit [3] is available at: `https://github.com/decompositional-semantics-initiative/decomp`. It contains the Universal Decompositional Semantics datasets as well as tools for querying the data and the underlying graph structures. Instructions for installing the data and toolkit, as well as illustrative examples, can be found at: `https://decomp.readthedocs.io/en/latest/index.html`.

3 Assignment

This assignment is composed of **Do** and **Discuss** sections. You will submit a report with responses to the items in the **Discuss** sections. The **Do** sections do not require responses unless otherwise stated.

Upload your code to a GitHub repository and include a link in your report. If it is a private repository, give the instructors view access. The instructors' GitHub usernames are: `vandurme`, `rawlins`, and `sethebner`.

3.1 Data collection

Do:

1. Choose 3 semantic roles to model, in addition to `AGENT` and `PATIENT`. They can be anything you like but should include a reference to the linguistics literature.
2. We will define the `AGENT` role via UDS properties as: `AGENT` := ((`volition` >0) $\vee$ (`instigation` >0)) $\wedge$ (`existed-before` >0). You must implement `AGENT` under that definition. Define `PATIENT` and the remaining three roles using UDS properties.

3. Write a `SPARQL` query[1] to execute against the UDS data for each of the 5 semantic roles based on your definitions. Execute the queries on the first 1000 items in the train split.[2] Place your query strings in a file in your repository.

4. Apply your definitions over the train, development, and test splits of the UDS v1.0 dataset to obtain a dataset for each of the semantic roles (5 datasets in total). Each item in your datasets should contain (at minimum): the tokenized sentence, an indicator of the predicate head token, an indicator of the argument head token, and the label.

Discuss:

1. Why did you choose the semantic roles you did?

2. Motivate the definitions you developed for each semantic role.

3. How often does each semantic role occur in your training sets? Why might some roles occur much more frequently or rarely than other roles?

3.2 Modeling

We recommend using AllenNLP (`https://allennlp.org`) to implement your models. For help getting set up with AllenNLP step-by-step, we suggest reading this series of blog posts: `https://jbarrow.ai/#allennlp`. The model for semantic proto-role labeling in [2] may offer some good modeling ideas.

Do:

1. Write a dataset reader to ingest the datasets you created. The model inputs[3] should include:

   - tokenized sentence
   - head of predicate
   - head of argument

   and the model outputs should include:

   - binary decision on whether the role applies

   You may include any other inputs or outputs you deem appropriate. You should **not** include semantic proto-role information in the model inputs.

2. Design and implement a model architecture that adheres to the input-output interface specified by your dataset reader.

3. Train a model on each of your datasets.

4. Test each of your models on its appropriate test set. Use binary (not micro) F1 as the test metric.

Discuss:

1. Describe your model architecture and intuition behind your major design decisions.

2. If you included additional model inputs or outputs, discuss what they are and why you used them.

3. How well do your models perform on the test sets? Is performance correlated with the frequency of the role in the training set?

[1] `https://decomp.readthedocs.io/en/latest/tutorial/querying.html#custom-queries`
[2] The queries may take a long time to run over the entire corpus, so we advise first separating out the items you will query.
[3] Note that the token indices in the Decomp Toolkit are 1-indexed.

3.3 Exploration

Discuss:

1. Examine the training data you collected for one of the roles. Do any of the coarse semantic role labels seem counterintuitive or wrong? How would you modify your definitions to better capture the semantic roles? Can any set of UDS criteria perfectly capture the characteristics of a semantic role?

2. Write a minimal pair of sentences that use similar words but differ in the expected semantic roles. Do your models correctly predict the difference?

3.4 Extension

Discuss:

1. After building a separate model for each semantic role, you might wonder whether the roles are independent. What are two ways by which you could determine if any of the semantic roles you chose are correlated?

2. It seems wasteful to build a separate classifier for each semantic role. Besides saving memory, why would a multi-class or multi-label classifier be a reasonable model for this task?

3. Would a bag-of-words model using type-level word embeddings (e.g., GloVe embeddings) do well on this task? Why or why not?

4. The models you developed included (at a minimum) only text and predicate/argument indicator information. What are two other linguistically motivated inputs that may be useful for the task? Why would you expect them to help?

References

[1] David Dowty. "Thematic proto-roles and argument selection". In: *language* 67.3 (1991), pp. 547–619.

[2] Rachel Rudinger et al. "Neural-Davidsonian Semantic Proto-role Labeling". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 944–955. DOI: 10.18653/v1/D18-1114. URL: https://www.aclweb.org/anthology/D18-1114.

[3] Aaron Steven White et al. "The Universal Decompositional Semantics Dataset and Decomp Toolkit". English. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020, pp. 5698–5707. ISBN: 979-10-95546-34-4. URL: https://www.aclweb.org/anthology/2020.lrec-1.699.