CS224n Problem Assignment 3

Enhao Gong, Yuming Kuang

November 7, 2014

1. Baseline Systems

1.1 One Cluster

One Cluster system just simply make all the mentions refer to the same entity. In Table 1, it gets high F1 score in MUC, but low F1 score in B3. In more details, it gets 100 for recall because it makes all the possible links. It gets low precision in B3, because for each word, it's linked to all the other words and most of the links are wrong. However it gets high precision score in MUC, because the MUC precision score in this case is just (# of words − # of true entities)/(# of words − 1) and '# of true entities' is much smaller than '# of words'. In summary, generally larger clusters can give higher MUC score, but low B3 score.

1.2 All Singleton

All Singleton system just put every word a single entity. In Table 1, it gets 0 F1 score in MUC, but higher F1 score in B3 than One Cluster. In more details, it get 100% precision because it actually makes no links. It gets 0 recall in MUC, because it doesn't actually group anything together. However, it gets higher recall score in B3, because link to itself is correct and contribute to the recall score in this case.

1.3 Better Baseline: Head match

In the Better Baseline system, we implemented the head match. There are actually two senses of head match.

- Exact head match: Two mentions that share the exact same head are coreference.
- Head pair match: If the two heads of two mentions also appear in the training data as pair of heads of coreference, we also treated the two mentions as coreference.

To score head pair information learned from training data, we use

```
Map<String, Set<String>> headCluster
```

to store head and the set of heads which contains itself and heads of its coreference. In Table 1, our Better Baseline system got pretty good F1 score in both MUC and B3, reducing the precision a little but improving the recall a lot. This is reasonable since head match is a little less accurate than exact match, but it introduces a lot more true coreferences.

2. Rule Based System

2.1 Overview

Our Rule Based system contains 3 passes. The first two passes are Exact Match and Head Match, dealing with general nouns. The third pass deals with pronouns, including that quoted pronouns and Hobbs algorithm. As the same time, in order to improve precision, we also do an agreement check for coreference candidates found by head match, quoted first speaker and Hobbs algorithm. The check includes gender agreement, number agreement and speaker agreement.

Table 1 shows the result of our Rule Based system. Compared to the Better Baseline, Rule Based system increases precision score a lot in MUC and dramatically in B3, while only loses a little in recall as compensation. Thus Rule Based system improves the F1 score in both MUC and B3. This is because Hobbs algorithm and quoted pronouns help the system make a better decision for pronouns than head match, which improves both precision and recall. Also, agreement check also help rule out wrong coreference to improve precision.

2.2 Rules, Motivations and Implementations

Rules in the two passes for general nouns:

- Exact Match: Similar to in baseline, make two mentions with exact string match to be coreferent.
- Head Match: Similar to in Better Baseline, make two mentions with same head and head pairs appeared in training data to be coreferent.

Rules in the third pass for pronouns:

- Quoted: If the pronoun is quoted and is first speaker, we match it with its speaker. This is quite reasonable since speaker will use first speaker pronouns to refer himself in his speech.
- Hobbs algorithm: For all the pronouns that don't have a coreferent yet, we use Hobbs algorithm to find coreferent candidates for them. All the candidates found in each step are stored in a queue *candidates*, and is checked for agreement whenever not empty. When a candidate pass the agreement check, we stop the Hobbs algorithm and propose the candidate as antecedent. Also we want to note two implementation details:
    - In order to quickly search for parent when trace up in the parse tree, and search for mention corresponding to a tree node, we construct two hashmap *parentMap* and *parseMap* to store (child, parent) node pair and (node, mention) pair at the beginning of $runCoreference$ for each document. Also we define a new $TreeNode$ class that stores not only $Tree < String >$ but also *sentenceIndex* and $beginWordIndex$, and use this class as key type for $parentMap$ and $parseMap$. Because $Tree < String >$ class's *equals* method may treat two tree nodes at different places as equal, by storing sentence and word information $TreeNode$ can make tree node unique thus suitable as key type. $TreeNode$ also replicates the children list, so that we can directly use $TreeNode$ to do traversal.
    - When Hobbs algorithm fails to propose a suitable candidate from the same sentence, it will turn to previous sentence. Here we set a threshold $MAX\_REVIOUS\_SENTENCE$ as maximal number of previous sentences to search. So far, it is set to 1 from tuning.

