CS224N PA3 Report

Team Members: John Gold, Justin Salloum
SUIDs: johngold, jsalloum

Naive Baselines

One Cluster

In this system every mention was assigned to the same entity. This system forgoes precision and aims for high recall, since it marks every single mention as being coreferent with one another.

All Singleton

This system assigns each mention to its own cluster. By marking every single mention as a singleton, the system aims for high precision and forgoes recall.

Better Baseline

The naive baselines are two very extreme methods of implementing a coreference system and the default baseline also does a very poor job. The baseline coreference system marks two mentions as coreferent if their text matches exactly. Without even needing training data, this system was improved significantly by examining the heads of mentions rather than the entire mentions themselves. If the heads of two mentions were identical then the two mentions were marked as coreferent. For example, even though the two mentions "President Obama" and "Barack Obama" are different, their head word is both Obama and thus our better baseline system marked them as referring to the same entity.

Rule-Based

Our rule-based coreference system harnessed a variety of heuristics to determine if mentions were coreferent. We made multiple passes over the data, starting off with high precision and low recall rules and moving slowly to rules with lower precision and higher recall:

1. Exact match - the two mentions had the exact same text
2. Exact head match - the mentions' heads were identical
3. Lemma match - the mentions had the same lemma
4. Pronominal anaphora resolution - if the two mentions were in the same sentence, their head words had the same gender, and the first one wasn't a pronoun and the second one was, then we marked them as coreferent
5. Relaxed head match - one mention's head word was a substring of the other mention's head word
6. Pronoun similarity - we treated the following sets of pronouns as being the same so that if the two mentions' lemmas were both pronouns and both fell into one of these sets, then they were marked as coreferent:
    1. i, me, my, myself, mine
    2. we, us, our, ours, ourself, ourselves
    3. you, your, yours, yourself, yourselves
    4. he, him, his, himself

5. she, her, hers, herself
6. they, them, themselves, theirselves

We maintained a HashMap<Integer, Integer> to keep track of which mentions were coreferent with which mentions, where both the key and the value represented indices into the document's list of mentions. If a mention was to be marked as a singleton then its value in the map was -1. For example, the pair {3, 2} means that mention 3 was coreferent with mention 2, and {5, -1} means that mention 5 was marked as a singleton. On our first pass we initialized our map by putting {key, value} pairs for every key from 0 to one less than the number of mentions, setting the values to be the index of another mention with the same text or -1 for the singleton case. On every pass afterwards, we updated the values for each key such that the mention satisfied the given rule. Finally at the very end, we constructed the List<ClusteredMention> by iterating over all the mentions and using the map to determine which mention to mark the given mention as coreferent with or if to mark it as a singleton.

Classifier-Based

We experimented with many different features in the Classifier Based system, with varying degrees of success. We define success as an improvement to the baseline F1 score (by default MUC), starting from the given ExactMatch feature. Some features increased the F1 score, the majority had no effect, and a select few even made our classifier perform worse.

Below we list the different features along with a brief explanation of what they do and assign each an index. Farther down is a chart of that corresponds to features we tried in different subsets and the scores they achieved. The rightmost column corresponds to the weights of the feature added in new subset. If it's an indicator feature we write the weight if the indicator is true of false, if it is a pair or bucket we choose one or two indicative weights of why the feature may have helped or not.

1. Initial Exact Match: checks if the glosses of both mentions are the same
2. Bucket Distance: buckets the distance of mentions from each other
3. Head Match: checks if the head of both mentions are the same
4. Same NER: checks if the NER of both heads are the same
5. InSameSentence: checks if both mentions are in the same sentence
6. PrixInCluster: checks if the first mention is in the second cluster
7. Lemma Match: checks if the lemma of the head tokens are equal
8. NameAndPronoun: checks if one headWord is a name and one headWord is a pronoun
9. Pair of NER: pair of features that each contain headToken's NER
10. Pair of Gender: pair of features that each contain headToken's Gender
11. Pair of POS: pair of features that each contain headToken's POS

Features	Dev Set MUC F1	Dev Set B^3 F1	New Feature Weights
1	0.619368	0.590613	(false): -1.4, (true): .8

