INFO 159/259 Homework 7
Coreference Resolution

November 14, 2017; due November 30, 2017 (11:59pm)

In this homework, you will be solving the problem of pronominal coreference resolution. You will try to answer the following question: *given a set of candidate mentions in a document, which is a correct antecedent for each pronoun?* You'll find the following files on bCourses in `Files/HW7`:

- `hw7.py`: A file with starter code
- `train.coref.data.txt`: training dataset
- `dev.coref.data.txt`: a dataset to help evaluate your model's performance
- `test.coref.data.txt`: the test dataset for us to evaluate your model's performance

Your training data will look like this:

document_id	**sentence_word_id**	**word**	**pos**	**mention_id**	**entity_id**
0910	0	Humphrey	NNP	1	0910:9
0910	1	Bogart	NNP	1	0910:9
0910	2	is	VBZ		
0910	3	married	VBN		
0910	4	to	TO		
0910	5	Lauren	NNP	2	0910:10
0910	6	Bacall	NNP	2	0910:10
0910	7	.	.		
0910	0	Bogart	NNP	3	0910:9
0910	1	starred	VBD		
0910	2	in	IN		
0910	3	the	DT	4	0910:14
0910	4	Maltese	JJ	4	0910:14
0910	5	Falcon	NN	4	0910:14
0910	6	.	.		
0910	7	He	**PRP**	5	0910:9
0910	8	also	RB		
0910	9	starred	VBD		
0910	10	in	IN		
0910	11	Casablanca	NP	6	0910:18

The training and development data sets will be in the format above. In the test data set, the **entity_id** column will be missing.

The **mention_id** column here specifies the identity of each unique mention in the document (here, [Humphrey Bogart] is mention 1, while the second [Bogart] is a separate mention 3). In the training data, **entity_id** specifies the entity in the real world to which those mentions refer; coreferent mentions are those that point to the same entity_id. Here, mention_id 1, 3 and 5 are all listed as referring to entity 0910:9 and are hence coreferent with each other.

Your task will be to identify the correct antecedent for each pronoun with a mention_id —i.e., every word for which the POS tag is `PRP` or `PRP$`, and is marked as a valid mention. You will submit your predictions for the test data: for each pronoun, select the single best mention_id it corefers with within the same document. Note that several possible antecedent mentions might be correct solutions; here, for example, selecting 1 ([Humphrey Bogart] as a coreferent mention of “He” is correct, as is selecting 3 ([Bogart]); all refer to the same entity_id. (Again, you’ll know the true entity_id for training and development data, but not for test data.) The evaluation function `check_accuracy` will accept as correct any (or multiple) `mention_ids` as long as they map to the correct `entity_id`.

For the test file you submit, you should replace the mention_id for the pronoun with your prediction for the correct antecedent mention. A valid response (in which the original mention_id for “He” is replaced with “3” as a choice denoting “Bogart” as the antecedent) could look like this:

document_id	sentence_word_id	word	pos	mention_id
0910	0	Humphrey	NNP	1
0910	1	Bogart	NNP	1
0910	2	is	VBZ	
0910	3	married	VBN	
0910	4	to	TO	
0910	5	Lauren	NNP	2
0910	6	Bacall	NNP	2
0910	7	.	.	
0910	0	Bogart	NNP	3
0910	1	starred	VBD	
0910	2	in	IN	
0910	3	the	DT	4
0910	4	Maltese	JJ	4
0910	5	Falcon	NN	4
0910	6	.	.	
0910	7	He	**PRP**	**3**
0910	8	also	RB	
0910	9	starred	VBD	
0910	10	in	IN	
0910	11	Casablanca	NP	6

In this assignment, you are free to use any resources or libraries that you wish, except those that explicitly perform coreference resolution (`spacy` in particular is a good python library for NLP: `https://spacy.io`).

1 Improving the baseline (2 pt)

a.) We have implemented the `naive_resolver` function. This function predicts that every pronoun refers to the antecedent that is closest to it. This very naive implementation achieves an accuracy of ~0.69. The first part of the assignment is to come up with a method that gains a higher accuracy than this method on the **development** set. Check your accuracy with the `check_accuracy` function (1 pt).

b.) After creating your model using the training and development data (use the dev data to guard against overfitting), make your predictions on the test set. Your submission is a text file `test_predictions.txt` with a similar structure to `test.coref.data.txt`, except with pronoun `mention_ids` changed to map to the appropriate `mention_id` of the predicted antecedent. We will test this submission and check if it does better than the naive implementation on the **test** data. A model that makes an overall improvement of at least 8% to the baseline will get full credit. (1 pt)

2 Model Comparison (1 pt)

In this section, we ask you to take a step back and see what the common mistakes your model is making. Propose an improvement to your model which tries to solve the identified weakness (and remember, just adding a feature or a pre-processing step makes it a new "model"). Evaluate both models against the development data set and compare and contrast their relative strengths and weaknesses. Does correcting for the mistakes from the first model result in a loss of accuracy for other examples? Print out 10 sentences that each model gets wrong and analyze the mistakes for any insight. Write up your results and your interpretation in a file `writeup.pdf`.

3 Extra Credit (1 pt)

Take your most robust and accurate model and provide predictions on the test data set. Starting Sunday, November 26, we will accept one submission of your `test_predictions.txt` file each day (from bcourses) and provide a score on how it fares on a part of the test set. You will also see how others fared on the same part of the data set. This would be similar to a Kaggle competition with a manually maintained leaderboard. After the deadline, we will look at the accuracy of the model relative to the rest of the class. The top 10% of submissions will get 1 point, top 25% will get 0.75 points and top 50% will get 0.5 points.

4 Deliverables

Submit `writeup.pdf, hw7.py, test_predictions.txt` on bCourses.