CS224N Programming Assignment 3
Assignment Report

Daniel Chia <danchia@stanford.edu>
Frank Chen <frankchn@stanford.edu>

We have used 2 late days on this assignment. Submitted 11:00am Friday Nov 9th 2012.

Introduction

In Programming Assignment 3, we are asked to implement in coreference systems based on both rule-based deterministic linguistics rules and a MaxEnt Classifier based machine learning model. We provide basic descriptions our methods, implementation details and testing, results and a general discussion of our methods.

Frank wrote the AllSingleton and OneCluster implementation and implemented the BetterBaseline Coreference System. Frank also wrote and tuned the rule-based coreference resolver, while Daniel wrote the MaxEnt-classifier based coreference resolver and Hobbs Algorithm. Daniel and Frank analyzed the results and wrote this report together.

Overview of Methods

Baseline, AllSingleton and OneCluster

The Baseline coreference system provided is very simple, where two mentions are coreferent if and only if they have the same string form. The AllSingleton coreference system is very simple as well, as it classifies all mentions of strings to be different, regardless of the textual content of the string. The OneCluster coreference system is extremely simple, as it marks every single mention of a reference as part of the same cluster. These systems are used to provide a baseline and help us understand the results of the training and evaluation.

BetterBaseline

The BetterBaseline system we devised marks pronouns with the same gender as coreferent with each other. In addition, we also collect common coreferences from all documents during the training phase and mark mentions that we have seen before in training as coreferent as well. All other mentions that are unclustered at the end of those two phases are marked as singletons.

Rule-based

The BetterBaseline system performs about average but cannot handle pronouns or pronoun-phrase mentions of entities. Our rule-based system will attempt to use common linguistics rules and methods to overcome the deficiencies of the baseline system.

We designed the system to apply rules first with we determined to be high precision, low recall rules so that we capture the precision, while general lower-precision but higher-recall rules came later. Finally, we marked all elements that are not classified by any of our rules as Singletons.

We developed the following set of rules, some of which we found decreased performance in combination with other rules and thus were not used.

- Exact String Matching -- We compared the exact english phrase of the mention. If they are the same, then we cluster the two mentions into a single set.

- Head Word Matching -- We compared the head words in across all pairs of mentions in clusters, and if any pair of head words match, we merge the sets of mentions.

- Inclusion Testing (not used in final product) -- We determine whether one phrase is a substring of another phrase. If they are, we merge the two mentions.

- Noun-Pronoun Comparison Testing -- We tried a simple rule-based noun-to-pronoun matching system where we compared attributes and tagged them as referring to the same entity if every single pronoun-noun comparison passed our sanity checks (including having the same plurality, same attributes, etc...)

- Speaker Testing -- We detect whether the mention is being used in a quote and if the mention in the quote is a pronoun in the first person. If it is, we mark the speaker mention with the pronoun mention within the quotes.

- Similar Modifier Tests -- We retrieve all noun phrases in the pairs of mentions and if the noun phrases are the same, then they are the same with high probability.

- Hobbs Algorithm -- We implemented the standard Hobbs algorithm and used it to determine noun-pronoun coreferent mentions.

- *Common Coreference Testing -- We also re-implemented in the single coreference system in the Baseline model to compare and further improve the system on our rule-based model. We recognize that this is not true rule-based coreference testing, and thus have decided to separate our results with and without this.

MaxEnt Classifier-based

The MaxEnt Classifier revolves around simple indicator features:

- Case-insensitive headword match. The headword of the two mentions either match/don't match.
- Number agreement. True if both mentions have a number and match.
- Gender agreement. True if both mentions have a gender and match, or if either of them are dual gender.
- Strict pronoun gender agreement. Only applies when both mentions are pronouns, and is almost the same as gender agreement, except if a pronoun has neutral gender (like "it"), it should not match a gendered pronoun (like "I").
- Possessive pronoun indicator. Indicates whether a mention is a possessive form of a pronoun.

- Reflexive pronoun indicator. Indicates whether a mention is a reflexive form of a pronoun.
- Sentence distance. Distances in sentences between the two mentions.
- Mention text. Combined text of both mentions separated by a separator.
- POS tag pair. Combined POS tag of both mentions.
- Mention i pronoun indicator, and Hobbs distance. Pair feature, indicating whether mention i is a pronoun (and thus Hobbs algorithm applies), and the Hobbs distance between the two mentions.
- NER tag of mention j, and text of mention i. This pair feature captures what words typically refer to a particular NER type.
- Either mention has a pronoun, and number agreement. This pair feature combines the number agreement feature, with an indicator of whether either mention has a pronoun. Number agreement is more critical when at least one mention has a pronoun, as pronouns often encode number, thus this pair feature allows the classifier to weight such cases more heavily.

