Coreference Resolution

Michael Comstock

December 6, 2013

Abstract

This document examines coreference resolution. It addresses three area's in particular. The first area is scoring. The second is establishing non-coreferent entities through type, gender and number conflicts. The third and final area addressed is using context to aid coreference resolutions for mentions with multiple compatible reference entities.

1 Introduction

For each of the three sections given in this paper, I will be evaluating my system's performance as tested on the first 30 documents of Stanford CS224n's dev set for programming assignment 3. Unfortunately, due to the need for hand labeling the mentions with meta-data, the test set size needed to be small for this paper. Future work may benefit from a larger test size, however as no learning was used in this coreference system, a larger test size will likely only give more reliable statistics. There is no reason to assume performance will improve with larger amounts of data.

2 Scoring

Before delving into the details of the approach I took for setting up the coreference system I did, I am going to say a few words about the inadequacies of the scoring system. To do this I will look at the scores of two very simple systems. The first system is assume all mentions are coreferent. Under this scheme, one attains the scores shown in Table 1.

Table 1: MUC Results for all mentions coreferent

MUC Precision	0.7584480600750939
MUC Recall	1.0
MUC F1	0.8626334519572955

Such a score is very good and as such, the usefulness of this metric is called into question. A simple examination of the text reveals that the mentions are in fact not all coreferent. And additionally calling this guess "good" also seems a bit misrepresentative of reality.

By contrast if we examine the B^3 scores we see very different results. These results are shown in Table 2.

Table 2: B^3 Results for all mentions coreferent

B^3 Precision	0.11141047028923072
B^3 Recall	1.0
B^3 F1	0.2004848312437394

The results here look quite poor which seems to more closely match reality. Additionally if we take the opposite approach and say no two mentions are coreferent we get similarly weird results form MUC and again better results from B^3. These results are shown in Table 3.

Table 3: MUC & B^3 Results for no mentions coreferent

MUC Precision	1.0
MUC Recall	0.0
MUC F1	0.0
B^3 Precision	1.0
B^3 Recall	0.2546125461254612
B^3 F1	0.4058823529411764

Due to the very strong bias of MUC to encourage the developer to just make everything coreferent, I will focus on the B^3 score for the majority of this paper.

3 Distinguishing non-coreferent mentions via conflicts

3.1 Rationale

The first issue which I addressed when designing my system was this : Should all things be considered innately non-coreferent and then merged or should everything be innately correferent and then separated out. I opted for the second approach. All things are assumed coreferent until evidence is given to the contrary. The reason for this is best demonstrated by the following toy problem. Given the following text :

Jack lives next door. He is tall.

Is "he" coreferent with "Jack?" Most people would agree the answer to this question is yes. But why? Is there any evidence that "he" refers to "Jack." If there is it can only be that Jack is most commonly a male name and "he" refers to a male subject. But if this is sufficient for the two to be merged then the phrases

Jack is short. He is tall.

should also merge "Jack" and "he." This of course is clearly wrong and most people would not do this. Thus, I am left to conclude the approach which most closely replicates what people do is to merge all mentions until evidence is given indicating a conflict arises when we merge.

3.2 Implementation

In order to create conflicts between non-coreferent mentions, I labeled each mention with 3 pieces of meta data. These are

1. Type $\in$ {Person, Place, Thing, Event, Group, Place}

2. Gender ∈ {Male, Female, None}

3. Number ∈ {One, Many}

Note that mentions may be tagged with several members of each set. For example "they" could be either male or female.

The results of adding these tags are as follows :

Table 4: B^3 Results after adding conflicts

No Conflicts	
B^3 Precision	0.11141047028923072
B^3 Recall	1.0
B^3 F1	0.2004848312437394
Type Conflicts	
B^3 Precision	0.3251689115763865
B^3 Recall	0.8557870044901602
B^3 F1	0.47127132351924444
Gender Conflicts	
B^3 Precision	0.17915305581773422
B^3 Recall	0.8807426042431444
B^3 F1	0.29774200401946277
Number Conflicts	
B^3 Precision	0.16147940776041972
B^3 Recall	0.8867154222956417
B^3 F1	0.2732054712322375
Combined (All 3) Conflicts	
B^3 Precision	0.38603295968573736
B^3 Recall	0.8398235787740319
B^3 F1	0.5289355998139048
Additional Tweaks	
B^3 Precision	0.590249753023775
B^3 Recall	0.7190328012407494
B^3 F1	0.6483076276636904

Here the additional tweaks are adding a few rules such as

- If two proper nouns are both times, then they must be identical strings to be coreferent.
- If two proper nouns are people then at least one word from one of the mentions must be present in the other mention.

3.3 Analysis

The above results show improvement to the overall F1 score. However, they do not show improvement to both the recall and the precision scores. In fact the Recall score decreases with the addition of conflicts.

At first this is a bit surprising because it seems that if two mentions conflict, then separating them should not reduce your recall score. The problem arises however, when a new mention is encountered which can belong to multiple categories. This can be seen in the following example :

Entity	Mentions
Mr. Tim	{Mr. Tim, he, him}
Mrs. Sue	{Mrs. Sue, she, she}
?	the person

Here when all mentions are coreferent, "the person" will trivially be added to the set of all other mentions with which it is coreferent. Here, however, there is no conflict with adding "the person" to either group. Since, it can only be coreferent with one entity there is a chance (50% if we assign randomly) that we assign it to the wrong entity. In the next section I examine a few schemes for dealing with mentions which are coreferent with multiple entities.

