Ariana Giorgi

ang2144

Assignment 3 - Python/OpenCalais

|  |  |  |  |
| --- | --- | --- | --- |
|  | # Entity references | Detected | Correct |
| stdin\_1 | 10 | 3 | 5 |
| stdin\_2 | 6 | 0 | 5 |
| stdin\_3 | 25 | 7 | 22 |
| stdin\_4 | 45 | 11 | 42 |
| stdin\_5 | 7 | 1 | 6 |
| Total | 93 | 22 | 80 |

All articles were selected at random from the homepage of Businessweek.com

Document 1 failures: imported beverage, wonderful crossover product, spokesman at the advertising agency, company spokesman, Chris Barnes

2: Karen L. Cates

3: Thierry Leyne, City hotel (from New York City hotel), Vanessa Strauss-Kahn

4: Dekkers Davidson, Linda Walsh, Deepika Pandey

5: Lucas Menge

The OpenCalais results were most accurate with companies like “Apple” and even “Dr. Pepper Snapple” and locations like “Mexico” and “New York”. It seemed that the links failed the most when the entity was a person, but not well known. Most of the people that it tried to create links for were the authors of the article or minor characters in the news story. Document 1 presented the most interesting failures because some of the entities classified are just nouns with adjectives like “imported beverage” or “wonderful crossover product”. I could see why “spokesman at the advertising agency” was identified because, in the demo text, “President of the United States” was identified and the structure is similar. It would be rare that our Wikipedia linking would work for this type of entity structure (with the exception of a well known person, like President of the United States).

Putting the dictionary into alphabetical order:

When I was testing the demo text (“George Bush was the President of the United States of America until 2009. Barack Obama is the new President of the United States now.”) Two of the entities found were “President” and “President of the United States of America”. I was using a replace technique with the links in the order that was collected in the dictionary (a python dictionary is “unordered” by definition). So if I replaced “President” *after* I replaced “President of the United States of America”, I would be placing a link *inside* of another link. I would get something like this:

<a href=…> <a href=…>President</a> of the United States of America </a>

It’s especially a problem in this simplified code because if our Wikipedia article link *contains* the second entity name, then the link becomes messed up because it tries to replace the word “president” in “www.wikipedia.com/wiki/president\_of\_the\_united\_states\_of\_america” with another Wikipedia link. Therefore the HTML becomes messed up.

To solve this problem in this experiment, putting the dictionary in alphabetical order allows “President” to be replaced before “President of the United States”. But because “president” is already replaced like so:

<a href=> President </a> of the United States of America

when the code looks for the sequence “President of the United States of America,” it is not found. This prevents links from being embedded inside each other, though sacrifices the number of links that are returned in the output. For the documents analyzed in this assignment, this didn’t seem to come up - it only came up in the demo test. For a more complex linking technique though, it would have to be decided how to handle this in order to return all links applicable to a phrase.