CS 224N Assignment 3: Coreference Resolution

John Miller, Catalin Voss

1 Introduction

Given a some document, our goal is to find all of the *mentions* in a document that refer to the same *entity*.

2 Evaluation Metrics

We report results for two evaluation metrics. All results are for the default training size of 100 documents.

- MUC considers each entity as inducing an equivalence relation over mentions. Graph theoretically, each entity represents a connected component in the graph of reference. Precision is the number of edges between mentions that are actually correct, i.e. that lie in the same connected component. Recall is the number of edges provided divided by the number of additional edges required to produce the correct equivalence classes.
- B^3 measures precision and recall for entries in a reference chain. For each chain, precision is the percentage of mentions in the predicted reference chain that are in the true reference chain, and recall is percentage of mentions that in the true reference chain that are in the prediction. Then, total precision and recall is a weighted average over each chain.

3 Baseline Systems

3.1 Single Entity

The first baseline optimizes for recall, so each mention is is resolved to a single entity. Every mention in the same connected component of the reference graph, so no links are added to connect components of the graph. Hence, this baseline should achieve perfect recall and low precision due to many spurious edges.

3.2 All Singletons

The second baseline optimizes for precision. Each mention is resolved to its own singleton entity, so no edges are added to the reference graph. None of the added edges are incorrect, so the system achieves perfect precision, but has low recall since edges must be added to connect components.

3.3 Better Baseline

We implemented a simple head-matching algorithm. Before test time, the algorithm examines the entire training corpus and maintains a counter for each pair of observed coreferent heads words. At test time, the algorithm tries to match a mention to an exact match. If no exact match exists, then the mention is added to any cluster that contains an observed coreferent head word. Otherwise, the word is added to its own singleton cluster.

We attempted to deal with pronouns by excluding them from the co-occurent head count, since these connection are less reliable. This increased precision by 1-2%, but decreased recall by 13% by removing many potential connections. Hence, this naive filtering actually reduced the F1 score in this case. Baseline performance in the following table.

(Dev, Test)		MUC			B3	
	Precision	Recall	F1	Precision	Recall	F1
One Entity	(0.769, 0.744)	(1.0, 1.0)	(0.870, 0.853)	(0.164, 0.127)	(1.0, 1.0)	(0.283, 0.225)
All Single	(1.0, 1.0)	(0.0, 0.0)	(0.0, 0.0)	(1.0, 1.0)	(0.247, 0.273)	(0.396, 0.429)
Better Base	(0.784, 0.792)	(0.726, 0.680)	(0.754, 0.732)	(0.699, 0.778)	(0.644, 0.645)	(0.671, 0.705)

4 Rule-Based Coreference Resolution

In this section, we describe a rule-based coreference resolution system that does not include a training step (except to collect statistical information) and uses a number of handwritten rules to resolve the coreference.

We implemented a multi-sieve approach similar to that described in the paper "A Multi-Pass Sieve for Coreference Resolution" by Raghunathan et al . We apply deterministic coreference models in multiple passes from highest to lowest precision, trading off precision for recall as the algorithm continues. Some of our features as well as the general ordering is the same as in the paper. In addition we implemented Hobbs' Algorithm as one pass for pronoun resolution. We describe each stage in turn.

- **Exact Match** The first pass clusters all mentions at are an exact textual match. For example, this step successfully clusters all references to 'Jiu Taiping" together. This step is extremely high precision and ensures that all completely unambiguous mentions are correctly resolved. Results in baseline performance of 60%.

- **Hobbs' Algorithm** The second pass uses Hobbs' Algorithm, a common set of rules specific to English language to resolve pronouns and merge mention clusters that Hobbs' predicts as connected in either direction. This algorithm is shown to have relatively high precision and resulted in a 6% score boost when included this early in the multi-pass process.

- **Strict Head Matching** Clusters mentions that have matching head words, subject to constraints such as word inclusion and modifier compatibility on the clusters created in steps 1 and 2. Improves performance by 1.8%, and has precision loss of only 0.1 due to the restrictiveness of the constrains. Successfully clusters "village" and "the village" despite the article. Fails to cluster "the peasants" and "China's peasants" due to word inclusion.

- **Relaxed Head Matching** Two passes successively remove the constraints on clusters used for head matching. Improves performance by 8% to 72% MUC by raising recall 10% to tradeoff for some loss of precision. Succesfully matches "Mom's cell phone" to "phone", despite the fact that "Mom's cell" fails the word inclusion constraint. On the other hand, frequently fails to cluster similar words with different morphology, such as "house" and "home."

- **Mention Agreement** Attempts to deal with distinct morphology by clustering mentions that have distint text, but identical lemma's, gender, named entities, and numbers. Results in a 4% MUC and 3% B3 improvement, and is able to succesfully resolve "the villager's", "villagers", and "the villagers '", in spite of the distinct morphology and surrounding context that caused relaxed head matching to fail. Frequently fails to correctly cluster pronouns, often erroneously assigning several incorrect pronouns to a single cluster or constructing mutliple entities around distinct pronouns that should be clustered together.

- **Pronoun resolution** Attempts to merge distinct clusters that are linked by common pronouns. We check if two pronouns share a common gender, type, number, and speaker. This

is a relatively imprecise process. We improve F1 by 1% by increasing recall, but losing 4.3% precision. We successfully manage to cluster "the members of the assembly", "their", "they", and "the National Parliament" using this approach, which linked "their" and "they" together to merge these clusters.

