CS372 HW5 report

20160632 Janghyun Cho

Our goal is to figure out which name does the given ambiguous pronouns indicates. We are given ID, a paragraph, pronoun, pronoun's position, nounA, nounA's position, whether nounA is coreferent name, nounB, nounB's position, whether nounB is coreferent name, and url of text. For example, we are given input as [test-1, "some sentence", "his", 383, "Bob Suter", 352, False, "Dehner", 366, True, http://en.wikipedia.org/wiki/Jeremy_Dehner]. Then we need to make two test cases. One using url, page-context, and other not using url, snippet-context. So we need to find relationship between nouns in the paragraph and give one of three cases. Pronoun indicates nounA, pronoun indicates nounB, and pronoun indicates neither of them.

To solve the problem, I decided to make multivariate equation with features got from given dataset. To get numbers in equation, I first sliced single paragraph into separate sentences. Then I chunked all sentences using ne_chunk, resulting tagged sentences with noun phrase trees. For implementation ease, I only thought of first part of the name. For "Bob Suter" in above example, I thought of it as "Bob". Then for each nouns A and B, I find 5 variables each. Thinking as a noun phrase as a single noun, 5 variables for noun A are like followings. 1. number of nouns prior to noun A in paragraph, 2. number of nouns prior to noun A in sentence, 3. number of times that noun A was used in paragraph, 4. position of preposition, 5. position of word. For page-context case, I used number 3, number of times that noun A was used in paragraph to number of times that noun A was used in the website. So we have 5 variables for two nouns A and B, resulting 10 variables in total. So thinking of equation a1*x1 + a2*x2 + … + a10*x10. We expect the result to be like following. If A, B is False False then 1, True False then 0, and False True then -1. Then using simple machine learning technique, multivariate linear regression, I implemented gradient descent method to find appropriate constants, a, to make adequate function. Then I got two sets of ten constants, which each will be applied to snippet and page case.

With set of constants made from development.tsv and changed constant based on result got from gap-test.tsv, I ran the code with validation and made two files CSV372_HW5_page_output_20160632 and CSV372_HW5_snippet_output_20160632 which

uses url and does not as shown on their names. I got the result and ran the test using gap_scorer and below figure is their precision. Thinking as a base accuracy as 33.3%, which is ideal percentage of random guess on three cases, it has low recall, high precision, and adequate f1. In overall, my model fitted better with Masculine than Feminine, though their accuracy rating are low in overall.

```
>>> Overall recall: 33.2 precision: 44.2 f1: 37.9
                tp 130  fp 164
                fn 262  tn 352
Masculine recall: 36.2 precision: 44.2 f1: 39.8
                tp 68   fp 86
                fn 120  tn 180
Feminine recall: 30.4 precision: 44.3 f1: 36.0
                tp 62   fp 78
                fn 142  tn 172
Bias (F/M): 0.91
```

By observing the outputs created by my constants, I concluded that this model just puts out something similar to random guess. To improve the result, I might need to make advancements in several parts. First, choosing more appropriate variables. Second, use more complex function for learning rather than simple linear regression. Currently I only choose my numbers based on nouns. It means that in two sentences like *"Mike did something."* and *"Though Mike did something to help, Duke failed."*, the word Mike is though to be same since it is the first noun in the sentence and also probably. So my approach to string, finding nth noun in sentence and paragraph can't be constants that can represent coreferent name, at least in their linear relations. Maybe we can just change nth noun in sentence to subject or object flag to be added. Second, I used linear regression model because it was the only learning model that I could implement without external models to run regression. I think this linear relation wasn't complex enough to represent relationship.

At the beginning I wanted to check how the result comes out using simple machine learning instead of using other approaches. However, the result I got was too simple to match the pattern of words.