Coreference Resolution

Kartik Sawhney (kartiks2) and Rebecca Wang (rwang7)

Overview

Coreference resolution refers to the task of clustering different mentions referring to the same entity. This is particularly useful in other NLP tasks, including retrieving information about specific named entities, machine translation, among others. In this report, we discuss our approach, implementation and observations for a few baseline systems, a rule-based system, and a classifier-based system. To quantify the effectiveness of our implementation, we use the MUC and B^3 measures (precision, recall and F1) for coreference evaluation. The difference in the two scoring metrics in how they define a coreference set within a text (in terms of links or in terms of classes or clusters) results in interesting observations as we discuss in the report.

Baseline systems

AllSingleton: In this model, every mention resolves to its singleton entities. None of the mentions within the clusters are incorrect, and hence we get a precision of 1.0. Recall, on the other hand, is pretty low for this model, since it is able to find no coreference whatsoever. This in turn results in a low F1 (0 for MUC due to 0 recall and 0.396 B^3).

OneCluster: In this model, all mentions are coreferent with a single entity, i.e. there is a single cluster. Clearly, there are several incorrect coreferent links (or edges) in this cluster, and so we would expect a low precision. Recall, on the other hand, is high here, since no more mentions need to be added to the cluster. This is in line with our MUC precision and recall scores of 0.769 and 1.0, resulting in an F1 score of 0.87, as well as our B^3 precision, recall and F1 scores of 0.165, 1.0 and 0.287. What is interesting here is the high MUC F1 score, which might be attributed to the dev set being highly skewed.

Better baseline: This system improves upon the baseline version by incorporating a simple head-matching algorithm. Initially, during the training phase, we maintain a list of head tokens that are coreferent with each other, and then use this data during the test phase to predict coreference. If it is the case that there is a match based on the data collected from the training phase, then the two mentions are marked as "coreferent"; otherwise the two mentions are added to their own singleton clusters. We observed an increase of around 9% and 5% in our MUC and B^3 F1 scores. Note that we exclude pronouns from consideration here due to the less reliable connections for pronominal resolution.

Rule-based coreference resolution

In this section, we discuss our hand-written rules used in the rule-based coreference system and their observed impact. Our implementation is heavily inspired by "A Multi-Pass Sieve for Coreference Resolution", Raghunathan et al. In particular, we adopt a similar multi-sieve approach, transitioning from high precision to high recall. The training phase collects statistics on head word matches to be able to implement the simple heard word matching algorithm described in the "better baseline" model above. The following is a discussion of the different passes.

Exact match (pass1): This uses the same approach as in the baseline system wherein if two mentions have identical text, they are marked as coreferent. Since this requires an exact match, precision is very high for this rule (B^3 precision of 0.985), however the recall is low (0.355), since this rule does not capture any other aspect that might lead to coreference.

Head matching based on training data (pass2): Here, we rely on the training data and the simple head matching algorithm used in the better baseline model. However, we use the additional constraint that the head word should be a noun or an adjective to avoid accounting for mentions that occur commonly or do not necessarily establish connection, such as pronouns. This rule increases the recall by almost 12%, increasing the F1 score by nearly 8%. The good performance suggests that the training data is highly representative of the dev set. Mentions such as "the lamb hot pot party hosted by the Beijing Association of Greater Washington" and the "Golden Pig Good Luck Spring Festival Party" (referring to the same party in the training and the dev sets) are marked coreferent after this step.
Strict head matching (pass3): Strict Head Matching Clusters mentions that have the same head words, subject to two more constraints-word inclusion and modifier compatibility. By "word inclusion", we mean that the coreferent mentions should have the same set of non-stop words, because it is uncommon to introduce novel information in later mentions. The compatible modifiers' constraint helps ensure that the mention's modifiers are all included in the antecedent candidate's modifiers. For this purpose, we only use modifiers that are nouns or adjectives (as they are more reliable to predict connection). Given these constraints, we observe a slight decrease in the precision, but a substantial increase in recall (18% for MUC and 7% for B^3), leading to a consequent increase in the F1 score (an increase of 0.14 MUC and 0.03 B^3 F1 scores).
Variants of simple head matching (passes4 and 5): In line with the work of Ragunathan et al., our goal here is to transition from high precision to high recall. Accordingly, in the next two passes, we relax the compatible modifiers and word inclusion constraints respectively. Comparing the scores after these two passes separately, we realize that the word inclusion feature is more precise than the compatible modifier (results in a greater decrease in precision). Overall, these passes results in an increase in MUC and B^3 recall by 7% and 6%, increasing the F1 scores to 0.74 and 0.66. Mentions such as "Houellebecq" and "the writer Michel Houellebecq…" are marked as coreferent after this step.
Relaxed head matching (pass6): This pass relaxes the cluster head match heuristic by allowing the mention head to match any word in the cluster of the candidate antecedent. To maintain high precision, we add the word inclusion constraint. We had hoped to observe a further increase in recall, however there is no perceptible difference. Our understanding is that perhaps, passes4 and 5 already account for relaxed head matching.
Nominal coreference using gender, number, speaker and lemmas (pass7): This pass attempts to account for distinct morphology by clustering mentions that have distinct text, but identical lemma's, gender, named entities, and numbers. This increases the recall by nearly 2%, slightly increasing the F1 scores. We had hoped to see a drastic improvement after implementing this rule (since the previous six rules relied on linguistic semantics, and this one was much broader), however the restrictive nature of the rule (all four properties should match) apparently lead to only a slight improvement.
Hobbs' algorithm (pass8): To implement pronominal coreference resolution, we implemented the Hobbs' algorithm using the pseudocode in the lecture. As expected, the algorithm helped our score, increasing the recall by 4% and the F1 score by 2%. Some mentions that get resolved include "their, the activists", "him, Michael" etc.
Pronominal coreference using gender, number and speaker (pass9): Finally, we use gender, number and speaker compatibility to merge distinct clusters linked by a common pronoun. This results in an increase of 2% in our MUC F1 score, thereby further fine tuning pronominal coreference obtained from Hobbs' algorithm in the last step. E.g. "They, them", "I, me" etc. get resolved after this step.