It is important to note that steps 3 and 4 (strict and relaxed head matching) are particulaly useful in our rule-based classifier, but virtually useless in the statistical classifier. This is because we are able to leverage the precise clusters from steps 1 and 2, as well as apply deterministic rules to potential head matches to ensure they are valid. In a classifier, we have no control over the precision of the clusters or the rules for using head matching. To validate this claim, adding this feature to the statistical classifier actually *reduced* performance by 7%.

The performance statistics for this rule-based approach are given in the table below.

(Dev, Test)		MUC			B3	
	Precision	Recall	F1	Precision	Recall	F1
Rule Based	(0.813,0.812)	(0.737,0.680)	(0.773, 0.740)	(0.707,0.785)	(0.683, 0.650)	(0.695,0.711)
Classifier	(0.802, 0.796)	(0.704, 0.648)	(0.750, 0.714)	(0.769, 0.807)	(0.615, 0.618)	(0.683, 0.700)

5 Discriminative Statistical Classifier

In this section, we describe our implementation of a maximum entropy classifier. Given the training data produced by the starter code, we construct a set of features indicators f_i over the negative and positive coreference classes and some data about the pair. For example, $f_j = \{c = \text{"coreferent"}, d = \text{JJ}\}$ is a feature that fires for coreferent pairs when one mention is an adjective. Each feature f_i has an associated weight λ_i, which is used to score each class assignment as $\sum_i \lambda_i f_i(c, d)$. So

$$\Pr(c \mid d, \lambda) = \frac{\exp \lambda^T f(c, d)}{\sum_{c'} \exp \lambda^T f(c', d)}$$

The parameters λ_i are optimized to maximize the likelihood of the training data. At test time, the model will consider new mention m_1 and attempt to find a previous coreferent mention. If there is some m_2 where coreferrent class assignment score for the pair (m_2, m_1) exceeds the not coreferent score, m_1 is added to m_2's cluster. Otherwise, we create a new singleton cluster for m_1.

To select the set feature functions that maximize performance, we implemented 25 features, and then performed a crude forward search, adding the features that increased performance one at a time.

- **Exact Match**:*Intution:* Each time we see a proper noun like "Obama" it should be clustered with all other "Obama" *Results:* Achieved 60% MUC and B3 on the Dev Set. For example, clusters all "Hezbollah" mentions into the same entity *Problems:* Requiring exact textual matching means that "Saddam" and "Saddam's" or "Saddam Hussein" are grouped into distinct clusters.

- **Head Word Matching** *Intuition:* Several exact match errors were failure to cluster "house" and "the house". By focusing on the head of each mention, it is easier to see that these refer to the same entity. *Results:* Achieved 68% MUC and 64% B3 on the dev. set. Successfully clustered "their team", "the national team", and "the team", which were not successfully clustered using only exact mention matching.

- **Lemma Match**: Check if both head tokens have matching lemmas *Intuition:* Exact and head word matching fail to cluster similar head words like "Paqueta" and "Paqueta's" with different morphology *Results:* Raised MUC to 73.7% F1 and B3 to 67.3% on the dev set. Successfully clustered "Paqueta" and "Paqueta's" and similar constructions. False for lemma matching received large negative weight, so distinct lemmas strongly suggest two entities should not be clustered. *Problems:* Frequently grouped he, his, him, etc into a single cluster because they share a lemma. E.g. all of these pronouns matched with "Al-Harti" instead of divided among "Jasim" and "Mr. Paqueta."

- **Pronouns with Matching Heads** Triggered if both mentions are pronouns with matching heads. *Intuition:* Signal for matching head lemmas is not as strong with pronouns *Results:* No improvment in MUC or B3, but changed output. All of the pronouns are no longer matched to a single entity as before, but rather clustered around different entities. E.g. "Mr. Paqueta" and "Jasmin' each have several "he, his, ...", but they are not the correct "he" and "his" mentions.

- **Distance to Previous Mention (Mentions and Sentences)** *Intuition:* Pronouns typically appear close to the entities they reference. *Results:* Slightly lowered in MUC F1 score and left the B3 score unchanged. Occassionally misclassified mentions of the form "he said...(entity)" because "he" is textually closer to (entity), but is not coreferent, suggesting this is not a reliable signal.

- **Gender Agreement** *Intuition:* Gender will better resolve pronoun coreferences. *Results:* 1% boost over only exact and head matching, and no gains when combined with lemma matching. Disagreement was assigned a large negative score, which makes sense since "he" and "she" are unlikely to refer to the same entity. For pronouns, much of this information is already captured in lemma head matching.

- **Parts of Speech and Speaker** *Intuition:* More syntactic information or speaker information will better disambiguate pronouns *Results:* No improvment.These features received low feature scores, so POS or speaker tags are too ambiguous to be useful.

- **Named Entity of Mentions** *Intution:* People are not typically coreferent with locations or organizations. *Results:* Modest 0.6% improvement in performance over subsets of previous features. Negative feature received a large negative weight, confirming our intuition. Previously, the model clustered "Xuzhou" (a place) and "the reporter" together, but now the mentions are separated into distinct clusters.

The core improvement came from exact and lemma matching of head words. Analysis of the output reveals the model is extremely accurate with regards to proper nouns and named entities , e.g. "The Saudi national team" $\rightarrow$ {the national team}, {the team}, ... Intuitively, these should be the easiest to resolve because there is little ambiguitity and the words themselves are relatively unique within clusters.

The both models still perform extremely poorly on pronoun coreference. The techniques for more concrete entities were useless for pronouns, and other features did not improve performance. The ambiguitity associated with pronoun use makes it difficult to construct meaningful features, and the same pronoun words appear in multiple clusters. We attempted to deal with this by incorporating more features specifically engineered for pronouns, namely as Hobbs' algorithm. However, the performance gains were marginal and the majority of pronouns still remained misclassified.