CSC321: Project2

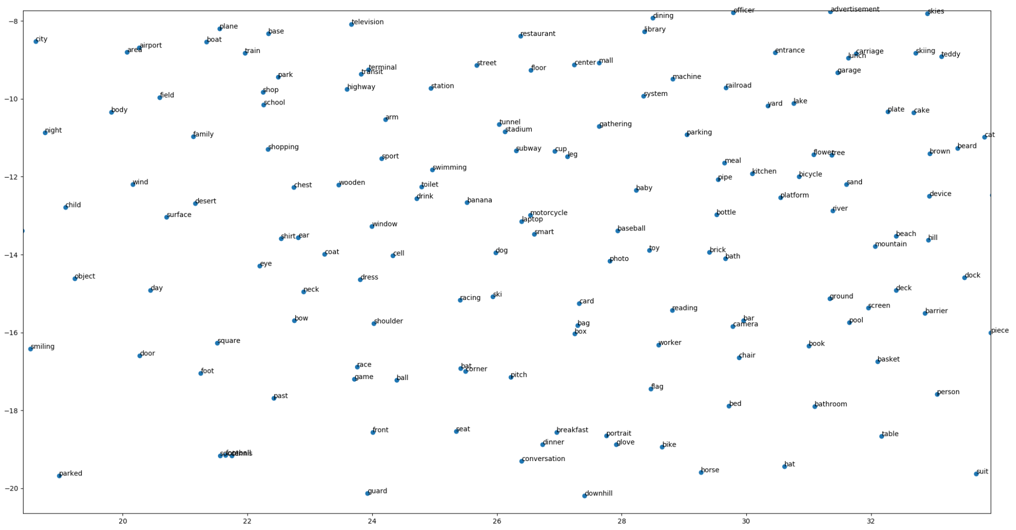
Due on Monday, February 22, 2017

Xiaodi Lu

February 23, 2017

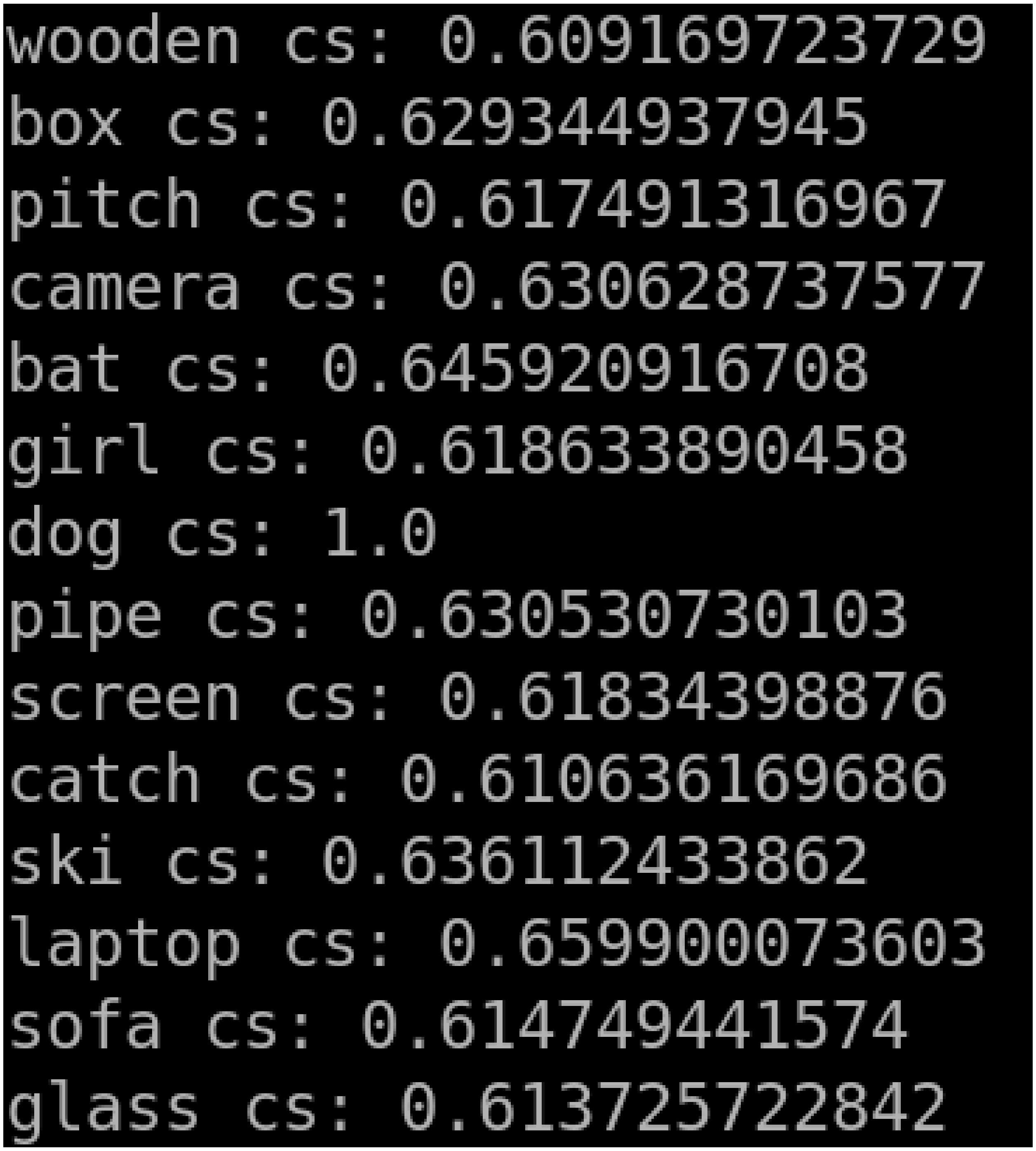
Part 1

Visualize Word Embedding



The image above is a selection of words that looks close to dog in 2d dimension.(the word "*dog*" is at around [26, -14])

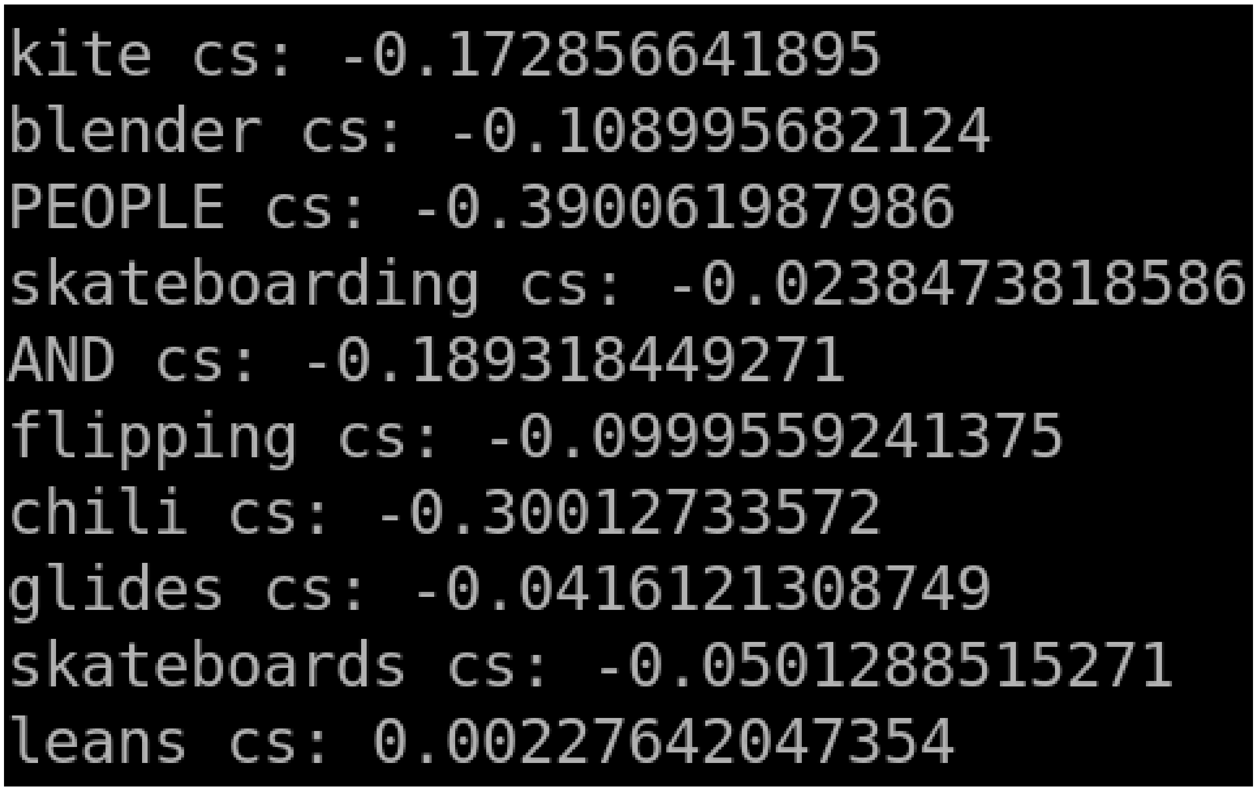
The cosine similarity that is larger than 0.6 for words with "*dog*" are the following with its cosine similarity score:



```
wooden cs: 0.609169723729
box cs: 0.629344937945
pitch cs: 0.617491316967
camera cs: 0.630628737577
bat cs: 0.645920916708
girl cs: 0.618633890458
dog cs: 1.0
pipe cs: 0.630530730103
screen cs: 0.61834398876
catch cs: 0.610636169686
ski cs: 0.636112433862
laptop cs: 0.659900073603
sofa cs: 0.614749441574
glass cs: 0.613725722842
```

