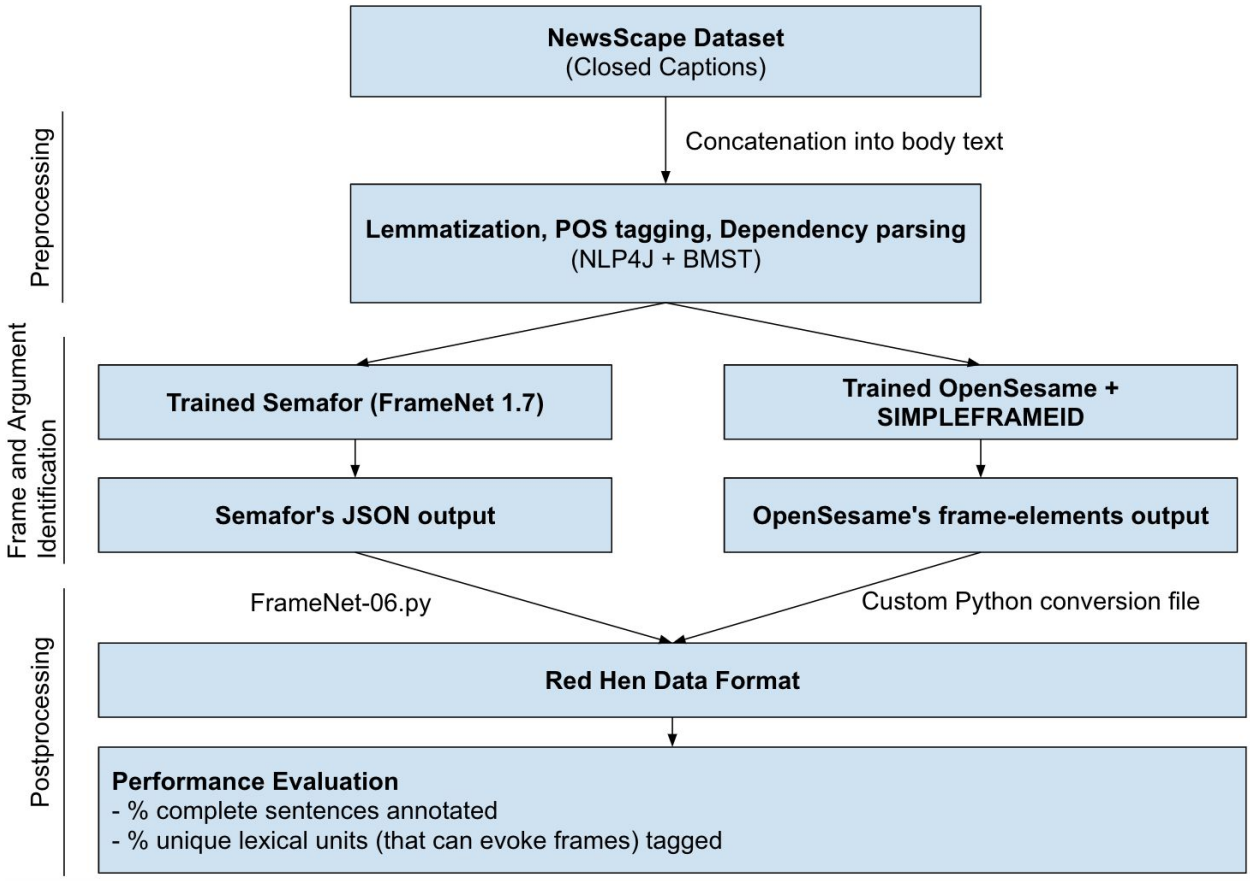
GSoC Phase 1: Report on the pyfn library in annotating NewsScape dataset

Yong Zheng Xin

Original Pipeline



Challenges

NewsScape dataset

The UCLA NewsScape data can be obtained from the gallina directory (`/mnt/rds/redhen/gallina/tv`) on CWRU HPC. The directory is structured chronologically.

I successfully created a Python script that uses regex to extract the concatenated closed caption text from the annotated NewsScape data (in .seg format) for the following pre-processing and annotation processes.