Rules for agreement check:

- Gender agreement: Two mentions have to be of the same gender as coreferent.
- Number agreement: Two mentions have to be of the same number type as coreferent.
- Speaker agreement: Two pronouns have to be of the same speaker type as coreferent.

Rules **tried** but not helpful:

- Appositive: If two nouns are the 1st and 3rd children of the same NP node in the parse tree, and the 2nd child has label ',', then they are treated as appositive coreferent. We implemented this pass, but didn't find any appositive case. By taking a look at the data, we found this case has already been dealt with, for example, 'the first business operated by the peasants' association after it was registered, the Zhaizi Village Paint Plant' is already put together as one mention.
- NER agreement: We also tried adding the NER agreement that two mentions have to be of the same NER type (if they have) to be coreferent. However we found that adding this constraint actually lower the F1 score. It was possibly because the NER tags are not accurate enough. So in our final system, NER agreement is not included.

3. Classifier Based System

3.1 Overview

For the classifier based system we basically need to develop features, including defining new indicator classes, implementing the feature evaluation (for fixed and/or candidate mentions) as well as putting new features into the list of Active Features. We implemented 3 additional features beyond the recommended ones.

3.2 Detail features

Motivated by RuleBased system, here we used mainly the following 10 features **(3 beyond the recommended ones)** which intuitively highly related to coreferencing:

- Exact Match (as in baseline): class Feature.ExactMatch
- Head Match (as in better baseline): class Feature.HeadMatch
- Gender agreement (as in RuleBased): class Feature.GenderMatch
- Speaker agreement (as in RuleBased): class Feature.SpeakerMatch
- Number agreement (as in RuleBased): class Feature.NumberMatch
- Distance between the mentions (in terms of other mentions): class Feature.MentionDistance
- Distance between the mentions (in terms of number of sentences): class Feature.SentenceDistance
- Paths through the sentence parse (if same sentence): class Feature.Path
- Whether the fixed or candidate mention is a pronoun: Pair.make(Feature.CandidatePron.class, Feature.FixedPron.class)
- The named entity type for both fixed mention and the candidate mention: Pair.make(Feature.FixedName.class, Feature.CandidateName.class)

3.3 Feature Tuning

Table below shows the detail results of the classifier based system with different feature in tuning with *dev* data.

	dev data					
Feature Ellimination	MUC			B3		
Leave-one-out	Precision	Recall	F1	Precision	Recall	F1
Use all features	**0.855**	**0.729**	**0.787**	**0.812**	**0.657**	**0.726**
w\o ExactMatch	-	-0.002	-0.001	+0.001	-0.002	-0.001
w\o HeadMatch	+0.004	**-0.091**	**-0.060**	+0.036	**-0.085**	**-0.043**
w\o GenderMatch	+0.001	+0.001	+0.001	+0.001	+0.001	+0.001
w\o SpeakerMatch	-0.002	+0.003	+0.002	-0.001	+0.002	+0.001
w\o NumberMatch	+0.001	+0.001	+0.001	+0.001	+0.001	+0.001
w\o MentionDistance	-0.006	**-0.012**	**-0.010**	-	**-0.016**	**-0.009**
w\o SentenceDistance	+0.001	**-0.008**	**-0.004**	+0.011	**-0.018**	**-0.006**
w\o Path	**-0.013**	**-0.013**	**-0.013**	+0.001	**-0.019**	**-0.011**
w\o Pronoun check	**-0.034**	**-0.116**	**-0.085**	+0.020	**-0.124**	**-0.076**
w\o Name entity	-	-0.002	-0.001	+0.001	-0.002	-0.001

So the performance improvement is mainly because of using: ***HeadMatch, Mention Distance, Sentence Distance, mention Path*** and ***Pronoun check.***

All of these 5 features improve the overall F1 and Recall significantly, while Path and Proune check also have great improvement for Precision.

