Stanford CS224n – Project 3
By Patrick Manion & Jie Shen

1. Baseline Experiments

For Coreference Resolution, we began by evaluating two simple baseline models: a singleton model, which 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 B^3 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. B^3 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 added a rule to combine the entities of mentions with the same headword. This provided a sizeable boost in recall but also a reduction in precision from discarding possibly important modifiers that distinguish unique entities (e.g. "the angry child" vs "the happy child"). However, net-net, this improved the F1 on the test set by 16% under MUC and 10% under B^3.

	MUC			B^3		
	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)	80%	60%	69%	83%	51%	64%
BetterBaseline (Test)	82%	58%	68%	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 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 the additional information already added by the precise rules. 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 – Rule Based

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 leveraged two papers on Stanford's multi-pass sieve systems for potential features (Raghunathan 2010 and Lee 2011). We also focused our efforts primarily on improving the training scores because there were 1,600 training samples but only 63 development samples, which we found prone to noise.

We began our sieve with an exact string match, but learning from our baseline, we excluded any pronouns matches, which keeps precision very high (99.2% B^3). Afterwards, we implemented an acronym match that checks for the capitalized first-letters from one mention in another. This helped catch common errors we found in the training data like "Taipei International Book Exhibition" and "TIBE" not being exact matches, but also introduced some errors like "peace talks sponsored by the United Nations" matching with "the UN" itself. However, after excluding any 1 letter acronyms, we found this rule maintained high precision (99.0% B^3) and a slight recall boost (0.11% B^3) for a 0.08% boost in B^3 F1.

We then tried a simple rule to capture appositive constructs, but we found this actually reduced model performance due to many appositive-looking 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 accuracy of the rule, but further refinement appears necessary to provide a net boost in F1.

We then moved on to somewhat looser rules. We started by matching 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 matching. We also added a rule to merge clusters that had mentions with the same words *before* the headword. 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" – a phenomenon that occurs frequently in discourse. This boosted B^3 recall from 42.1% to 43.9% but still gave 97.7% precision.

However, we noticed that extra description occurs at the beginning as well as the end of the sentence, and a rule merging clusters with the same headword boosted recall an additional 7.5% (to 51.3%) while lowering precision to 95.1% - a net boost of 6.1% to F1. This helped with a number of additional errors 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 all mentions with over 2/3 overlap in unigrams. When using the shorter mention as the reference, this rule ended up incorrectly joining things like "those who fought and won World War II" with "World War II". However, even using the longer mention as the reference did not improve performance as even small differences in very long phrases can entirely change the entity they refer to such as "the **weaker** of the two networks" vs "the **stronger** of the two networks" and "**a member** of a terrorist group with links to al Qaeda" vs "a terrorist group with links to al Qaeda".

Another example of a loose rule that did not work 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 first implemented a version of Hobb's Algorithm that found a single match but only merged the cluster if the pronoun and match agreed on NER, gender, and number. For MUC, this raised recall from 0.452 to 0.501 but dropped precision from 0.972 to 0.932. We also implemented a simple pronoun rule that picked the closest antecedent that matched on NER, gender, and number. This ended up increasing recall far more to 0.679 but also dropped precision to 0.789. However, on net, Hobb's only increased MUC F1 by 0.035 while the simple pronoun match increased MUC F1 by 0.126. This seems to be driven by our Hobb's implementation only returning a single match, which keeps precision high but limits how much it can increase recall.

4. Features – Classifier

For the classifier-based model, we started by porting over a large number of simple rules that we leveraged in our rule-based model and turned them into indicators. We started with indicators when there an exact match overall or a headword exact match, part-of-speech match, NER match, lemma match, and Noun/Proper Noun/Plural Noun match. These simple rules, surprisingly, reached 0.711 MUC F1 and 0.694 B^3 F1 on the training data, which is far better than the head-match rule achieved on our rule based model.

These first features we picked appear to be perfect for a machine learning system. Headword matches were a powerful in the BetterBaseline model, but we could only use the NER, lemma, and other characteristics as all on or all off. However, the ML model can learn how important each feature is and give them a fractional weight so that some combinations are sufficient for a match but not others (e.g. PoS, Plural Noun, and NER).

Unfortunately, we tried a number of other features that only hindered the final performance. For example, we tried a word inclusion indicator if every word in the candidate mention occurred in one of the entity's mentions. This was driven by a desire to capture discourse situations where someone describes an entity in a novel way using already used words. For example, we may recognize "**huge** scary **monster**" and "frightful **green** creature" as part of the same entity, in which case it would be reasonable to assume "**huge green monster**" is part of the same entity.

We tried a number of other indicators that didn't work including headword number alignment, gender alignment, person alignment, whether one or both words were pronouns, whether the words contained upper case characters, and modifier alignment, which were all tried or deliberately included/excluded in our rule-based system. We also tried numeric features that would be difficult in a rule-based system including distance between the sentences of the mentions, the distance between the mentions themselves, and the count of word overlaps, but none of them improved performance.

5. Results

The results of our tests show strong performance on both MUC and B^3 F1.

	MUC			B^3		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
RuleBased (Train)	79.15%	68.23%	73.28%	77.48%	64.02%	70.11%
RuleBased (Dev)	78.94%	73.44%	76.10%	69.20%	62.60%	65.74%
RuleBased (Test)	80.89%	70.62%	75.41%	78.79%	67.28%	72.58%
Classifier (Train)	82.16%	62.68%	71.11%	84.71%	58.85%	69.45%
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%

6. Improvement Ideas

Of the remaining issues, by far the most common are pronoun misalignments so improvements like a more advanced Hobb's implementation, tagging features (PoS, NER, etc.), and incorporating more entity-level knowledge seem promising areas. In addition, adding outside knowledge sources like Wikipedia for synonyms, abbreviations, etc. could replace some of the rough, broad rules we used with more intelligent mapping approaches that could eliminate another common source of errors.