Assignment 4

Name: Zian Wang

1. How did you design your opinion extraction module with CoreNLP?

For the opinion extraction, I use Enhanced Dependencies annotation result from coreNLP. I use the annotate function to get the annotation results. There are three strategies to extract opinions from the coreNLP result.

(1). Find all “nsubj” in the results. We will use “dependentGloss” as the attribute and “governorGloss” as the value. This is because in nsubj, the dependentGloss values are nouns, and the governorGloss values are adjectives.

(2). Find all “amod” in the results. The difference here is we will use “governorGloss” as the attribute, and use “dependentGloss” as the value. In amod, the governorGloss values are nouns and the dependentGloss are the adjectives.

(3). Find all “compound” in the results. We find compound because some attributes are constructed by two nouns, for example: California salad, and these attributes will be annotated as compound. However, we need one more step to see if it is an attribute. We need to see if the former word in compound has amod dependency, or if the latter word has nsubj dependency. If one of these two conditions is satisfied, we will denote these two words as an attribute. For example, if the sentence is “This is a good bread stick.” The compound will be (bread, stick), and also there will be an amod: (bread, good). Therefore, we firstly find (bread, stick), and then find bread has an amod dependency (bread, good), so we denote “bread stick” being together is an attribute. Another example is “Waffle fries are good.” The compound in this sentence is (waffle, fries), and the although waffle does not have amod dependency in this sentence, fires has a nsubj dependency in this sentence which is (fries, good). Therefore, we denote “waffle fries” together as an attribute.

There is another filter in the extracting task, which is a black list named “not_consider”. This is a list that stores some words like “we”, “I”, and “He”. When we find the attribute is one of these words, we will just abandon this opinion. The reason is if an opinion’s attribute is, let’s say, we, it is highly likely that this opinion is not for the restaurant.

2. How did you measure the opinion similarity? How do you tune the threshold?

For the opinion similarity, we calculate the cosine of two opinions’ attributes, then calculate cosine of tow opinions’ values. Only if these two calculated cosine is both larger than the threshold, these two opinions will be denoted as similar to each other. In my opinion, this way can ensure the two opinions’ attributes and values are both similar. If an opinion has three words, for example, (California salad, huge), we will firstly find opinions whose attribute part has only one word, and compare the attribute to “salad”, which is the second word in the attribute. Then secondly, we will find opinions whose attribute has two words, just like (California salad, huge) itself. Then, we will compare the words in the corresponding positions in the two opinions respectively. For example, if the two opinions are (California salad, huge) and (soup, large), we will compare salad with soup since salad is the second word in the attribute. Then compare huge with large. Also, if the opinions are (California salad, huge), and (Philly steak, big), we will compare California with Philly, then compare salad with steak, and finally is huge with big.

The method I tune the threshold is firstly trying cosine threshold from 0, 0.1, 0.2 to 1. We can find when we set threshold to a high value, like 0.8, there are no similar opinions returned. If we set it to 0.7, there is only one similar opinion of each query, and for no similar opinion for (food, delicious). When the threshold comes down to 0.5, there start to exist similar opinions for (food, delicious). But the results are still far less than the manually denoted results. Therefore, I continue reducing the threshold, and when it comes to 0.3, the result is pretty close to the manually denoted one. But if we set it to 0.2, there will be too many result opinions found. Therefore, I start trying 0.29, 0,28, ..., to 0.25. Finally, I found 0.26 is the best cosine threshold I can find to make the result as close as possible to the manually denoted one. If we keep reducing the threshold, there will be many non-sense opinions, for example, it will say (food, great) is similar to (service, great), and (atmosphere, good) is similar to (California salad, huge). Also, the results are far too more. Therefore, we choose 0.26 as our tuned threshold since the results is closest to the manually denoted one.

3. Discuss the successful cases that your system can handle.

My system can successfully handle most of the cases in the query (service, good), (atmosphere, good), and (food, delicious). It can find very similar opinions, like (service, great), (service, excellent) for query (service, good). This is intuitive since these opinions' attributes and values are quite similar to the query's. The threshold we tuned makes sure most of these opinions will be found.