While our implementation is able to achieve a score within the ballpark of the reference solution's score on both the dev and the test sets, we observed several incorrect coreferences. Despite Hobbs' algorithm and pronominal coreference using gender, number and speaker (as discussed above), we observed that the

performance with pronouns was usually bad (e.g. “China” and “his”, and “Michael” and “its” were clustered together. Perhaps, a mapping from common pronoun types to their corresponding NER tags might have helped here. Similarly, nominal coreferences were better resolved in the dev set than the test set, which seems to suggest the training data to be highly representative of the dev set. A larger corpus (such as Wikipedia articles) might have helped to address this disparity.

Classifier-based coreference resolution

In this section, we discuss the features we added to our statistical classifier and the effect of each. The classifier extracts these features from a set of examples of coreferent pairs (both positive and negative), and then assigns weights to these features during training, which can then be used to classify new examples in the dev/test sets as coreferent or not coreferent. We used features that were given as examples in the assignment handout, features from Soon et al. described in the introductory video, as well as features that we came up with ourselves.

These are the 15 features we used, with some motivation of why they may be useful:

- Exact string match: If the candidate and fixed mentions match exactly, they are very likely to be coreferent.
- Distance in mentions and distance in sentences: Smaller distance is more likely to indicate coreference since mentions of the same entity probably occur close together.
- Whether the candidate/fixed mention is a pronoun: Since it is often the case that an entity is mentioned first not as a pronoun, and later on as a pronoun as a shorthand, the candidate is probably more likely to be coreferent if it isn’t a pronoun, and the fixed mention is more likely to be coreferent if it is a pronoun.
- Whether gender matches: Coreferent mentions should be compatible in gender (note that there are 3 possible cases: the feature returns 0, 1, 2 for at least one of the mentions doesn’t have a gender, gender matches, and gender does not match, respectively).
- Whether number matches: Coreferent mentions should be compatible in number (singular or plural). Like gender, there are 3 possible cases.
- Whether the candidate is a definite noun/demonstrative noun: These features are added in just to see if there are statistical trends when the candidate is a certain type of expression.
- Whether both the candidate and fixed mentions are proper nouns: The mentions should be more likely to be coreferent if they are both proper nouns.
- Whether NER tags match: If the headwords of the candidate and fixed mentions have the same NER tags, they are likely to be coreferent.
- Whether headwords match: If the headwords of the candidate and fixed mentions match, they are very likely to be coreferent.
- Whether part of speech matches: If the head tokens of the candidate and fixed mentions have the same part of speech, they are more likely to be coreferent.
- Whether both the candidate and mention head tokens are both nouns: This is a more specific case of the previous feature, and is useful since nouns are one of the most common types of POS for mentions.
- Whether the lemmas of the head tokens match: It is likely that there are cases where the head tokens are very similar, but don’t match exactly, and comparing the lemmas instead of the whole words will find additional coreferent mentions.

