Report of Natural Language Processing, HW1

109550027 紀竺均

1. Describing your methods in detail.(50%)
   I did this homework based on the hint TA provided. And I mainly modified two parts.
   First, in the verb_idx function, I add a condition that if token.pos = “AUX”, I will also counted it as a verb since it might be passive or be-verb.
   Second, I turned the get_phrase function into two functions, get_subj and get_obj. In get_subj(), I iterated through the tokens which exist before the verb. That is, on the left hand side of the verb.

```
for token in sen[0:head_idx]:
```

   For each token, I check if token.dep is in my subject list tag, which contains

```
["nsubj", "nsubjpass", "csubj", "csubjpass", "agent", "expl"]
```

   If it does, add itself and its conjuncts into the return list.
   As for get_obj, it is quiet similar except it **focus on the right hand side** of the verb and have different tag list.

```
["dobj", "pobj", "dative", "attr", "oprd"]
```

   After the get_subj / get_obj function, I will use a for loop to add “every element” that return from the function instead of adding the whole return list.

```
if len(subj)!=0:
    for su in subj:
        s.append(su.text)
```

2. Is there any difference between your expectations and the results? Why?(20%)
   Yes. In fact, sometimes I add more constraints in the function, such as checking if the head-index is the same as the verb index, which I expected to have higher accuracy, end up having a lower accuracy. I guess the reason is that not every possible subject and object will be connected to the verb in this case. And adding that constraint make me missed some of the correct subject and object.

3. What difficulties did you encounter in this assignment? How did you solve it?(30%)
   I think the most difficult part of this assignment is to understand the meaning of the “tags” that Spacy using. Plus there aren’t plenty of reference on the internet. So it cost me some time to totally understand every tag and dep_ in POS tagging.
   Another obstacle is what I mentioned in Part 2 – Though I think of some clever method to improve the function, it may not be really helpful in practical

classifying. I think the best way to solve this is by testing and modifying.