Now That You Mention It

I Do Know Who You're Talking About!

David Philipson and Nikil Viswanathan
{pdavid2,nikil}@stanford.edu
CS224n Fall 2011

1 Introduction

In this paper, we display the rule-based system we developed for coreference resolution. Using this system we are able to identify the real world entity to which each noun phrase refers. We were able to obtain a B^3 F1 score of 68.8 on our development set and 67.4 on the test data.

2 Related Work

For our coreference resolution algorithm we drew heavily off of the Multi-Pass Sieve work in Raghunathan et al. (EMNLP 2010)[1]. After trying several different techniques, our final approach closely models this sieve and also adopts several of the techniques for handling and merging clusters of coreferent mentions with a couple extensions. We found that we actually obtained different amounts of performance increases with each level of the sieve than the paper. On certain passes we saw a greater increase than expected and other passes resulted in a more modest improvement than Raghunathan et al. experienced. We attribute this to using different data sets than the paper did, which also indicates that the algorithms are at least somewhat tuned for the specific domain of the corpus.

3 Simple Models

We initially implemented several simple models to test out different approaches and see how the scoring functions reacted to precision / recall tradeoffs.

Assigning all mentions to the same cluster leads to perfect recall scores in both the MUC and B^3 metric, which makes sense as we are indeed putting together all mentions which are coreferent. However, while the MUC metric gives this algorithm a rather high precision and respectable F1 score (0.85), the B^3 metric gives the algorithm a very low precision and correspondingly low F1 score (0.25). This is a good indication of why we will want to care about the B^3 metric.

Assigning all mentions to singleton clusters gives perfect precision in both metrics, since no cluster contains mentions from more than one coreferent group. On the other hand, both metrics give low recall and hence low F1 (MUC: 0, B^3: 0.25).

For our improved baseline algorithm, we implement a simple head-matching algorithm as follows: during training, we count the number of times each headword appeared in a sentence with each other headword. Then given a document, each mention "points" to the other mention whose headword appeared most frequently with this mention's headword. We then assign clusters so that each mention appears in the same cluster as whichever mention it pointed to. This gave us a baseline score, in the B^3 metric, of precision 0.71 and recall 0.53, with F1 score 0.61.

We observe that compared to the provided baseline, which had B^3 scores of precision 0.89, recall 0.42, and F1 0.57, our improved baseline has traded away precision in order to gain recall. This makes sense, as the improved baseline marks mentions as coreferent more aggressively, so we should expect to see lower precision (our clusters will contain more mentions from distinct true clusters, incorrectly) but higher recall (we are more likely to have mentions from the same true cluster assigned together). The pattern we will see throughout is that methods of aggressive clustering can be used to improve recall, while methods of conservative clustering can be used to improve precision.

4 Rule Applying Algorithm

Our primary algorithm was a rule based coreference resolution classifier which deterministically took repeated passes over the mentions and attempted to cluster them together.

4.1 Initial Attempts

In the beginning we invented rules ourselves and tried adding them each as their own pass. Initially we just added an exact match clustering algorithm which combined mentions if they had the same text, and this gave us a baseline B^3 F1 score of about 57-58. We then added rules to match number and gender of the mentions and other simple features. What we found was that since our rules were fairly simple, they made the performance actually go down as they would either merge freely by including too many clusters or have a very strict requirement for merging and not merge any clusters. We decided to follow a more structured approach and looked at the Raghunathan paper for inspiration.

4.2 Mention Processing

We had several strategies and techniques for the general approach to handling clusters. We applied these on every pass through the mention and coreference clusters.

4.2.1 Mention Order

When looking at possible candidate coreferent mentions, we look at all of the mentions in the current sentence in order of mentions in a breadth first search traversal of the parse tree of the sentence. For the previous sentences we follow the same traditional left to right breadth first search order if current mention is a pronominal mention and flip the order and explore the nodes in right to left breadth first search order if the current mention is a nominal mention. This order provides syntactic salience for the respective mention type and also favors closer mentions.

4.2.2 Merge Selection

When considering each cluster, we only looked at the first mention in each cluster as a possible anchor point for merging with other clusters. We chose this strategy as this mention is more likely to have more modifiers and also more likely to be better defined than subsequent mentions which drop some words. In addition only looking at the first mention in the cluster improves the accuracy of the merging as there are fewer possible previous mentions.

4.2.3 Attribute Sharing

For each cluster, we track the union of all of the target attributes over all of the mentions. This mitigates missing attributes. The attributes we track include NER type, gender, number, speaker, and animacy. Keeping track of the union of these attributes allows us to experiment with different types of constraints when merging clusters (discussed below).