3.4 Comparison

Compared with BetterBaseline and RuleBased system, classifier consistently results in better Precision, worse recall but better F1 score in most cases. The improvement in precision is bacause that a much more sophisticated classifier is trained using the features including but more than only using binary criteria based on head match, agreement checks and pronoun identity. The reduce of recall is bacause the classification errors and tradeoff of prudent coreferencing which increases overall F1.

4. Result Table

Our results are better than the reference solution (MUC F1 of 0.701 and a B3 F1 of 0.685) by about 0.05 F1 score in test data.

5. Error Analysis

	dev data					
	MUC			B3		
	Precision	Recall	F1	Precision	Recall	F1
OneCluster	0.769	1.0	0.869	0.164	1.0	0.282
AllSingleton	1.0	0.0	0.0	1.0	0.246	0.395
Baseline	0.805	0.503	0.619	0.862	0.448	0.590
BetterBaseline	0.765	**0.778**	0.772	0.541	**0.708**	0.613
RuleBased	0.814	0.745	0.778	0.739	0.665	0.700
ClassifierBased	**0.855**	0.729	**0.787**	**0.812**	0.657	**0.726**
	test data					
	MUC			B3		
	Precision	Recall	F1	Precision	Recall	F1
OneCluster	0.743	1.0	0.853	0.126	1.0	0.225
AllSingleton	1.0	0.0	0.0	1.0	0.273	0.429
Baseline	0.807	0.394	0.530	**0.913**	0.435	0.589
BetterBaseline	0.762	0.736	0.749	0.617	0.707	0.659
RuleBased	0.808	**0.737**	**0.771**	0.738	**0.709**	0.723
ClassifierBased	**0.840**	0.654	0.734	0.853	0.628	**0.723**

Table 1: Summary of Result

5.1 Rule Based System

- Although Hobbs algorithm can help better determine pronoun's coreferent, it can not cover all the cases. For example, 'to be a reason for holding {{others}} back , and to smother {{their}} abilities and capabilities', 'others' and 'their' are actually from the same SBAR node in the parse tree, but Hobbs algorithm won't stop at SBAR, thus it won't propose 'others' as antecedent of 'their' when traversal.
- By tuning, we pick the maximal previous number of sentences to search in Hobbs algorithm to be 1. But clearly there are some long distance coreferences that are not captured. For example, '{{My}} head is raised' appears two times but sentence index difference is 3, thus we can't recognize the two 'My' are coreferent.
- The number agreement check is not reliable in the case that one of the nouns can be either single or plural. For example, the system fails to recognize 'People' and 'Their' as coreferent in 'Buying {{People}} and Exploiting {{Their}} Needs', because it just doesn't treat 'People' as plural word.
- Head match for general nouns is limited in the case of semantic coreference. For example, in 'on the part of {{Hezbollah}} and {{its}} supporters and a confirmation of the strength of {{this organization}} .' the system is not able to know that 'Hezbollah' is an organization thus make coreference between 'Hezbollar' and 'this organization'.

5.2 Classifier Based System

- Classifier based result in better coreferencing. Here is a example of mistakes made that are really hard for the classifer: {{They}} try to recruit {{them}} from the areas of the south known for extreme poverty. The system failed to distinguish the two one pronuns since it is probably different from common cases in training data.
- Although path feature provide siginficant improvements but it also result in some false positive such as: remember {{people}} sniffing at {{it}}

6. Improvement Ideas

- A more sophisticated algorithm than Hobbs algorithm can help deal with pronoun coreference, maybe combine with machine learning methods. Also a Hobbs-like algorithm for general nouns is also helpful to determine semantic coreference.
- A better use of lexicon information can help in both semantic coreference and agreement check.
- Additional features can be added to address semantics and web appearnances.