Implementation Details

Rule-Based

We decided to group the mentions into a Set<Set<Mention>>, and we initialize each individual mentions into individual sets containing one mention each. We then run each rule in sequence, from the highest precision and lowest recall rules (e.g. Exact String Matching) to more general rules with lower precision and higher recall. Whenever a rule has determined that two sets of mentions should be merged, we add the set into the

After all the rules are finished and sets of mentions have merged, we mark all Mentions in each of the sets that are left as co-referent with each other. We found this way to be much easier than using the default ClusteredMentions class and it makes our code much more straightforward and easier to program for. In turn, it allowed us to iterate and try new rules much faster.

Classifier-Based

Implementation revolves mainly around implementing features for the classifier. What helped was learning that essentially, the output of each <feature class, feature value> essentially becomes a binary indicator with a weight learnt by the classifier. This helped inform the design of features, such as how to indicate a “not applicable” response, which was done either by using an IntIndicator and reserving values for true, false, and not applicable, or by using StringIndicator and directly coding these responses.

Baseline Systems Testing

Single Entity and All Singletons

As suggested by the assignment handout, we coded up the Single Entity and All Singleton classifiers, where every mention is either associated with one single entity or all mentions are given their own entities, respectively.

Classifier	Test	Precision	Recall	F1
Single Entity	MUC	0.743	1.000	0.853

Single Entity	B3	0.126	1.000	0.225
All Singleton	MUC	1.000	0.000	0.000
All Singleton	B3	1.000	0.275	0.431

We can see that both these methods are bad at determining any sort of coreference. However, we can also observe peculiarities with the MUC scores on both datasets. In the Single Entity classifier, we have the MUCH score having a precision of 0.74 and a recall of 1.0, while in the All Singleton classifier, our recall drops to 0 while our precision rises to 1.0.

These results are to be expected: in the All Singleton test, we never classify any pair of mentions as coreferent, so we never produce erroneous results, causing our precision to go to 1. However, in the Single Entity test, our recall goes to 1 as we manage to classify correctly identify every single pair of actual coreferent mentions are coreferent as we mark every single pair of mentions as coreferent.

From these testing, we know that B^3 is a much better measure subject to less edge cases than MUC, where the F1 score in question could be abnormally high.

BetterBaseline

The BetterBaseline system with basic coreference training and standard pronoun matching did relatively well in our tests for such a simple system. We found during testing that the simple coreference training is particularly effective, indicating to us that our data contained repeated mentions.

Classifier	Test	Precision	Recall	F1
BetterBaseline	MUC	0.761	0.467	0.578
BetterBaseline	B3	0.827	0.486	0.612

Rule-based / MaxEnt Classifier Testing

We tried multiple rules (see our Methods section) and combinations of these rules in order to find the best results. We ran all our development and testing on the development set. The final results are reported using the test set as required by the assignment.

We are using B^3 to measure our results, with the reasons stated in our analysis of the results from SingleEntity and AllSingleton tests, as B3 is less suspectible to edge cases and is a much better measure.

Rule-based

Rules	B3 Precision	B3 Recall	B3 F1
Training	0.943	0.497	0.650
Dev	0.947	0.478	0.635
Test	*0.921*	*0.590*	*0.720*

MaxEnt Classifier-based

Dataset	B3 Precision	B3 Recall	B3 F1
Training	0.901	0.726	0.804
Dev	0.801	0.698	0.746
Test	*0.864*	*0.705*	*0.776*

Discussions

Rule-based System

We investigated the debug information for the Rule-based System extensively, and identified instances where it did well and instances where it could use improvement. We also suggested improvements in the following section of our code.