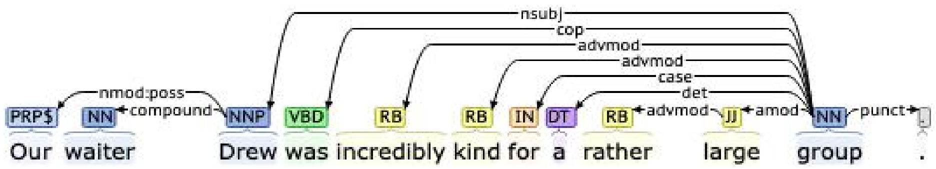
Also, the system can find many trick opinions, for example, (fish taco, good), (waffle fries, amazing), and (bread stick, delicious) for query (food, delicious). This is because firstly we have codes to handle finding these three words opinions, also because the threshold we tuned is good.

In conclusion, if the opinions' attributes and values are similar to the query, most of these opinions will be found, also includes some little complex opinions like (bread stick, delicious).

4. Discuss the cases that your system fails.

This part is quite interesting since we can know what is my system's disadvantage is through this part. There are two types of failure, the first is the system fails to extract some opinions, the second is the system successfully extract the opinions, but it fails to match it to the similar query.

For the first type of failure, the main reason I found is because the enhanced dependencies annotation also fails. For example, the system fails on extracting the opinion (waiter, kind). I put review 16 into the demo website to see the result, and it comes out this:



We can see there is no nsubj dependency of waiter exist. Also we can see the JJ we want is kind, but its dependency is on the word "group". Therefore, the enhanced dependency annotation does not think the word "waiter" has obvious link to the word "kind", so my system cannot extract this one. Also, the same reason for failure on (atmosphere, fun). Enhanced dependency thinks the word “fun” here is a NNP, so atmosphere and fun only have compound dependency here. The detailed reason for failures on different opinions are quite diverse. Except two examples we mentioned above, there are some other reasons, for example, grammatical errors (“the potatoes ... yum.” in review 2) and spell mistakes (“slow to serve” in review 19) in the reviews. I tried to find some more strategies to solve these failures, but I did not find until now, more strategies are needed in the future.

For the second type of failure, which is the system successfully extract the opinion, but fails to match it to the similar query, the first reason is the pre-trained word embedding we use. For example, the query (service, bad), the similar opinions my system finds are all good opinions for the service. For example, it finds (service, good), (service, great). It is very counter-intuitive since we know “bad” and “good” are antonyms. But if we use the word2VecObject.most_similar() function to see the similar words list of “bad”, we can find this result.

```
000 = {tuple: 2} ('good', 0.7190049886703491)
001 = {tuple: 2} ('decent', 0.40417149662971497)
002 = {tuple: 2} ('nice', 0.399969220161438)
003 = {tuple: 2} ('just', 0.3929983377456665)
004 = {tuple: 2} ('great', 0.392876535654068)
005 = {tuple: 2} ('big', 0.3885195255279541)
006 = {tuple: 2} ('hard', 0.38604509830474854)
007 = {tuple: 2} ('ok', 0.3751773536205292)
008 = {tuple: 2} ('positive', 0.3737700283527374)
009 = {tuple: 2} ('probably', 0.3643215298652649)
010 = {tuple: 2} ('like', 0.3633710741996765)
011 = {tuple: 2} ('sick', 0.35692980885505676)
012 = {tuple: 2} ('do', 0.3550606667995453)
013 = {tuple: 2} ('best', 0.35135000944137573)
014 = {tuple: 2} ('really', 0.34899231791496277)
015 = {tuple: 2} ('actually', 0.34788426756858826)
016 = {tuple: 2} ('rude', 0.344826877117157)
017 = {tuple: 2} ('cold', 0.33403071761131287)
018 = {tuple: 2} ('think', 0.33214449882507324)
019 = {tuple: 2} ('greasy', 0.3295784592628479)
020 = {tuple: 2} ('kind', 0.3273546099662781)
021 = {tuple: 2} ('not', 0.32701149582862854)
022 = {tuple: 2} ('pleasant', 0.3228386640548706)
023 = {tuple: 2} ('useless', 0.3207995593547821)
024 = {tuple: 2} ('definitely', 0.32037752866744995)
025 = {tuple: 2} ('awesome', 0.3199988007545471)
026 = {tuple: 2} ('lot', 0.31911003589630127)
027 = {tuple: 2} ('it', 0.3123636543750763)
028 = {tuple: 2} ('but', 0.3049958050251007)
029 = {tuple: 2} ('solid', 0.30313631892204285)
030 = {tuple: 2} ('wonderful', 0.3011701703071594)
031 = {tuple: 2} ('too', 0.3011515140533447)
032 = {tuple: 2} ('thing', 0.30096641182899475)
033 = {tuple: 2} ('picky', 0.30083492398262024)
034 = {tuple: 2} ('excellent', 0.298440545797348)
035 = {tuple: 2} ('that', 0.29686781764030457)
036 = {tuple: 2} ('pretty', 0.2955917716026306)
037 = {tuple: 2} ('little', 0.29363948106765747)
```

