Comp5211 2019-Fall Final Project

COMP5211

Date assigned: Nov 9, 2019
Due time: 23:59pm on Dec 1, 2019
How to submit it: See the last section of the document.
Penalties on late submission: 20% off each day (anytime after the due time is considered late by one day)

Reasoning

The following two questions are exercises for you to use a mechanical theorem prover to solve problems by reasoning or finding models. The theorem prover you need to use is Z3. The official release is here:
`https://github.com/Z3Prover/z3`
If you use unix, you can install it using pip:
`https://pypi.org/project/z3-solver/#description`
Here is a good description of how to install it by Bob Moore of UT Austin:
`http://www.cs.utexas.edu/users/moore/acl2/manuals/current/manual/index-seo.php/SMT____Z3-INSTALLATION`

Problem 1: The Lady or The Tiger Problem (25 pts)

There are three rooms. Each contains either a lady or a tiger but not both. Furthermore, one room contained a lady and the other two contained tigers. Each of the rooms has a sign, and at most one of the three signs was true. The three signs are:

- Room I: A TIGER IS IN THIS ROOM.
- Room II: A LADY IS IN THIS ROOM.
- Room III: A TIGER IS IN ROOM II.

Which room contains the lady?

Problem 2: The Ranking Problem (25 pts)

Given the following facts:

1. Lisa is not next to Bob in the ranking

2. Jim is ranked immediately ahead of a biology major

3. Bob is ranked immediately ahead of Jim

4. One of the women (Lisa and Mary) is a biology major

5. One of the women is ranked first

What are possible rankings for the four people?

NLP - Problem 3: Evaluating pretrained langauge model BERT (50 pts)

In this problem, your task is to investigate recent advances in pretrianed natural language models. More specifically, you are required to use a BERT pretrained model to perform a task that you designed and report on your experiments.

BERT [1], short for Bidirectional Encoder Representations from Transformers, was proposed in 2018 by Google AI Research. It significantly pulled up the state-of-the-art performance on 11 NLP tasks, such as machine reading comprehension, natural language inference, etc.. The core contribution of BERT is to capture the complex world-knowledge of words via unsupervised large-scale pretraining, improving the representation ability of words and sentences.

Once published, BERT has been used on many NLP applications. Variants of BERT, such as XLNet, RoBERTa, and Electra, have been presented. It makes great influence on the way of natural language processing research. Researchers are not only exploring the how to use BERT-like pretrained models, but also why BERT-like models work. For example, in EMNLP2019 (5 Nov. 2019), there are 17 accepted paper directly examining BERT, and BERT-like models have become standard baselines in almost all NLP tasks.

As is well-known, training large-scale BERT model is resource and time consuming. Therefore, in this project, you are ONLY required to explore BERT by using a well-trained BERT model without re-training to solve your tasks. Specifically, you are required to use the pretrained model called `BERT-small-uncased` [2], querying it with only lower-cased English texts.

Experiment Environment

The experiments are based on a service produced by Han Xiao [3], named bert-as-serivce. The advantages of it are:

1. it is very fast for inference, compared to a standard Tensorflow / PyTorch framework

[1] link: `https://arxiv.org/pdf/1810.04805.pdf`

[2] donwload from here `https://storage.googleapis.com/bert_models/2018_10_18/uncased_L-12_H-768_A-12.zip`

[3] `https://github.com/hanxiao/bert-as-service`

2. it is so light that it can run on even laptop. Therefore, you can install and paly with BERT on your own computer, without waiting for public resources.

3. it provides "for dummies" API, you can query and get the inference results with few lines of code.

We also provide a department server for querying BERT, but it may get slow when there are too many queries simultaneously, as it is a public server for all the members in CSE department. Therefore, you are encouraged to use your own computing resources.

The detailed instruction and tutorials is in `https://github.com/CodePothunter/COMP5211_Project`

Your task

For this problem, you are to use the BERT model to perform a specific task. Examples of the possible tasks include but are not limited to:

- Investigate if the BERT model includes some common sense knowledge, and if so to what degree. Examples of common sense knowledge: Every man is mortal, the earth is round, and so on. You may find the following paper relevant: `https://arxiv.org/pdf/1909.01066v2.pdf`.
- Investigate if the BERT model knows some relationships about words. Examples: If John loves Mary, then John likes Mary. If it rains, then the road is wet. Somewhat related to the last tasks but more sophisticated.
- Investigate if the BERT model understands some connectives like "not".
- others...

1 Submission

Please submit your solution on Canvas before the due date. You should submit a zip file named as `YourStudentID.zip`, which includes:

- for Problem 1: `lady.py`, which is a z3 query file for your answer. For example, if your answer is that the lady is in room 2, and you use l2 to denote it, then your z3 query file is to check whether not(l2) is consistent with the KB of this problem. For example, the following python code is a z3 query file to check if q follows from p and $p \supset q$:

```
-------------------------------------------------------------
from z3 import *

p = Bool('p')
q = Bool('q')
s = Solver()
s.add(Implies(p,q))
```

```
s.add(p)
# to prove q, add not(q) to see if it causes a contradiction
s.add(not(q))
print(s.check())
-----------------------------------------------------------------
```

- for Problem 2: `ranking.py`, which is your z3 query file to find a model from which to read out a ranking. For example, the following is a python code to compute a model of $p \vee q$ and $\neg(p \wedge q)$:

```
-----------------------------------------------------------------
from z3 import *

p = Bool('p')
q = Bool('q')
s = Solver()
s.add(or(p,q))
s.add(not(and(p,q)))
print(s.check())
print(s.model())
-----------------------------------------------------------------
```

- for Problem 3: `report.pdf` and `query.txt`. Your pdf report file must clearly describe the task that you have chosen to experiment with the BERT model, and your experimental results: how do you query the BERT model, how many queries you have made, what are the results, how good the results are in your opinions. Don't forget to include references that you have read for this problem. Your 'query.txt' file should be a standard text file including all your queries and the answers from the BERT model.

You should give credit to what you have referenced to. Otherwise, it will be regarded as plagiarism.