One immediate obstacle that we observed was that both MUC and B^3 F1 scores decreased slightly and stayed consistently below baseline (only feature is ExactMatch) up to adding about the first 10 features, and then every additional feature added consistently improved the F1 scores. One possible explanation for this is that ExactMatch has incredibly high precision, and initially adding features that are weak indicators of coreference drastically brings down this precision, which in turn brings down F1; however, once you have enough features, they collectively bring recall (and precision) back up. Therefore, when we evaluate the usefulness of each feature, it is not effective to simply measure the improvement from baseline with each additional feature added. Instead, we remove each feature from the total set of features, and measure the difference between our final results and the F1 scores of the new feature set (positive difference means the feature improved the classifier). Here are the results on the dev set (in the format feature name: MUC F1 improvement, B^3 F1 improvement): ExactMatch: -.003, -.002; DistanceInMentions: .029, .010; DistanceInSentences: .002, -.001; IsCandidatePronoun: .033, .020; IsFixedPronoun: .013, .002; GenderMatches: .012, .007; NumberMatches: .029, .010; IsCandidateDefiniteNoun: -.001, -.004; IsCandidateDemonstrativeNoun: -.002, -.003; BothProper: .000, -.002; NER: .033, .018; Headwords: .002, -.001; POS: .013, .004; BothNoun: .027, .016; Lemma: .023, .014. Most useful features were distance in mentions, whether candidate mention was a pronoun, whether number matches, whether NER tags matched, whether both mentions were nouns, and whether lemmas of head tokens matched. None of these were particularly surprising, but what was surprising were some of the features that turned out to be not particularly useful, and even contributed negatively. As a couple examples, even though the magnitude of the difference was very small, it seems strange that ExactMatch contributed negatively, comparing headwords had almost no contribution, and checking whether gender was compatible ended up contributing less than expected (relative to checking whether number was compatible). Pronominal resolution continues to be a concern with the classifier-based system as well. A few examples of this are: “I”, “me”, “the roadblock” and “it” are all marked as coreferent, and so are “I” and “the car”. One solution is to include a feature that checks compatibility of human/inanimate mentions. Like before, because we do not map different noun types with the pronouns that they can take, a mapping of pronouns to NER tags might help. Another error was marking “I” and “the copyright owner” as coreferent when “he” was the gold mention. This type of error could be handled by Hobbs’ algorithm, and thus adding features to our system that incorporate Hobbs’ would probably improve pronominal resolution issues. Another example was not identifying “fire” and “bonfire” as coreferent – our classifier system doesn’t currently have a way to capture this – the closest feature is comparing lemmas of head tokens; instead we could also try similar ideas like edit distance. In general, this model does a good job with proper nouns and named entities (e.g. the team and the national team are marked coreferent).

Results

DEV SET	MUC Precision	MUC Recall	MUC F1	B^3 Precision	B^3 Recall	B^3 F1
All Singleton	1.0	0.0	0.0	1.0	0.247	0.396
One cluster	0.770	1.0	0.870	0.165	1.0	0.282
Better baseline	0.783	0.647	0.709	0.721	0.592	0.650
Rule-based	0.806	0.782	0.794	0.621	0.740	0.676
Classifier-based	0.831	0.705	0.763	0.790	0.598	0.681

TEST SET	MUC Precision	MUC Recall	MUC F1	B^3 Precision	B^3 Recall	B^3 1

All singleton	1.0	0.0	0.0	1.0	0.273	0.429
One cluster	0.744	1.0	0.853	0.127	1.0	0.225
Better baseline	0.781	0.543	0.641	0.804	0.550	0.653
Rule-based	0.797	0.738	0.767	0.683	0.720	0.701
Classifier-based	0.829	0.631	0.717	0.842	0.601	0.702

A discussion of the multi-pass sieve for correference resolution (Ragunathan et al.) (extra credit)

In the past, most coreferent resolution models have determined coreference using a single function over a set of constraints or features. This, however, results in less precise rules/features overwhelming the more precise ones, resulting in poor scores, in addition to the limited information that we have to rely on at each stage. In their work, Ragunathan et al., proposed and implemented a multi-pass sieve model that apply these rules one at a time from highest to lowest precision, each building on the previous output. The model also propagates global information by sharing attributes collected from the previous passes, such as gender and number across the same cluster. By doing so, we are able to ensure that stronger features are given more weight and that we utilize all of the available information at the time. Despite a simple framework, the model was available to outperform several state-of-the-art systems on various corpora. What is also interesting is that all of these rules are deterministic and unsupervised, requiring no training on the gold set.

Our implementation is heavily inspired by this work. In particular, passes1, 3, 4, 5, 6 and 7 in their paper correspond to passes 1, 3, 4, 5, 6 and 9 (exact matching, strict head matching, the two variants of strict head matching, relaxed head matching and pronominal resolution) in our implementation. While the authors use a variety of data for different constraints, we rely on the Stanford POS and NER taggers. Despite this, the general trends in performance are similar across our implementations. For instance, strict head matching and its variants result in a solid boost in performance, and so does pronominal resolution. Relaxed head matching, on the other hand, gives minimal improvement (to the point that we decided not to use it in our system).

While there are similarities between our implementations, there are also some important differences. Unlike Ragunathan et al., we use statistical data collected during the training phase to predict coreference based on head word coreference (pass2). While this results in some discrepancies based on the data we're working with, we observed a solid boost in performance and decided to retain this as a rule. Using multiple sources or a big corpora of English sentences as training set might help. We also implement the Hobbs' algorithm for pronominal coreference. While Ragunathan et al. do not explicitly use this as a separate pass, they use ideas from the algorithm in the first step (mention processing) in their work.

Overall, we confirmed through our implementations of the rule-based and classifier-based systems that the former approach is equally good, if not better (as evident through our higher MUC F1 score), than the latter, thereby revalidating the claim made in this work.