

Chord Prediction Using Regression Analysis

Dimuth Kulasinghe (dimuthk)

1 Introduction

Chords are an essential construct in tonal music. They can be ambiguous, however, and whether a certain chord exists at a particular time is often up to human interpretation. Nevertheless, there are a multitude of factors which can help in building automated prediction models which emulate human-annotated progression markings with high precision.

2 Methodology

The `RuleBased` model doesn't take into account training data, and instead subjects each `Mention` pair in a document to various user-designed boolean rules, and marks them based on their performance.

2.1 Implementation

Our final design was a single-pass model that, for each `Mention` in a document, added it to an existing `ClusteredMention` if possible, and otherwise marked it as a new `ClusteredMention`. It determines compatibility with an existing `ClusteredMention` if it passes at least one rule for at least one `Mention` from the cluster. The same ruleset is applied to every `Mention` pair. Certain rules don't apply to a given pair; in this case, the rule is automatically failed.

The rules were thus designed such as to minimize the chance of a false positive while collectively achieving high recall. We composed the following rules:

- **PronounSimilarity:** This rule applies when both mentions are pronouns. We consider certain pairs of pronouns, such as *I*, *me* or *us*, *we* as highly likely to be coreferent, and pass them if they are within our composed list. We wrote 17 possible pronoun pairs altogether.
- **MatchingHead:** Given any two mentions, this rule passes if the head words of the sentences, according to their parse trees, are the same (case insensitive).
- **GenderSpecific:** This rule applies if one of the mentions is a pronoun referencing a gender, such as *he*, *she*. If the other mention contains a noun strongly inferring a specific gender, such as *brother* when the first mention is *he*, then the rule passes. We created a "hitlist" of 8 nouns for each gender.

- **PersonReference**: Similar to **GenderSpecific**. If the head word of the first mention is a gender-specific pronoun, and the head word of the second mention has the NER tag **PERSON** and POS tag **NNP**, then the rule passes.
- **SameText**: If the two mentions are an exact match (case sensitive), then the rule passes.

2.2 Results

Rule-Based	MUC			B3		
	Precision	Recall	F1	Precision	Recall	F1
Development	0.809	0.706	0.755	0.760	0.651	0.702
Test	0.805	0.678	0.736	0.775	0.658	0.712

The **RuleBased** model was much more accurate than the previous baseline models. In accordance with our design strategy, we got high precision, and reasonable recall values.

2.3 Alternatives

We tried improving the recall values further by introducing more rules. Other rules included detecting plural nouns and respective plural pronouns such as *they*, *them*, or recognizing when two mentions shared the same proper noun, or recognizing when they shared a certain threshold percentage of similar words.

More Rules	MUC			B3		
	Precision	Recall	F1	Precision	Recall	F1
Development	0.768	0.732	0.750	0.640	0.672	0.656
Test	0.716	0.706	0.711	0.658	0.712	0.603

While incorporating these rules boosted the recall values, they compromised the precision values, resulting in lower F1 scores overall. We also tried implementing a multi-pass model. The intuition was that each pass could incorporate a different rule, so that early passes could utilize our original rule set, which would keep precision relatively consistent while later passes boosted recall values using less precise methods.

Multi-Pass	MUC			B3		
	Precision	Recall	F1	Precision	Recall	F1
Development	0.784	0.675	0.723	0.780	0.570	0.659
Test	0.763	0.651	0.702	0.792	0.611	0.690

While effective, the multi-pass model failed to achieve the F1 scores of the single-pass model. We assume this is due in part to the fact that our rules were designed to work collectively, and each mention assignment is permanent; the assignment of a mention is less likely to be accurate given only one of the rules.

3 Classifier-Based

The **ClassifierBased** model is trained on a set of **Mention** pairs that are tagged as coreferent or non-coreferent. By recognizing the usefulness of user-designated features in the training pairs that may indicative of coreference, the model uses these features to determine the coreference of **Mention** pairs in test documents.

3.1 Implementation

We found the **ClassifierBased** model to work most effectively when given a few simple features. All features were used independently (no pairs). We used the following features in our final implementation:

- **CaseInsensitiveMatch**: Simple boolean feature that returns true if the mentions are identical (case insensitive).
- **PronounSimilarity**: Same as the rule from **RuleBased** model. It returns true if both mentions are pronouns, and are pairs from the hitlist.
- **MatchingHead**: Same as the rule from **RuleBased** model. It returns true if the head word from both mentions are the same (case insensitive).

3.2 Result

Classifier-Based	MUC			B3		
	Precision	Recall	F1	Precision	Recall	F1
Development	0.821	0.659	0.731	0.824	0.583	0.683
Test	0.824	0.615	0.704	0.862	0.586	0.697

The model returned F1 scores comparable to the **RuleBased** model given the simplicity and small number of features. The recall values were noticeably lower.

3.3 Alternatives

We tried several different features and sets of features to boost the recall values without compromising the high precision. This included distance rules in terms of sentence/mention index, head word NER similarity, head word POS similarity, and existence of proper nouns, all which resulted in lower F1 scores due to lowered precision.

A common problem with our model was an inability to detect gender pronouns; a name, for example, failed to be coreferred to *him*, *her*. We tried implementing features similar to the gender-specific ones implemented in the **RuleBased** model, but they lowered the precision values significantly.

4 Conclusion

We found the **HeadMatching** and **SimilarPronoun** rules/features to be by far the most useful; they independently achieved precision/recall values of around 0.85/0.6. We were surprised that proper noun-matching wasn't very useful; many of the names were found to be used in different contexts, and lowered our system precision. Both our implementations focused on retrieving as many simple coreference pairs as possible; we found that trying to take into account more complicated or ambiguous coreference pairs (such as mentions with the pronoun *it*) simply detracted from the performance of the model, since these pairs tended to hurt precision in recognizing the simpler pairs.

The **RuleBased** model was more effective as a whole; since we were able to designate the exact control flows of our rules and appliances, we could fine-tune the impact of each without having to worry about weights assigned by the system. We had trouble incorporating gender phenomena into the **ClassifierBased** model, for example, because of the assymetric nature of the model (given a candidate and principle mention pair), while our **RuleBased** model was able to disregard this altogether and treat the mentions identically. We found the training aspect of the **ClassifierBased** model unnecessary altogether; the feature weighing process was generally less effective than simply constructing rules by hand. We believe the **ClassifierBased** system may be more effective when analyzing general datasets though, or in a situation where the rule-maker is unfamiliar with the intricacies of the given dataset (a foreign language, for example).

Though our multi-pass **RuleBased** model was not as effective, we believe it is capable of being the most power coreferencer of the ones we worked with. By incorporating rules in descending generality, we probably could, given more time, find specific rules that take into account the more allusive coreference pairs, and increase recall value/F1 scores overall.

Ashwin created the Baseline and **ClassifierBased** models, and Dimuth created the **RuleBased** model.