We can find in this pre-trained word embedding file, the most similar word to “bad” is good. The similarity is about 0.719. Then is the word “decent”, and then “nice”. We can find the first bad meaning word in this list is “sick” ranked 11th, and the similarity is 0.357, not to mention the word we want: “bad” is not even in this list. Therefore, the system do “find” many counter-intuitive opinions.

The second reason for the second type of failure is the threshold we set. Although we use the cosine threshold 0.26, there is some opinions that can be manually found, failed to be matched by my system. For example, the opinion (meat, tender) for query (food, delicious). The failure on this opinion is because the word "tender" only has 0.2 similarity to the word "delicious". Therefore, although we think the meat is tender should be found, the system does not find it since the threshold is 0.26 > 0.2. This is like a trade-off between finding all manually found opinions with not finding too many non-sense opinions. After all, if we set the cosine threshold too small, many unrelated opinions will also be found and put into the results.

5. Other explanations.

I use Stanford coreNLP package downloaded from the link provided. Also, I use the pre-trained word embedding file "assign4_word2vec1.bin". The materials provided help a lot, especially the demo website of coreNLP, it helps a lot when I cross-check the results of my system. The version of Python is 3.9, and the version of Java is 16.0.2.

6. The output of your algorithm with your best results (threshold).

The cosine threshold I tuned is **0.26**. Here are the results.

[meal, wonderful] appears in review [1]

[California salad, huge] appears in review [1]

[California salad, delicious] appears in review [1]

[salad, huge] appears in review [1]

[salad, delicious] appears in review [1]

[stick, little] appears in review [1]

[bread stick, little] appears in review [1]

[bread stick, delicious] appears in review [1]

[stick, delicious] appears in review [1]

[Service, excellent] appears in review [1 , 2]

[This, place] appears in review [1]

[place, great] appears in review [1]

[experience, upscale] appears in review [1]

[lunch experience, upscale] appears in review [1] [spots, other] appears in review [1]

[Delicious restaurant, old] appears in review [2]

[restaurant, old] appears in review [2]

[bar restaurant, old] appears in review [2]

[woodwork, ornate] appears in review [2]

[tablecloths, white] appears in review [2]

[potatoes, red] appears in review [2]

[potatoes, skinned] appears in review [2]

[potatoes, mashed] appears in review [2]

[meat, tender] appears in review [2]

[meat, flavorful] appears in review [2]

[Cole slaw, delicious] appears in review [2]

[slaw, delicious] appears in review [2]

[atmosphere, nice] appears in review [3]

[list, local] appears in review [3]

[draft list, local] appears in review [3]

[draft list, great] appears in review [3]

[list, great] appears in review [3]

[fish taco, world] appears in review [3]

[taco, world] appears in review [3]

[fish sandwich, world] appears in review [3]

[sandwich, world] appears in review [3] [potions, HUGE] appears in review [3]

[menu, huge] appears in review [3]

[that, thing] appears in review [3]

[thing, good] appears in review [3]

[people, regulars] appears in review [3]

[people, echoed] appears in review [3]

[note, positive] appears in review [4]

[meal, delicious] appears in review [4]

[sandwich, grilled] appears in review [4]

