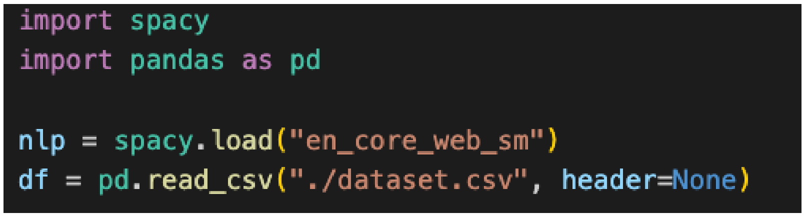
NLP HW1

109550003 陳茂祥

1 Methods

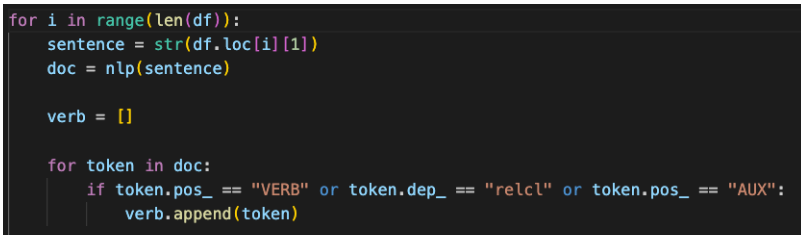
1. Include required packages and read the CSV file using pandas.



```python
import spacy
import pandas as pd

nlp = spacy.load("en_core_web_sm")
df = pd.read_csv("./dataset.csv", header=None)
```

2. Find out all words with POS-tagging ”VERB”, ”AUX”, and DEP-tagging ”relcl”.

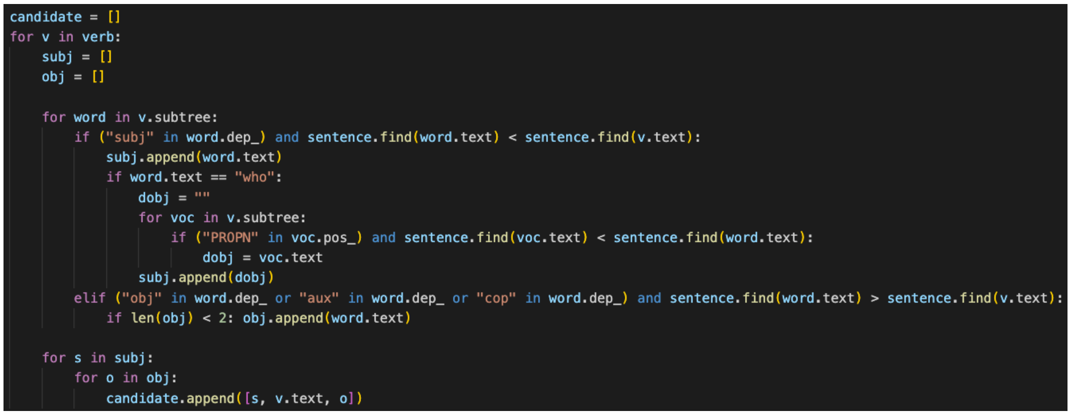


```python
for i in range(len(df)):
    sentence = str(df.loc[i][1])
    doc = nlp(sentence)

    verb = []

    for token in doc:
        if token.pos_ == "VERB" or token.dep_ == "relcl" or token.pos_ == "AUX":
            verb.append(token)
```

3. For each of them, use subtree property to find out all the other words with which they have relationship. Find the nearest subject in front of the verb and the nearest two objects behind the verb to form a list. Store it in List ”candidate” for later.

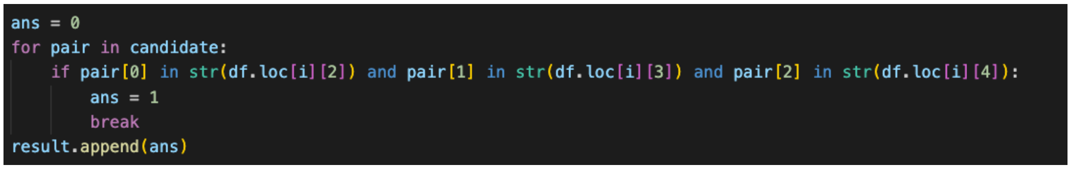


```python
candidate = []
for v in verb:
    subj = []
    obj = []

    for word in v.subtree:
        if ("subj" in word.dep_) and sentence.find(word.text) < sentence.find(v.text):
            subj.append(word.text)
            if word.text == "who":
                dobj = ""
                for voc in v.subtree:
                    if ("PROPN" in voc.pos_) and sentence.find(voc.text) < sentence.find(word.text):
                        dobj = voc.text
                subj.append(dobj)
        elif ("obj" in word.dep_ or "aux" in word.dep_ or "cop" in word.dep_) and sentence.find(word.text) > sentence.find(v.text):
            if len(obj) < 2: obj.append(word.text)

    for s in subj:
        for o in obj:
            candidate.append([s, v.text, o])
```

4. After finishing finding all possible pairs of [S,V,O], check whether any of them appear in the answer slot on the right at the same time. If any of them appear in the answer at the same time, then the output will be 1, otherwise, it will be 0.



```python
ans = 0
for pair in candidate:
    if pair[0] in str(df.loc[i][2]) and pair[1] in str(df.loc[i][3]) and pair[2] in str(df.loc[i][4]):
        ans = 1
        break
result.append(ans)
```

Improvements from simple to strong baseline

1. Limit the number of subjects and objects

My thought is that most sentences don’t have lots of objects for a single verb, in order to prevent from misusing the objects of other verbs, I limit the object number to two by testing through multiple numbers. By the way, I have tried including more candidates of subject, but the accuracy decreased.

2. Finding more POS-tagging and DEP-tagging

I run the program with the example_with_answer.csv and find the differences between my output and the answer. Finally, I found some useful tagging that is not included in the original code. I add the words with dependency tag ”relcl”(relative clause modifier) and the part of speech tag ”AUX”(auxiliary) to verb, and add the words with dependecy tag ”aux” and ”cop”(copula) to object. The accuracy increased about 0.1, which is a huge improvement.

3. Replace the relative pronoun with subject

I found out that the relative pronoun has dependency tagging of ”nsubj”, however, most of the answers of the subject slots contains the original subject rather than the relative pronoun. Thus, I replace the relative pronoun with the nearest ”nsubj” from the left-hand side. The accuracy increased slightly.

2 Questions

1) Is there any difference between your expectations and the results? Why?

1. The hint provided by TA recommend that adding auxiliary to objects will improve the accuracy. Although the accuracy really increased, I don't understand why an auxiliary can be seen as an object.
2. I have tried to precheck the POS tag of the answer. For instance, I checked whether there is any verb in the subject slot or any noun in the verb slot. However, a past-tense verb can serve as adjective and there are so many other possibilities. Therefore, it doesn't match my expectations.

2) What difficulties did you encounter in this assignment? How did you solve it?

1. Some of the verbs will not be recognized as VERB in spacy. Therefore, I analyzed these kinds of words and found out that dependency tag can provide certain information, one of them is ”relcl”. However, there are so many kinds of dependency tags that I can not find all of them by using the example dataset.