Proper Nouns: As we expected, our Rule-based System performs very well when resolving proper nouns. Our system managed to find and allocate most of them to a single entity correctly, using in some cases pure string matching. We found that the exact string matching test we use as our first pass found a majority of the cases below.

```
{God} -->  {God}; {God}; {God 's}; {God 's}; {God}; !{he}; !{him};
{Allah} -->  {Allah}; {Allah};
{Saudi Arabia} -->  {Saudi Arabia};
```

Rephrased Proper Nouns: The Head Words test where we only compare the head words of each mention rather than the exact string performed very well in phrases like the ones below. In fact, incorporating these mentions in our system as part of the Head Words rule gave us the largest improvement in F1 score (from 0.615 -> 0.717), indicating that a large number of mentions are rephrased proper nouns.

```
{George W. Bush} -->  {George W. Bush}; {Governor Bush}; {Bush}; {Bush};
{ABC News} -->  {ABC 's}; {ABC News}; {ABC News};
```

Headword Filtering: The headword matching rule was generally adept at matching proper nouns especially when there is minor rephrasing of the entity in different locations, but a key word, such as “strike” or “tank” is maintained in all the mentions.

```
{strikes} -->  {these strikes}; {the strikes};
{the village water tank} -->  {the tank's};
```

Hobbs Algorithm: Hobbs algorithm was able to match pronouns to entities without significant loss of precision, indicating a low number of false positives. Indeed, manual inspection revealed this to be the case, with Hobbs’ algorithm successfully matched pronouns to entities where other algorithms would have failed badly.

```
{a suspect who has been in custody for several weeks} -->  {his};
{FBI agents and divers} -->  !{they};
```

Areas for Improvement:

We have noted a couple of areas for improvement by analyzing our errors during coreference resolution.

1. *Synonyms* -- We can construct a database of synonyms for proper nouns. For instance, “register” and “account” might mean the same thing in certain contexts, but our system has no way of determining that at the moment.

2. *Items and Specific Names* -- Our system could benefit from a database of proper nouns to categories as well. For instance, the current system will be unable to identify “USS Cole” as a “ship” even though they refer to the same thing.

3. *NP Filtering* -- During headword detection, we want to detect words in noun phrases such as “every”, “all”, “no”, “some” or “each”, and reject them as being coreferent, as they refer to classes and negations of mentions, and thus should not be coreferent.

4. *Scoping Rules* -- More sophisticated scoping rules should be introduced in our system. For instance, genderless pronouns (it, they, etc...) will have a limited scope as, in general, speakers will tend to move on to something else and the meanings will be revised. Gendered pronouns will tend to have a longer life. Proper nouns usually have a very wide scope. Our current system assumes all nouns will have a wide scope, which is not true in English text.

   For instance, we have, in our development set classified the mention “people” to various other references, including other mentions of “people” in other parts of the text: “{people} --> ! {People}; !{its}; !{they}; !{them}; !{they}; !{them}; !{they}; !{they};”. With scoping rules, we will be able to limit

5. *Conjunction Processing* -- We want to identify conjunctions and eliminate them from head words processing. For instance, we have classified “Iran”, a country, to be “Iran” and “Iran and Syria”. Obviously, the second mention does not refer to the same entity, and we would not have committed this mistake if we detected the conjunction before attempting to compare head words.

Max-Ent Classifier System

Development of features for the classifier was guided by B^3 scores, looking at common mistakes and occasionally the model weights as well.

Exact match: High precision, but very low recall.
B^3 Precision: 0.8914532490449796
B^3 Recall: 0.4157685200816629
B^3 F1: 0.5670624630510022

Case insensitive headword match: Looking through the mistakes, exact match missing out a lot of links because the mentions have auxiliary words. In addition, given the extremely low recall of the system, it makes sense to relax the constraint as a first cut and have a more general feature. Further observation

shows some mistakes are merely capitalized vs non-capitalized words, often because of position in sentence rather than semantics, so use case-insensitive compare instead. Recall improved at cost of some precision.
B^3 Precision: 0.8564227424030268
B^3 Recall: 0.5301627053059763
B^3 F1: 0.6549086444663743

+Gender Agreement, Number Agreement, Possessive Pronoun Indicator, Reflexive Pronoun Indicator: Looking at mistakes, it's clear that the algorithm is still missing a lot of pronoun-antecedent links. Based on the classifier weights, this is not surprising, as the lack of a headword match turns out to be a very strong negative indicator. Unfortunately, pronoun-antecedent matches typically involve headword mismatch. Thus, we introduce a set of features related to pronoun-antecedent matching to try and give the classifier more to work with.
B^3 Precision: 0.8470242117758936
B^3 Recall: 0.558145815593716
B^3 F1: 0.6728908392591654

+Has Pronoun, Number Agreement: Looking at classifier weights, the number agreement feature isn't being assigned a very high weight, although this should be an important feature. This is possibly because it's more important when there is at least one pronoun present, so this pair provides this conjunction of information to the classifier. Very slight increase in performance.
B^3 Precision: 0.8380364347158274
B^3 Recall: 0.5626793030790107
B^3 F1: 0.6732925808102652