[chicken sandwich, grilled] appears in review [4]

[waffle fries, amazing] appears in review [4]

[fries, amazing] appears in review [4]

[option, choose] appears in review [4]

[toppings option, choose] appears in review [4]

[toppings option, something] appears in review [4]

[option, something] appears in review [4]

[atmosphere, has] appears in review [4]

[ambiance, nice] appears in review [4]

[they, seemed] appears in review [4]

[selection, great] appears in review [4]

[server, rude] appears in review [4]

[server, visited] appears in review [4] [meals, are] appears in review [4]

[thing, note] appears in review [4]

[restaurant, small] appears in review [4 , 5]

[tables, close] appears in review [4]

[hostess, asked] appears in review [4]

[table, OWN] appears in review [4]

[ride, useless] appears in review [5]

[Uber ride, useless] appears in review [5]

[distance, walking] appears in review [5]

[meals, best] appears in review [5]

[fan, huge] appears in review [5]

[Service, great] appears in review [5]

[drinks, large] appears in review [5]

[Girlfriend, ordered] appears in review [5]

[sandwiches, French] appears in review [5]

[dip sandwiches, French] appears in review [5]

[restaurant, nice] appears in review [5]

[atmosphere, great] appears in review [5]

[food, great] appears in review [5]

[place, rotation] appears in review [6]

[rotation, regular] appears in review [6]

[lunch rotation, regular] appears in review [6] [food, Excellent] appears in review [6]

[Turkey Devonshire, awesome] appears in review [6]

[Devonshire, awesome] appears in review [6]

[Chicken tenders, disappoint] appears in review [6]

[tenders, disappoint] appears in review [6]

[deserts, recent] appears in review [6]

[deserts, mini] appears in review [6]

[bottle, deal] appears in review [6]

[deal, great] appears in review [6]

[draft selection, good] appears in review [6]

[beer selection, good] appears in review [6]

[selection, good] appears in review [6]

[mugs, seem] appears in review [6]

[They, undercooked] appears in review [7]

[They, tasteless] appears in review [7]

[quality, bad] appears in review [7]

[They, care] appears in review [7]

[humbleness, is] appears in review [7]

[day, bad] appears in review [7]

[food, Good] appears in review [8]

[food, hearty] appears in review [8]

[prices, decent] appears in review [8] [service, warm] appears in review [8]

[beer fish, battered] appears in review [8]

[fish, battered] appears in review [8]

[Rainbow trout, treat] appears in review [8]

[trout, treat] appears in review [8]

[good, bad] appears in review [8]

[service, Bad] appears in review [9]

[fingers, homemade] appears in review [1 0]

[chicken fingers, homemade] appears in review [1 0]

[Burgers, good] appears in review [1 0]

[beer, Good] appears in review [1 0]

[beer, priced] appears in review [1 0]

[value, great] appears in review [1 0]

[menu, large] appears in review [1 1]

[portions, larger] appears in review [1 1]

[prices, reasonable] appears in review [1 1]

[quality, excellent] appears in review [1 1]

[They, have] appears in review [1 1 , 2 0]

[menu, entire] appears in review [1 1]

[menu, dedicated] appears in review [1 1]

[specialities, Other] appears in review [1 1]

[specialities, cakes] appears in review [1 1] [wait, come] appears in review [1 1]

[restaurant, favorites] appears in review [1 2]

[favorites, top] appears in review [1 2]

[favorites, local] appears in review [1 2]

[restaurant, has] appears in review [1 2]

[feeling, warm] appears in review [1 2]

[They, offer] appears in review [1 2]

[bottles, deal] appears in review [1 2]

[deal, cheapest] appears in review [1 2]

[wine deal, cheapest] appears in review [1 2]

[you, find] appears in review [1 2]

[portions, generous] appears in review [1 2]

[order, particular] appears in review [1 2]

[you, hungry] appears in review [1 2]

[you, wait] appears in review [1 2]

[course, main] appears in review [1 2]

[soup, French] appears in review [1 2]

[onion soup, French] appears in review [1 2]

[Summer, last] appears in review [1 3]

[time, wonderful] appears in review [1 3]

