**Stanford CS224n – Project 3**

**By Patrick Manion & Jie Shen**

**1. Baseline Experiments**

For Coreference Resolution, we began by implementing two simple baseline models: a singleton model, which simply assigns every mention in the document to its own cluster, and the one cluster model, which assigns every mention to the same cluster, using the MUC and B3 metrics. The results indicate serious issues with MUC, which gives excessive high scores for one cluster and low scores for singleton. Both issues stem from MUC’s strategy of comparing links between the model and reference outputs, which are poorly defined when too few or too many links are present. B3 gives much more reasonable answers by evaluating the purity of each cluster and how well the clusters tie together the entities from the reference.

Two additional baselines were evaluated. The first was a simple Baseline model that merges mentions that are exact string matches of each other. The precision of this is much lower than might be expected driven primarily by pronouns, which are all clustered together despite potentially referencing numerous different entities. A BetterBaseline was also tested which combined entities that had exactly the same headword in the mention. This provides a sizeable boost

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **MUC** | | | **B3** | | |
|  | **Precision** | **Recall** | **F1** | **Precision** | **Recall** | **F1** |
| **All Singleton (Dev)** | 100% | 0% | 0% | 100% | 25% | 40% |
| **OneCluster (Dev)** | 77% | 100% | 87% | 16% | 100% | 28% |
| **Baseline (Dev)** | 81% | 50% | 62% | 86% | 45% | 59% |
| **Baseline (Test)** | 81% | 39% | 53% | 91% | 44% | 59% |
| **BetterBaseline (Dev)** | 83% | 60% | 70% | 83% | 51% | 64% |
| **BetterBaseline (Test)** | 85% | 58% | 69% | 87% | 56% | 69% |

**2. Rule-based and Classifier-based Overview**

Expanding on the baselines, we first explored a rule-based model, which consists of a number of different rules that can build on each other to achieve the maximum performance. Generally, more precise rules are used early on to begin building meaningful clusters of entities and more lax rules are added at the end to take advantage of precise clusters. Lax rules are not added up-front as this can cascade errors through all the remaining steps and reduce performance.

The second model is a classifier-based model that uses a multiclass logistic regression model to learn weights to user-created features. The benefit of this approach is the model can help decide which features are important and include interactions between different features. However, the downside is it becomes more difficult to have features that build upon each other like a rule-based system, and it is also more difficult to include subject-matter expertise.

**3. Features**

For the rule-based system, we implemented a system similar in spirit to the multi-pass sieve Stanford system, which begins with high precision rules and moves to more relaxed ones later in the sieve.

We began our sieve with an exact string match, but learning from our baseline, we excluded any pronouns matches, which keeps precision very high. Afterwards, we implemented an acronym match that checks for the capitalized first-letters from one mention in another, which caught common errors in the training data like “the United Nations” and “UN” not being exact matches. After excluding any 1 letter “acronyms”, we found this rule gave high precision and a slight recall boost.

We then tried a simple rule to capture appositive constructs, but we found this actually reduced model performance due to many appositive-like constructs. For example, TV anchors often state their employer after their name like “Thelma Gutierrez, CNN,” and lists of items like “John, Tom, Mary, …”. Additional restrictions like requiring a named-entity match improved the rule but not sufficiently enough to include in the final model.

We then moved on to somewhat looser rules based around the head-words of the mention. First, we incorporated a match for mentions that had all the same words after excluding 25 of the most common stop-words. This helped capture obvious mistakes like “the Urban Institute” and “Urban Institute” not being merged. We also added a rule to merge clusters that had mentions with the same words *before* the head word. This captured a number of issues such as more descriptive early mentions like “the bill allowing food and medicine sales” not being matched with later mentions of the same topic like “the bill”, which occurs frequently in everyday discourse.

We then found that a rule combining mentions with the same headword boosted recall significantly and, despite a small decrease in precision, also boosted F1. Similar to the rule above, this helped capture mentions with more adjectives early on in the discourse that did not show up later like “the aged shuttle Discovery” being shortened to “the shuttle”. The F1 was further enhanced by matching headwords regardless of case, which helped solve errors in the text itself like (e.g. “the kursk” and “the Kursk”) as well as proper nouns being referred to later in the discourse as general nouns (e.g. “the Second World War” and later “the war”).

We then tried even more loose rules, but many did not work out as planned such a combining mentions with over 2/3 overlap in unigrams. Another example is a lemma match on the headwords, which ended up incorrectly combining a mention of a plural group with a mention of a single individual like “U.S. **officials**” with “One senior **official**”.

After all of these rules, we also tried several pronoun matching strategies. We implemented Hobb’s Algorithm, but this only returned a single candidate and did not provide a sufficiently large boost. However, we did find that finding the closest mention whose headword matched the named entity, gender, and number of the pronoun improved performance even more. This is likely due to the added constraints on NER, gender, and number, and Hobb’s Algorithm may have provided a stronger boost if we had time to include these.

For the classifier-based model, we tried a large number of features including distance between mentions, word set similarity/overlap, pronoun matching, named entity recognition, and more. However, we actually found that a set of initial headword features were actually the most powerful of all, and adding more features often didn’t improve or actually hurt the performance.

The power of the head words was surprising at first but makes sense as the head is the core of the mention and contains some of the most significant information while many other features appeared to distract the model. For example, “the Jingle Cat” and “his cat, Cheesepuff” only has one word in common, and “his” in the second mention can be misleading. They wouldn’t be linked using exact word match, word overlap or pronoun match, other than head word match. Another example “a Norwegian transport ship” and “the damaged ship” are correctly classified into one entity by focusing on head words.

One interesting thing from modeling perspective is adding more features sometimes can hurt the performance. This is a bit contradictory to what our understanding of machine learning, in which we dump a bunch of features into algorithm to let data choose. Here apparently more personal judgment from linguistics on variable selection is important.

**4. Results**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **MUC** | | | **B3** | | |
|  | **Precision** | **Recall** | **F1** | **Precision** | **Recall** | **F1** |
| **RuleBased (Dev)** | 79.85% | 67.88% | 73.38% | 73.09% | 58.71% | 65.12% |
| **RuleBased (Test)** | 84.13% | 66.89% | 74.53% | 82.29% | 64.16% | 72.10% |
| **Classifier (Dev)** | 81.69% | 67.84% | 74.12% | 80.82% | 57.92% | 67.48% |
| **Classifier (Test)** | 81.76% | 63.75% | 71.64% | 84.37% | 59.92% | 70.08% |

The tests of our classifiers showed

3. Result analysis

1. The two mentions have the same meaning, but in different wording. For example “ the U.S.” and “ the United States”, “the US. carrier” and “the ship” are classified into different mentions. This can be improved using external data source for abbreviation. Synonym matching is another way to handle this situation. We tried Synonym matching in BetterBaseline algorithm by extracting the two words which often occur together in the coreference in training data as synonym. But the training data is not large enough and the approach didn’t word due to data sparsity.
2. The difficulty to link a pronoun to a non-pronoun word. For example, “the disease he got” and “it”, “Dan” and “he”. One way is to improve Pronoun matching features. Another way is to apply Artificial Intelligence into the coreference to learn from the contents using Machine Comprehension of Text, which will be our final project topic.