1. COMP5211 Winograd Schema Challenge

Name: Xiandong QI
Student ID: 20403284
Email: xqiad@cse.ust.hk
Date: 2017-11-13

1.1. Method-1: Answer by Bing Search Engine

1.1.1. Design Philosophy

For sentence “Lions are chasing the sheep because they are predators”. We will replace the target pronoun “they” by the correct antecedent and incorrect antecedent, “Lions” and “sheep” for above example separately. We believe that the search engine will pretend to provide more information about the “world knowledge”.

```
sent1 = "Lions are predators"
sent2 = "Sheep are predators"

print google_search(sent1)
print google_search(sent2)
```

The total number of response of above 2 queries are:

```
<div id="resultStats">約 24,600,000 項搜尋結果<nobr> (0.36 秒) </nobr></div>
<div id="resultStats">約 1,300,000 項搜尋結果<nobr> (0.38 秒) </nobr></div>
```

The algorithm would answer that the pronoun - “they” is “Lions” because that 24,600,000 > 1,300,000 which is intuitively correct.

1.1.2. Why not Google Search Engine?

When using Google Search Engine for the total 279 questions, we saw “503 Service Unavailable Error ”. This is because Google Search detects the automated traffic to Google sent by my network.

Unusual traffic from your computer network, so we switched to Bing search engine for answering the generated queries.

1.1.3. Evaluation of Bing Search Engine

```
==== The total number of questions is:  279  ====
Accuracy of Bing Search Engine:  51.61 %
```

1.2. Method-2: Answer by SVM/Machine Learning

1.2.1. Design Philosophy

1. Given training sentence, we split it into 2 clauses.
2. Find the key word (verb and adjective) of each clause by using $nltk.word_{t}okenize()$ and $nltk.pos_{t}ag()$ in $make_{t}ags()$ function.
3. Find the similarity of each pair of key word by using $wordnet.synsets()$ and $nltk.wup\_similarity()$ in $get\_sim\_value()$ function.
4. Find the positve or negative mood of each clause.
5. Return the above 6 features as a feature vector for each sentence.

```
# part of the "extract_feature" function
def extract_feature(sentence1, sentence2):

    sent1_tags = make_tags(sentence1)
    sent2_tags = make_tags(sentence2)

    for v in sent1_tags[::-1]:
        if is_verb(v):
            s1_v = v[0]
            break
        else:
            s1_v = "COMP5211"  # as a filler

    for adj in sent1_tags[::-1]:
        if is_adj(adj):
            s1_j = adj[0]
            break
        else:
            s1_j = "COMP5211"

    sim_s1_v_s2_v = get_sim_value(s1_v, s2_v)

    sent1_positive = 1
    if is_negative_from_list(sent1_tags):
        sent1_positive = -1

    return [sim_s1_v_s2_v, sim_s1_j_s2_j, sim_s1_v_s2_j, sim_s1_j_s2_v, sent1_p
```

1.2.2. Evaluation of SVM

```
# train model
clf = svm.SVC()
clf.fit(train_feature, train_target)

# predict answers
test_answer = clf.predict(test_feature)
```

```
==== The total number of queations is:  279  ====
Accuracy of SVM/machine learning:  53.16 %
```

1.3. Implementation

1.3.1. Installation

```
$ cd xxx-file-folder
$ pip install -U pip
$ pip install -r requirements.txt --force-reinstall
```

1.3.2. Run Code

```
$ python2 answer_by_Bing.py
```

```
$ python2 answer_by_machine_learning.py
```

1.4. Rethinking

1.4.1. Drawback of Search Engine

The Google/Bing search engine may return some unexpected results for our question. For example, “The police is chasing an accountant because he is a bad guy”. In search engine filled with news, it will answer “police is the bad guy” which is not the correct answer for our question.

1.5. Future work

The pain spot of Winograd Schema Challenge is that it does not have large amount training data which can be utilized as the commonsense knowledge like human have. It is possible to incorporate various commonsense knowledge bases, including ConceptNet, WordNet, Standford NLP as the knowledge base in training process.

1.6. Reference

[1] Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 777-789, 2012.
[2] http://www.google.de/search?
[3] http://www.bing.com/search
[4] http://www.nltk.org
[5] http://scikit-learn.org/stable/modules/svm.html