CS224n PA3

William Hamilton, Raphael Townshend*

November 9, 2014

1 Performance

Tables 1 and 2 outline the performance of our non-naive systems.

	MUC			B3		
	Precision	Recall	F1	Precision	Recall	F1
Better Baseline	0.785	0.679	0.728	0.684	0.631	0.656
Rule-based	0.797	0.745	0.770	0.728	0.654	0.689
Classifier-based	0.820	0.695	0.752	0.776	0.621	0.690

Table 1: Performance on the dev set.

	MUC			B3		
	Precision	Recall	F1	Precision	Recall	F1
Better Baseline	0.772	0.574	0.658	0.757	0.577	0.655
Rule-based	0.785	0.694	0.737	0.760	0.657	0.705
Classifier-based	0.823	0.632	0.715	0.837	0.610	0.706

Table 2: Performance on the test set.

2 Baseline Systems

The most naive baselines implemented were the one cluster and all singletons baselines. The recall of the one cluster is 100% because every coreferrence in the gold set is covered in the one cluster. However, the precision suffers because of this. On the other hand, the all singletons baseline has a 100% precision because it essentially marks no coreferrents. Yet in this case the recall suffers greatly (0 in the MUC metric!).

For the better baseline, we kept statistics on which mention heads were coreferrent with one another. Then, mentions were marked as coreferrent if they matched either other

*Both authors contributed equally to this assignment. Code and writing was contributed by both authors. Raphael focused more on the rule-based system, while William focused more on the classifier-based system.

exactly or if the previously collected statistics indicated their heads could be corefferent. This method obtained good baseline results, and yielded a useful starting point for the following, more complex systems.

3 Rule-Based System

3.1 Overview

The rule-based system employed the better baseline as a starting point, and then added a number of other rules on top to improve performance. The primary rules added were an augmented Hobbs' algorithm along with exact head-matching.

3.2 Rules

A useful rule added was performing an exact head matching step after the exact matching step but before the coreferrent head checking. This allowed for a noun phrase such as *God's* to be linked to a reference such as *God* before relying on the coreferrent head statistics computed in training. The reasoning behind this is that exactly matching heads are a much better predictor of co-reference as opposed to just having seen the two heads co-refer previously. This feature was extended and further used in the classifier system by encoding edit distance between heads (see **HED** under 4.1). However, positioning this pass between the exact matching and the coreferrent head matching would not be easily implementable in a classifier based system because such a system would make only a single pass and not be able to account for such a multi-pass approach.

The basic Hobbs algorithm was also implemented, but was then modified by improving the final check stage. If the proposed mentions were both pronouns, gender, number, and person were checked to make sure they were the same. However, we also estimated non-pronoun plurality by checking the plurality of the head word (i.e. if the last letter is an 's'). In addition, a first pass at recognizing places was made by creating a Country idenitification module. Mentions that were recognized as being a country then imposed additional constraints on the coreferrent node, forcing it not to be a first or second-person singular pronoun (e.g. *I* could not refer to *Yemen*). Further work could be performed in this direction by adding a person recognizing module (e.g. if *Mr.* or *Mrs.* is present in the mention) and encoding that places and people cannot be coreferrent. It should also be noted that Hobbs algorithm could not be used in a classifier based system because of the multi-pass and state-based nature of the algorithm.

3.3 Testing and Error Analysis

Rules were designed by observing the system's output mistakes and adding rules to correct for the observed errors. The exact head matching rule was designed using this procedure when the following sentence was observed (*God* and *God's* were coreferrent):

```
{{God 's}} bless {{you}} and guide {{your}} steps to good ,
and {{God}} make {{you}} someone {{he}} is satisfied with [...]
```

Another source of errors is when the speaker in the document changes. It is not always possible to tell definitively when the speaker has changed, though the formatting of the document provides cues. For example:

```
May {{Allah}} forgive {{you}} in the protection of {{Allah}} .
-----------------------------------------------------------
{{My}} outstanding brother : May {{Allah}} reward {{you}} [...]
```

In this case, the first *you* and *my* are coreferrent because the speakers have changed. In a classification based system it may be possible to learn the separating line os present when the speaker changes, but designing a purely rule-based approach to this would be very tuned to whatever examples we are dealing with (since the solid line delimiter is very example-specific).