It is clear to see that the words that are similar to the word "*dog*" are the objects that the dog usually play with or the dog is usually around. The words are mostly nouns with 2 verbs each corresponding to a activity that the dog usually participate in. The nouns are objects that dogs sometimes destroys or around, like box.

The following words are a selection that have cosine similarities with the word "*dog*" smaller than 0.01:



```
kite cs: -0.172856641895
blender cs: -0.108995682124
PEOPLE cs: -0.390061987986
skateboarding cs: -0.0238473818586
AND cs: -0.189318449271
flipping cs: -0.0999559241375
chili cs: -0.30012733572
glides cs: -0.0416121308749
skateboards cs: -0.0501288515271
leans cs: 0.00227642047354
```

In the words there are nouns like "*AND*" and verbs like "*skateboarding*". The words are not usually the ones we associate with "*dog*". Actually those things have nothing to do with dogs. For example like the word "*PEOPLE*", the word "*dog*" is very different from people, although they have the same part of speech. People represents human and dog is a kind of animal that is popular for pets.

In general, we found that the higher the score of cosine similarity, the more two words have in common in every day life, they are more associated with each other. The lower the score, the less they are associated with each other in daily life.

Part 2

Train and Evaluate Model

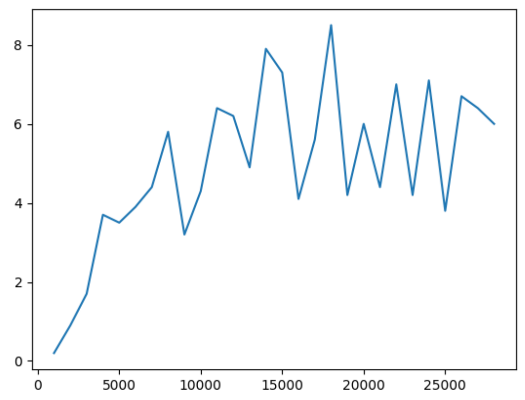


Figure 1:
Learning rate:0.43
Momentum:0.23
BLEU:7.9

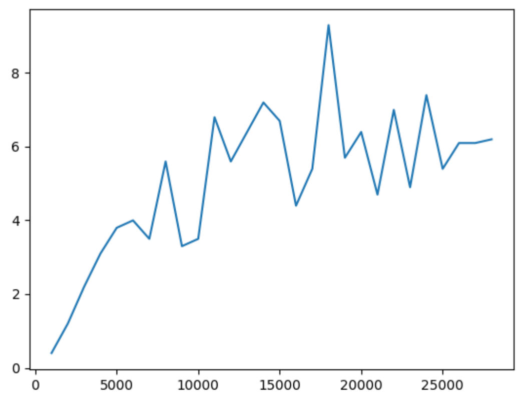


Figure 2:
Learning rate:0.51
Momentum:0.23
BLEU:9.3

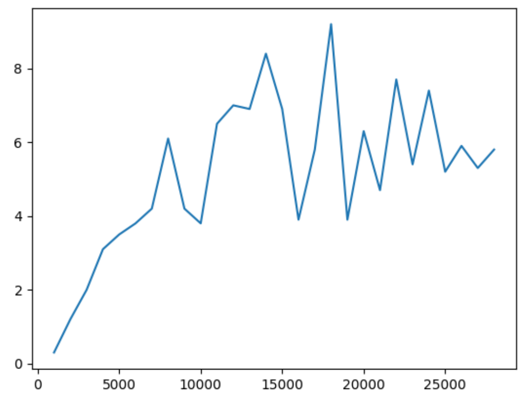


Figure 3:
Learning rate:0.51
Momentum:0.30
BLEU:8.7

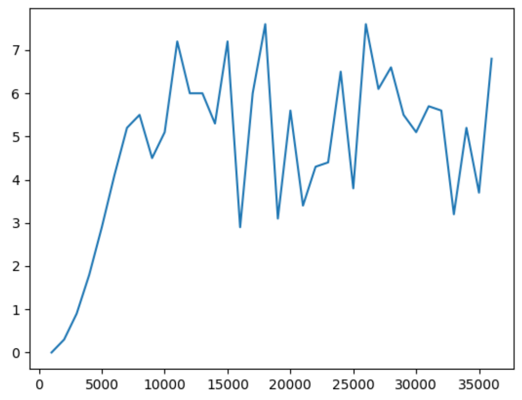


Figure 4:
Learning rate:0.30
Momentum:0.51
BLEU:7.4

Question: Try different choices of learning rate and momentum and compare the corresponding validation performance. You should at least experiment with two different values of learning rate and momentum respectively (totally 4 settings including the one already in the code). Show the curve of BLEU scores on the validation set during training (Hint: you can change *prog*[*_bleu*] in the trainer.py to modify how often the program computes the BLEU score). Explain why one set of hyperparameters might work better than another.

The y axis corresponds to BLEU score, and x axis corresponds to the number iteration.

Figure 2 graph has a higher best *bleu* score than Figure 1 because the learning rate in the second one is higher. The higher the learning rate, the higher the changing rate, and the momentum is best at 0.23. It is clear to see while the curve is vibrating the graph with higher learning rate takes bigger learning steps therefore is faster to train, and the variation of weight with each update is greater. Because we have a higher changing rate and we only store the *bleu* score if it is higher than the one we have, the probability of it getting a higher *bleu* score is higher, since we could jump really high.