1,2	0.619368	0.590613	(0/11): .1, (10/11): -.2
1,2,3	0.688688	0.636136	(false): -1.2, (true): 1.0
1,2,3,4	0.688688	0.636136	(false): -.3, (true): 0
1,2,3,4,5	0.688688	0.636136	(false): -.5, (true): .4
1,3,5,7	0.741551	0.674784	(false): -1, (true): 1
1,3,5,7,8	0.741551	0.674784	(false): -.3, (true): .5
1,3,5,7,9	0.740054	0.672970	(change in scores are negative)
1,3,5,7,10	0.741551	0.674784	(M, M): .4, (F, F): .4
1,3,5,7,10,11	0.737803	0.672944	(change in scores are negative)

Figure 0: Incremental testing of classifier-based system on the dev set

Results

	MUC			B3		
	Precision	Recall	F1	Precision	Recall	F1
Rule-based	0.833	0.786	0.7994	0.766	0.699	0.7313
Classifier-based	0.811	0.683	0.7416	0.801	0.583	0.6748

Figure 1: Results for rule-based and classifier-based systems on the dev set

	MUC			B3		
	Precision	Recall	F1	Precision	Recall	F1
Rule-based	0.802	0.692	0.743	0.786	0.656	0.7153
Classifier-based	0.812	0.641	0.7164	0.837	0.602	0.7004

Figure 2: Results for rule-based and classifier-based systems on the test set

Discussion/Error Analysis

Rule-based

Exact matching and exact head matching achieved a very solid baseline F1 score, but failed to account for several cases that would be obvious to an English reader. There were many cases where the headwords themselves were different, but their underlying lemmas were the same, so we added lemma matching as our third pass. One of the most common sentence structures in the dev set was a mention of a subject followed by a pronoun reference to that same subject later on. For example, one of the sentences contained "...the Iraqi Red Crescent , for example , has suffered bombings and mass kidnappings , yet its volunteers…", where the mentions "the Iraqi Red Crescent" and "its" refer to the same thing. We also noticed that some head words were very similar even though they weren't exact matches. For instance, "bonfire" and "fire" weren't caught by our system, so we decided to relaxed head matching. Finally, there were many examples of sentences with multiple pronouns that were either all first person, second person or third person. For instance, in "...they made their rounds", the mentions "they" and "their" were coreferent but our system didn't initially pick them out.

To evaluate the effect that each rule had on our system, we tested our system on the dev set after each rule was added and monitored the F1 score. As we added on more and more rules, we observed that the improvements in the F1 score gradually became smaller and smaller. Our 2nd and 3rd rules took the MUC and B3 F1 scores all the way up to 0.743 and 0.676 from the baseline, and from there on each rule contributed to small increases. We also experimented with the order in which to perform our rules, to see which order yielded the best scores. We found that beyond the first three rules there wasn't much variability in the scores with regards to rule order.

Classifier-based

Based on Figure 0, we decided to go with features (1,3,5,7,10) for our final run on the test set. These are the ExactMatch, HeadMatch, InSameSentence, LemmaMatch, and GenderPair features. We tested many more subsets than were demonstrated in this chart, including additional features that weren't shown to keep it concise. A few of the takeaways we found are that trying to find ultra specific features (such as if both mentions are numeric) are too vague to have any effect. Additionally, the features that worked the best were more broad, such as the head and lemma match.

Future Improvements

Our pronoun anaphora resolution can be improved by enforcing number agreement and person agreement in addition to gender agreement, rather than just gender agreement. Very often there was a great deal of surrounding text between the pronoun and the earlier mention and while enforcing gender agreement improved our scores significantly, our system failed to pick out several examples that would easily be caught by gender and number agreement.

Another possible improvement is a more relaxed policy on head matching. At training time our system could collect data on the heads of mentions to predict if two heads indeed to correspond to the same entity; for example, "President Bush" and "George". Currently our system is not able to determine that these mentions are coreferent upon initial examination, but if our system makes use of the training data then this relaxed policy can help further improve our scores.

Conclusions

Coreference Resolution is a very hard problem, and once basic rules and features are implemented it is very difficult to obtain marginal improvements without implementing very complex models and algorithms.