4.3 Passes

We have a multi-pass model which runs through the current clusters multiple times and iteratively uses different criteria to merge them. The passes run in order of level of confidence and the initial passes have a very high precision while hesitating to merge clusters if it is not absolutely sure of the coreference. The subsequent passes use weaker restrictions in an attempt to merge potential clusters and improve the recall.

4.3.1 Pass 1 - Exact Match

In the first pass, we merge mentions that have the same text. We initially did this for all types of words, and after inspecting the data, limited the pronoun exact text match. In general, for example, the word "he" could be used in many different sentences within a given document to refer to different people (see the discussion below).

4.3.2 Pass 2 - Appositives and Predicate Nominative

We attempted to implement matching based on appositives and predicate nominatives. To identify appositives, we searched for two mentions, both children of the same noun phrase (NP), which were separated by a comma. To identify predicative nominatives, we searched for cases where we had a NP with two children, the first of which was the first mention, and the second of which was a VP representing a linking verb (one of "am," "are," "is," "was," or "were) followed by the other mention. In either case, we would declare the two involved mentions to be part of the same cluster. Neither aspect of this pass proved to be particularly effective: very few such examples were found throughout the documents, and so the effect on our scores was negligible.

4.3.3 Pass 3 - Strict Head Matching

In the third pass, we used several filters to determine if a mention was strongly coreferent to another mention. This phase still maintained a high precision while improving the recall about 2 B^3 F1 points. We used the conjunction of all of the subsequent features:

Cluster Head Match

The head word of the current mention matches any headword in the current cluster of the antecedent under consideration.

Word Inclusion

All of the words in the current cluster, ignoring stop words, are contained in the cluster of the mention under consideration for possible coreference. Note that these words are taken from all of the words in the mention list, not just the headword of each mention. We used a list of 25 stop-words from the Stanford NLP[2] site.

Compatible Modifiers Only

The modifiers of the current mention are all contained within the modifiers of the potential link.

This is a mention to mention comparison and does not include the entire cluster. We only consider modifiers which are nouns or adjectives.

Not i-within-i
We exclude mentions which are in the i-within-i structure. This structure happens when one mention is a noun phrase and is the child of the other mention which is also a noun phrase structured modifier.

This pass provided the tools for the next several passes as we used these four feature constraints as the building blocks for the subsequent increases in recall by picking and choosing choosing which ones to apply.

4.3.4 Pass 4 - Alternative Head Matching

In this pass we alternatively relax the constraints on the previous pass in hopes to merge clusters which are possibly coreferent. For this pass we enforce the Cluster Head Match, Word Inclusion, and Not i-within-i features while ignoring the Compatible Modifiers Only constraint. ∼.2 B^3 F1 increase In the following passes we explore tests by omitting different combinations of constraints.

4.3.5 Pass 5 - Alternative Head Matching

In addition to this, we experimented with including and excluding other features on our own.

4.3.5.1 Pass 5a

In this pass we ignore the Word Inclusion requirement from pass 3 and keep the other features.

4.3.5.2 Pass 5b

We tried relaxing different combinations of constraints on our own. In this pass, we only enforce the Cluster Head Match and Not i-within-i features.

4.3.5.3 Pass 5c

The final tweak we tried to improve the recall was adding a final pass after 5.b which only requires the Cluster head Matching to hold. We saw small gains with 5b and 5c and slightly more significant gains with 6c.

4.3.6 Pass 6 - Relaxed Head Matching

We implemented a relaxation of the cluster head matching feature and allowed the head word to match any headword in the mention's cluster. We also kept the word inclusion and not i-within-i requirements for this pass. Modest improvements resulted from this pass and we also tried our own variations.

4.3.6.2 Pass 6b

As the attempt to improve recall still yielded a high precision and lower recall, we added an extra pass that only required the relaxed head matching from the pass 6 and didn't require the features of word inclusion and not i-within-i and the results of the attempt were to increase the overall F1 score by a couple tenths of a point which was fairly good at this stage.

4.3.7 Pass 7 - Pronouns

In pass 7, we assign attributes to clusters in the categories of number, gender, person/speaker, animacy, and NER tags. The attributes held by a cluster are the union of the attributes of each of the mentions contained in the cluster (so in particular, it is possible for a cluster to have attributes of, for instance, both "singular" and "plural"). For a given mention, we compute attributes as follows:

Number - if the mention headword is a pronoun, we produce the number (singular or plural) based on the pronoun. We also look at the part-of-speech tag of the mention: if it appears NN*S, then we add the attribute "plural," while other tags beginning with NN get the attribute "singular". Finally, we refer to a dictionary[3] of singular and plural words.
Gender - if the mention headword is a pronoun, we go with the gender of the pronoun. We also refer to a dictionary[3] of words for each gender ("male", "female", and "neutral").
Person/Speaker - if the mention headword is a pronoun, we go with the gender of the pronoun.
Animacy - if the mention is a pronoun, we go with the animacy of the pronoun. We also use the NER tags, marking "PERSON" as animate but any other NER tag as inanimate. Once again, we also refer to a dictionary[3] of animate and inanimate words.
NER Label - the attribute is simply the NER label.