+Mention i is a pronoun, Hobbs Distance: The classifier is still missing a lot of pronoun-antecedent links. To provide the classifier with more information about probable guesses, the Hobbs distance is a good candidate. Pairing it with an indicator of whether mention i is a pronoun helps the classifier ignore cases when mention i is not a pronoun.
Looking at mistakes, the classifier is now doing somewhat better. For example, one of the examples involves coreferences of "Qingqing" to "he"/"I"/"himself"/"me". These were completely missed earlier, but with Hobbs Distance the classifier is now able to pick up more of these.
B^3 Precision: 0.8102611912852874
B^3 Recall: 0.6007683039097816
B^3 F1: 0.6899632406976525

+NER Tag of Candidate, text of onPrix: To further help the classifier with pronoun-antecedent links, another useful feature is figuring out what words typically go with an entity type. For example,
{other people} --> {their};
the classifier is now able to classify this correctly, as it has additional information that their is usually associated with a PERSON NER tag.
B^3 Precision: 0.837056263350753
B^3 Recall: 0.6309953591226897
B^3 F1: 0.7195640935418944

+POS Pair: This feature helps to encode that certain certain combinations of POS are more likely to appear, and others not so. This is done by coding the POS tags of both mention headwords. Looking at the mistakes, this help provide the classifier with more evidence about prounoun - noun linkage, like in {some local government officials} --> {they}; {they}; !{they}; !{they}; !{they};

Earlier, the classifier has not managed to match any of the they's. However, more false positives have started to appear. Overall though, performance still increases as recall is increased a fair bit.
B^3 Precision: 0.7687116398568492
B^3 Recall: 0.6698627955007658
B^3 F1: 0.71589111463736

+Raw mention text pair: In machine learning it's often a good idea to try included the raw data as one of the features, which is exactly what this feature does. Intuitively, this can help the classifier resolve aliases indirectly from data, as well as commonly coreferent pronouns (such as I/my, we/our, us/we). This can be verified by looking at the weights the classifier learns - common and intuitive pairs of phrases are seen with high weights. A significant improvement in performance is seen, which is somewhat surprising.
B^3 Precision: 0.8917265547701728
B^3 Recall: 0.706053884639923
B^3 F1: 0.788102022659018

+Sentence distance: So far, with the except of Hobbs distance, there hasn't been much notion of distance between mentions. It is more likely that a mention is coreferent with something more recent, hence sentence distance is a useful feature.
B^3 Precision: 0.9024915930342449
B^3 Recall: 0.7244020065465214
B^3 F1: 0.8036994196225771

+Strict gender agreement: At some point, it was noticed that the classifier was trying to group either-gendered pronouns like "we" together with ungendered pronouns like "it". This is partly an artefact of the loose gender agreement comparision earlier - if either pronoun is marked as accepting either gender, then the check passes. Making sure that a neutral gender won't match an either gender type should provide a small performance increase, which it does (more so on the validation set).
B^3 Precision: 0.9013745124023964
B^3 Recall: 0.725717563935463
B^3 F1: 0.8040642872608429

<u>Common Mistakes / Areas for Improvement</u>

1. Understanding conjunctions. None of the features right now attempt to understand common conjuctions, which affect both number as well as the semantics of the entity itself. For example, the system wrongly coreferenced "Iran and Syria" with "Iran", as they both had the word "Iran," and also failed to link "them" as it probably thought "Iran and Syria" was singular.
   {Iran and Syria} --> !{Iran};
   By building features that attempt to understand simple conjunctions such as "and," this can help with such cases.
2. Understanding modifiers. Even though the head word might be the same, a modifier might change the entity being referred to by the noun.
   {last year} --> !{2006}; {the year}; !{next year};
   Here, "last year" and "next year" are not the same entity, but the system thinks it is as it treats the modifiers "last" and "next" as garbage words. One way to try to improve this would be to have a feature that matches modifiers, where a modifier is a word that comes immediately before a noun.
3. Lack of context. The features don't gather any context about the document, thus using the same example above,
   {last year} --> !{2006}; {the year}; !{next year};

it has no way of knowing that 2006 is “last year”. This is probably a limitation of the current framework, as it’s difficult to obtain general context about the document without making some sort of inference pass first before starting entity-mention classification.
Another example of this would be one development set test case, where the document was basically a piece of prose by Pang Dongsheng. The classifier missed out all pronouns such as “I”/”my”/”me” that were supposed to refer to Pang, because the only time his name appears is when he signed off at the end. Such long-reach references are difficult to resolve, and again would be easier with whole-document context.