Additionally it is good to note that there appears to be a trade-off between precision and recall. This makes sense, because the more groups you have (which can be necessary for higher precision) the more opportunities you have to mismatch ambiguous mentions.

A final important note about this implementation is it makes no use of lexical data to distinguish entities. For example, "the box" and "the ship" are both things, both singular and both neutral gender. Thus under the scheme implemented for this text, they would be classified as coreferent. The reasoning behind this is my goal here was to separate our groups which cannot be coreferent. The goal was not to find things that are most likely not

coreferent. This system tries to use meta-data to ensure correctness. It does not take advantage of the statistical fact that most words are not synonyms.

4 Selecting Matching Entities from Multiple Matches

4.1 Rationale

In the previous section, I mentioned that the reduction in Recall was due to mismatching mentions which were compatible with multiple entities. In this section I will describe 4 scenarios for matching. These schemes are as follows

- Random
- First Match
- Nearest Match to Mention
- Nearest to Previous Match

The reasoning behind this is as follows. The random match provides a good baseline for how well a matching scheme performs. If it does not perform better than random then it is clearly flawed. The first match is the most naive approach and provides a baseline for deterministic matching. The match nearest to mention approach takes from the philosophy that coreferent mentions tend to appear closely together in text. For example if you are talking about a box in one paragraph and the next four paragraphs are spent talking about a boat then a mention such as "it" most likely refers to the boat and not the box. The approach which matches a mention to the match nearest to the previous match, attempts to provide some context for mentions.

The driving motivation behind this last matching method is the following text.

```
And why (BLUE){Maureen Dowd} (CYAN){thinks} (MAGENTA){women of
her generation} were sold a bill of goods /.
(3) All the things (MAGENTA){we} did that (MAGENTA){we} thought
would make (MAGENTA){us} more fascinating like high - powered
careers /.
```

```
(4) And (MAGENTA){we} wanted (BLUE){that snappy Hepburn Tracy
dialogue} because actually a lot of guys find (BLUE){that}
draining /.
(5) We 'll talk to (BLUE){her} about (CYAN){that} and a whole
lot more /.
```

This text has two instances of the mention "that." Correctly, labelled, the first mention is coreferent with "that snappy Hepburn Tracy dialog." The second mention is coreferent with "thinks" which is a few sentences prior.

If we use a scheme where we find the nearest compatible mention to the one we are matching then in the case of the first "that" we get the correct pairing with "that snappy Hepburn Tracy Dialog." However, the second instance of "that" would also be matched to this mention which we know is incorrect.

I propose that the reason we know this second matching is incorrect is because the previous mention "she" is matched to "Maureen Dowd" and thus puts us in that context. Since context is not rigorously defined I was forced to provide a definition. The one I came up with is *near in the text*. This lead to the algorithm which says, since "she" is matched to "Maureen Dowd" and "that" is near "she" it should be coreferent with something as near to "Maureen Dowd" as possible. This implementation gives the correct matching in the case above.

4.2 Implementation

As stated in the previous section I implemented four matching schemes for mentions which were compatible with one or more previous mentions.

The results shown in Table 5 below are for matching schemes used with the combination of all 3 labels : TYPE, NUMBER and GENDER as well as the additional tweaks.

4.3 Analysis

The results of the first two implementations, random, and first match are quite expectedly poor and uninteresting. What is encouraging however is both the two other schemes show increases in both Precision and Recall over both these baseline schemes. This indicates that more mentions are correctly matched (rather than just more mentions are matched together overall).

Table 5: B^3 Results with various matching schemes

Random Matching	
B^3 Precision	0.559031
B^3 Recall	0.675984
B^3 F1	0.609453
First Match	
B^3 Precision	0.530198947043191
B^3 Recall	0.6750244908670264
B^3 F1	0.5939102460646387
Nearest Match To Mention	
B^3 Precision	0.5867825024461544
B^3 Recall	0.7170174884002429
B^3 F1	0.6453954887175916
Nearest Match to Previous Match	
B^3 Precision	0.5862118470986227
B^3 Recall	0.731038625174989
B^3 F1	0.6506636540043743

The most interesting comparison however is between the last two implementations. While the nearest to previous match does show some improvement over the nearest to mention scheme the improvement is slight. Encouragingly though, the improvement appears to boost Recall with only a very slight loss to precision indicating that this improvement is most likely an actual improvement rather than just a shuffling of the numbers.

As it was implemented, the use of context to improve matching was very naive. Given the improvement it makes sense to try further improvements to the contextual approach. Such improvements may be things like giving preference to mentions which are in the same sentence as the previously matched mention.

5 Final Comments

In conclusion I would like to offer the following comments about the work done above. Overall the data seems to suggest the biggest improvement gains

were provided by adding types (and thus type conflicts) to mentions. While this was quite useful, the addition of meta data in general is a slow process as is need be done by hand. However, it seems very reasonable that large future improvements may be had from further detail in meta data. For example *type place* could be further broken down into cities, countries, or sub-city sized areas. Mentions of *type thing* could be further labelled as physical things or abstract things. The number of labels is potentially very large so proper choices for what to use as meta data will be needed.

Secondly, as mentioned above, the definition of context could be refined. Ideally a context would be a collection of sentences or phrases that represent a single idea. The vagueness of this concept is what makes it most challenging to implement in software.

Finally, though our system did show large improvements in the B^3 score when compared to either of the two baselines, the final MUC score was worse than simply lumping all the mentions together. Such results lead me to question the validity of either my approach or the MUC score as an accurate guide for improvement.