[food, excellent] appears in review [1 3]

[food, priced] appears in review [1 3] [service, solid] appears in review [1 3]

[group, larger] appears in review [1 3]

[Grill, Best] appears in review [1 4]

[Reuben Grill, Best] appears in review [1 4]

[Reuben Grill, made] appears in review [1 4]

[Grill, made] appears in review [1 4]

[turkey, roasted] appears in review [1 4]

[service, great] appears in review [1 4]

[food, fresh] appears in review [1 4 , 1 8]

[food, interesting] appears in review [1 4]

[fish taco, good] appears in review [1 5]

[taco, good] appears in review [1 5]

[waiter, slow] appears in review [1 5]

[time, long] appears in review [1 5]

[thing, second] appears in review [1 5]

[thing, sure] appears in review [1 5]

[food, cold] appears in review [1 5]

[waiter Drew, group] appears in review [1 6]

[Drew, group] appears in review [1 6]

[group, large] appears in review [1 6]

[food, satisfying] appears in review [1 6]

[food, strike] appears in review [1 6] [food, typical] appears in review [1 6]

[food, American] appears in review [1 6]

[way, bad] appears in review [1 6]

[options, Vegan] appears in review [1 6]

[options, are] appears in review [1 6]

[you, take] appears in review [1 6]

[salad, Large] appears in review [1 7]

[salad, lot] appears in review [1 7]

[Salad, Greek] appears in review [1 7]

[nachos, Good] appears in review [1 7]

[fries, great] appears in review [1 7]

[waffle fries, great] appears in review [1 7]

[waiter, friendly] appears in review [1 7]

[waiter, attentive] appears in review [1 7]

[you, heard] appears in review [1 8]

[They, wines] appears in review [1 8]

[wines, decent] appears in review [1 8]

[Pinot Noir, favorite] appears in review [1 8]

[Noir, favorite] appears in review [1 8]

[food, Fast] appears in review [1 8]

[restaurants, busy] appears in review [1 9]

[hopes, high] appears in review [1 9]

[food, OK] appears in review [1 9]

[service, good] appears in review [2 0]

[ambience, pleasant] appears in review [2 0]

[Pittsburghers, older] appears in review [2 0]

[game, airing] appears in review [2 0]

----

query opinion [service, good] has similar opinions:

 [Service, great] appears in review [5]

 [Service, excellent] appears in review [1 , 2]

 [service, great] appears in review [1 4]

 [service, solid] appears in review [1 3]

 [service, warm] appears in review [8]

 [service, good] appears in review [2 0]

query opinion [service, bad] has similar opinions:

 [Service, great] appears in review [5]

 [Service, excellent] appears in review [1 , 2]

 [service, good] appears in review [2 0]

 [service, great] appears in review [1 4]

 [service, solid] appears in review [1 3]

query opinion [atmosphere, good] has similar opinions:

 [ambience, pleasant] appears in review [2 0]

 [ambiance, nice] appears in review [4] [feeling, warm] appears in review [1 2]

[atmosphere, great] appears in review [5]

[atmosphere, nice] appears in review [3]

query opinion [food, delicious] has similar opinions:

[meal, wonderful] appears in review [1]

[meal, delicious] appears in review [4]

[meat, flavorful] appears in review [2]

[bread stick, delicious] appears in review [1]

[chicken fingers, homemade] appears in review [1 0]

[restaurant, nice] appears in review [5]

[beer selection, good] appears in review [6]

[fish taco, good] appears in review [1 5]

[salad, delicious] appears in review [1]

[California salad, delicious] appears in review [1]

[turkey, roasted] appears in review [1 4]

[taco, good] appears in review [1 5]

[fries, amazing] appears in review [4]

[waffle fries, amazing] appears in review [4]

[fries, great] appears in review [1 7]

[waffle fries, great] appears in review [1 7]

[waiter, attentive] appears in review [1 7]

[food, hearty] appears in review [8]

[food, satisfying] appears in review [1 6]

[food, great] appears in review [5]

[food, fresh] appears in review [1 4 , 1 8]

[food, interesting] appears in review [1 4]

[food, excellent] appears in review [1 3]