To combine the clusters, we sort the clusters in the order they first appear in the document. Then for each cluster in order, we propose to merge that cluster with each preceding cluster. In order for a merge to occur, the clusters must "agree" on each attribute type: for example, they must have compatible genders.

5 Final Results

Run once on the final test set

	MUC Precision	MUC Recall	MUC F1	B^3 Precision	B^3 Recall	B^3 F1
Baseline	0.786	0.382	0.515	0.891	0.416	0.567
Pass 1	0.950	0.312	0.479	0.984	0.414	0.583
Pass 2	0.942	0.312	0.469	0.981	0.414	0.583
Pass 3	0.898	0.418	0.571	0.956	0.459	0.620
Pass 4	0.899	0.425	0.577	0.955	0.463	0.624
Pass 5 (abc)	0.893	0.496	0.638	0.940	0.512	0.663
Pass 6	0.892	0.495	0.637	0.939	0.511	0.661

Pass 6b	0.893	0.497	0.639	0.939	0.513	0.663
Pass 7	0.822	0.580	0.680	0.853	0.557	0.674

6 Investigation

6.1 Exact Match Refinements

While visualizing the mentions and exploring algorithm's process of clustering for pass 3, we noticed that there were several first and second person pronouns that were clustered together. This seemed correct but this led us to consider whether third person pronouns would be good fit for exact match clustering. We reasoned that the first and second person pronouns typically all corresponded to the same person throughout a document but the third person pronominal reference quite often was linked to non coreferent mentions. We played around with excluding different combinations of pronouns from the exact match and found that to our surprise only having the first person pronouns in the exact match actually performed the best. The classifier with second person pronouns added in were a close performer and had much better precision than the first person only so we kept them both in for the final classifier also because this made more lexical sense. We evaluated this when we only had implemented passes 1-3 and after final testing with all 7 passes we found that this choice was overshadowed by the final passes and only made a slight performance difference.

Number of previous sentences

We tested out different parameters for the backward exploration distance when looking for candidate mentions. Initially we only looked in the same sentence and found a significant increase in performance by looking to the previous sentence mentions. Adding additional previous sentences offered additional gains up to about 3 sentences. This corresponded to our linguistic intuition as coreference between sentences only typically span a few sentences at most, and the upper levels for the parameter values that we tested (~5 depth in previous sentences) showed the same performance.

Order of mention search

We found that the order of the BFS search had little no no effect upon results.

Attribute compatibility in pass 7

In pass 7, we need to determine when two sets of attributes are "compatible." We experimented with two different definitions of compatibility: the first was to require that the attribute sets for the two clusters overlap: for instance, {MALE} and {MALE, FEMALE} are compatible, but {MALE} and {FEMALE} are not. The second definition expands the first by also considering the clusters compatible if either of the sets is empty, for instance {} and {MALE, FEMALE}. Call the first definition INTERSECT compatibility, and the second INTERSECT-OR-EMPTY. We note that INTERSECT-OR-EMPTY leads to more aggressive clustering, and thus should be considered when we wish to raise recall, while the converse holds for INTERSECT. Through experimentation, we found that best results were achieved when using INTERSECT to check compatibility of number and

speaker attributes, while using INTERSECT-OR-EMPTY to check compatibility of gender, animacy, and NER tag attributes.

7 Conclusion

We found that a multi-pass approach was effective for trading off precision and recall until both could be improved. Our final model extends Raghunathan's, produces F1 scores in the high sixties, and effectively performs across a variety of documents. We saw modest gains with each new pass: no one pass produced huge gains, but we saw steady improvement as additional passes were added. Our final results were: 68.8 on the Dev Set and 67.4 on the Test Set.

References

1. Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, Christopher Manning
   A Multi-Pass Sieve for Coreference Resolution
   EMNLP-2010, Boston, USA. 2010.

2. Stop word list from the Stanford NLP site http://nlp.stanford.edu/IR-book/html/htmledition/dropping-common-terms-stop-words-1.html
3. All dictionaries referred are from the StanfordCoreNLP library http://nlp.stanford.edu/software/corenlp.shtml