This leads to the more general point that rule-based and non-statistical nature of the system can easily lend itself to brittleness because there is no notion of uncertainty. Thus, a rule designed to fix one issue could easily cause breakage in other parts of the system.

4 Classifier-Based System

4.1 Description and Motivation of Features Used

The features used in the final system are described below.

- **MDI** [IntIndicator] - The number of mentions in the document occurring between the candidate and target mentions. Motivation: captures salience of candidate.
- **HED** [IntIndicator]- Levenshtein distance between candidate and fixed mention head words (subsumes exact match). Motivation:"fuzzy" extension of the exact match criterion.
- **(CAND/FIXED)_NER** [StringIndicator] - NER tag of candidate and fixed mention head words (adding the conjunction hurt the scores). Motivation: different NER types are more/less likely to act as referents.[1]
- **(CAND/FIXED)_POS** [StringIndicator + Conjunction] - POS tag of candidate and fixed mention head words (and conjunction of both). Motivation: words with different POS tags are more/less likely to act as referents in general and/or co-refer.
- **GEN** [Indicator] - Whether the gender of the fixed and candidate mentions match. Motivation: gender of coreferents usually correspond (except in small number of cases, e.g. "sailboat" referred to as "her").
- **SPEAK** [Indicator] - True if both mentions are quoted, have first-person pronoun head words, and have the same speaker. Motivation: first person pronouns almost always refer to the speaker.

Some intuitively reasonable features that were examined but actually hurt results on the dev set included: a feature on the distance between the sentences in question,

[1]Despite the fact that it seems strange to have an indicator feature on the target mention only, this actually helped results a marginal amount.

a feature on the compatibility of the grammatical number of the fixed and candidate mentions, and a feature on the Levenshtein distance between the entire mentions (not just head words). Our hypothesis is that these features led to overfitting.

4.2 Error-Analysis

Perhaps the most glaring systematic error in the classifier system is its tendency to chain together pronouns, regardless of their compatibility. For example, in the sentence "They turned against it", the system chains together "They" and "it". Or, even more egregiously, in the sentence"I thank you my pious brother.", the system chains together all the pronouns! These errors could be addressed by adding a more fine-grained feature on the pronoun type. That is, the **(CAND/FIXED)_POS** features could be augmented to include more more refined categories of pronouns. The conjunction of these more refined features would also be necessary in order to learn binary relationships (e.g.,"it" and "you" are incompatible).

The above example errors also illustrate a second systematic shortcoming of the system: it tends to chain together subjects and objects, whereas this should only happen in rare cases with reflexive verb constructions (e.g., with the "self" suffixed reflexive pronouns). This issue could be addressed by adding a conjunction feature on the grammatical role of the target and candidate mentions, and a feature indicating the presence of a reflexive verb/pronoun construction (this second feature would also be conjoined with the grammatical role conjunction).

Lastly, the system often failed to properly deal with first-person speaker pronouns despite the fact that we explicitly added a feature, **SPEAK**, to address this. Our hypothesis is that, since this feature fires rarely in training, it has low weight (which we verified) and thus is overshadowed by other simpler features (e.g., the **HED** feature).

4.3 Feature Ablations

Features	All	-**MDI**	-**HED**	-**NER**	-**GEN**	-**POS**	-**SPEAK**
MUC F1	0.752	0.727	0.424	0.750	0.744	0.716	0.751
B3 F1	0.691	0.670	0.520	0.688	0.672	0.657	0.683

Table 3: Performance on dev set with different feature (sets) ablated. For the -**POS** and -**NER** conditions we are ablating all features of that type.

Ablation studies revealed that the head word edit distance, mention distance, and POS-based features were by far the most important, as the scores dropped drastically when any of these features were ablated. This makes sense since the **MDI** feature is the only one that captures saliency, the **HED** feature captures (near)-exact matches (very important), and the POS-based features contain information about whether or not the mentions are pronouns etc. and captures some aspects of compatibility.

That said, running the system with only these 3 most important types of features gave an MUC F1 of 0.739 and a B3 F1 of 0.669, a significant drop from the full system's performance. Thus all the features played a part in the overall performance of the system.