Lemmatization, POS tagging, and Dependency Parsing

For dependency parsing using the BMST parser, it requires the installation of DyNET 2.0.2. However, the DyNET 2.0.2 that is installed via `pip install` was not working for dependency parsing so I had to install manually through `cmake`, which was not possible on the HPC cluster as I was not one of the system administrators. Since pipeline had to be eventually deployed in a Singularity container, it means that I had to build my own Singularity container from my personal computer.

To overcome this obstacle temporarily and verify that the annotation pipeline can function, I performed pre-processing of the NewsScape data files on my personal computer where I am the system administrator. The inability to run the pipeline on the HPC clusters, taking advantage of the computing power necessary for training the neural network in the semantic parsers, signifies that my progress became significantly slower as I tested out the `pyfn` library on my personal Mac laptop.

Frame Identification with SimpleFrameID

When I started training the SimpleFrameID, I ran into a few bugs, and I opened the issues on GitHub. The biggest obstacle is that even though the task of SimpleFrameID is to identify the frames for a sentence, the code (sourced from the authors who publish the paper on SimpleFrameID) does not include an algorithm for identifying the target word responsible for the frame. This problem suggests that SimpleFrameID only works for the train, dev, and test datasets of FrameNet where the target words are readily identified.

I spent a few days generating the target words from the NewsScape dataset that has been annotated with FrameNet 1.5 to test out the above hypothesis, which turns out to be true. This means that to move forward, I have to search for a parser for identifying the target words in the NewsScape dataset based on FrameNet 1.7. This was not reported in the proposal because I didn't expect to encounter such a problem. After the discussion with Prof. Tiago, we came up with three potential methods to overcome this problem. The first is to explore Google SLING library. The second is to use Daisy that has been working

with FrameNet Brasil. The third is using a brute force heuristic to identify all the potential target words (in their lemma form) by checking their appearances in the FrameNet 1.7.

The first method (SLING library) seems to be more tedious because the library is not only dedicated to FrameNet. Apparently, I would have to create the training documents from FrameNet 1.7 train datasets using the SLING Python API, and the SLING library only works on a Linux machine. This means that it is very difficult for me to test out the SLING library on my personal Mac computer since I need to install a virtual machine. As I tested out the library on CWRU HPC, I ran into the error of the missing Intel license file necessary for training the SLING parser.

I am currently exploring the Daisy library for annotating the text. If possible, I would not want to resort to the third method because first, searching for all potential target words will be a really slow process given the size of the NewsScape dataset. Second, it seems that the precision of frame identification will deteriorate if a sentence contains multiple target words.

Argument Identification with SEMAFOR and Open-Sesame

Argument identification should be performed after the process of frame identification is completed, since the process identifies every role of each labeled frame instance and detects a span of words in the sentence that fill the semantic role.

Even though I couldn't test the two parsers with the NewsScape dataset because of the failure with frame identification, I ran into two major issues when I followed the tutorial to perform the argument identification using SEMAFOR. The first issue is that the output could not be converted from Semafor to Semeval XML data format. This impacts the following step on converting the data format into the Red Hen Data Format. Moreover, when I ran the SEMAFOR training in my personal computer, I could not perform training due to the error of missing `argmodel.dat`. To mitigate the issue, I had to train the model on the HPC cluster and download it. The second issue is that both parsers cannot be directly applied to a new dataset whose frames are predicted with SimpleFrameID. Upon quick investigation of the error `[INFO] StaticSemafor:112 - frameSplitsMap.size = 71`
`Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException`
, it seems that I needed to generate a file of specific frames elements corresponding to the predicted frames before I could apply the parsers.

Moving Forward with DAISY and Timeline

To use DAISY on NewsScape, I need to change the database of FrameNet Brasil to Berkeley FrameNet 1.7 because the former contains frames mainly targetting soccer games and tourism. After importing Berkeley FrameNet 1.7, I will test DAISY with a few NewsScape data file. Since the library relies a lot less on dependencies, I believe I can attempt deploying DAISY in a readily available Singularity container that contains the Python 3 environment.

Therefore, my timeline for week 4 is as followed:

June 17 - June 20: Change the database into Berkeley FrameNet 1.7 and ensure the code is working for annotating English sentences from NewsScape dataset

June 21 - June 22: If the code is working, deploy the library in a Singularity container. Otherwise, continue working on the issue and/or reach out to Prof. Tiago.