Choose the best model based on the validation performance. Test the model and report the BLEU score on the testing set.

The best model has:
Learning rate: 0.51
Momentum: 0.23
BLEU: 9.3
BLEU score on the testing set: 7.7

To understand what the model has learned, you can run *run_interpreter.py* to obtain the $top - K$ probable candidates out of the whole vocabulary for each word in the sentence, where K is the beam width. You can tweak the beam width and explain what you found from the candidate words. For example, you may find that word *am* follows word *I* at a very high probability.

```
top 5 (word, log probability) @ time step 1:
(A, -9.4957963384)
(The, -9.85236566781)
(a, -9.1658069138)
(An, -8.85216406113)
(Two, -9.53136140834)
top 5 (word, log probability) @ time step 2:
(man, -10.0469853239)
(group, -11.2917571294)
(young, -8.05689819384)
(person, -9.42425360334)
(woman, -11.1916691526)
top 5 (word, log probability) @ time step 2:
(man, -8.80895478638)
(young, -10.126192815)
(tennis, -7.43497600697)
(woman, -8.45688466156)
(court, -9.86095830855)
top 5 (word, log probability) @ time step 2:
(man, -8.95683860251)
(tennis, -10.6928247784)
(young, -7.77088637316)
(woman, -8.68987493337)
(court, -10.2123905241)
top 5 (word, log probability) @ time step 2:
(young, -8.74996788666)
(unk, -9.95636857765)
(tennis, -7.74101210854)
(man, -7.81840264631)
(woman, -9.16411898798)
top 5 (word, log probability) @ time step 2:
(people, -8.72216765832)
(unk, -8.7561774326)
(young, -7.97039543431)
(tennis, -7.77553922395)
(children, -8.50588234851)
```

From the partial output with beam width equal to 5 above, we could see that the wording actually makes a lot of sense in every day life, like nouns like $'man'$ or $'woman'$ most likely follow articles like $'A'$ or $'a'$ or $'the'$. After setting the beam width to 2, more repetitive words are given in the output, less new words.

Look through the results of good or bad captions and trying to explain why they occurred.

good captions:
A group of people standing on the road
A couple of people

bad captions:
A group of people in the road
A man of a man
A man of people in the road

Good captions occurs because through the probability of the next words given context, the model could generate sentences with the probability it calculates. Since the model is trained on the sentences that we use in every day life, the sentences the model generated, with respect to grammar, is mostly correct.
Bad captions occurs because the context size is too small. Sometimes the model failed to look at the whole sentence to detect whether it has a high probability of occurrence in the data set, it only looks at a small context. And the model also only look at the probability of the next words not whether the words would all occur in the correct order in the sentence.