A Comparative Study of Rule Based and Classifier Based Approaches for Anaphora Resolution

Aashwin Vaish, Chayan Kochar, Suyash Vardhan Mathur
{aashwin.vaish, chayan.kochar, suyash.mathur}@research.iiit.ac.in

INTRODUCTION

In linguistics, Anaphora is the phenomenon of referring to a topic or entity encountered earlier in discourse. Resolving anaphora is a key part of NLP applications.

In this project we will be comparing CPG based rules with a decision tree classifier on their individual accuracies for resolving anaphora in news dataset. Further, we will be experimenting with a *Hybrid Approach*, using the classifier to resolve the anaphora that the rule based model is unable to.

RELATED WORK

Various syntax-based approaches have been used previously for anaphora resolution, including Hobb's Algorithm, Lapin and Leass Algorithm. Most of the earlier works used phrase-structure parse as a source of syntactic information. However for free word ordered languages like Hindi, dependency based approaches are more suitable. This use of dependency based approach has been explored in [1] , [2].

DATA

In this project for anaphora resolution, we have used the data from the Hindi Dependency Treebank [1](HDTB) . In this corpus the dependency annotation is based on the Computational Paninian Grammar framework. Thus they represent syntactico-semantic relation with the notion of 'Karakas'.

The whole data is annotated in Shakti-Standard-Format (SSF) (as per [3]) . We have used the coreference annotated data, annotated on intra-chunk level. It also had annotations for animacy along with the other SSF annotations for gender, number, person, chunks, etc.

Data from the same corpus that was annotated for NE was used to create a bag of words for named entities, that was used as a semantic feature in the project.

EXPERIMENT

For our approach, we focus on resolving Entity(Concrete) Pronouns. We created a rule-based module that made use of dependency relations, syntactic information such as gender, number, person, and semantic information such as named entity categories to resolve simple anaphoras. We made use of a SSF API to extract these information from the annotated data. The

[1] http://cdn.iiit.ac.in/cdn/ltrc.iiit.ac.in/treebank_H2014/HDTB_pre_release_version-0.05.zip

pronouns were divided into 6 categories based on the rules that could be applied to a particular category.

Reflexives: Reflexive pronouns include pronouns such as 'अपना', 'स्वयं', 'खुद', etc. We know from Chomsky's Government and Binding Theory that the referent for a reflexive pronoun is the 'SUBJECT' of the clause or sentence, which corresponds to the *k1(Karta)* label of the CPG framework. Thus, reflexives can be resolved by selecting the noun phrase with dependency label *k1* within the same clause as reflexive pronoun. Therefore, we move up a dependency tree to find the root of the clause (verb node), and pass through all its children to find the one labelled with *k1* label, and return it as the referent of the reflexive pronoun.

Relatives: Relative pronouns in Hindi refer to the noun that is being modified by a relative clause (that the relative pronoun is a part of), and this relative clause is attached to the modified noun by the dependency label '*nmod-relc'*. Thus, this relativised noun is the referent of the relative pronoun (in the relative clause). Therefore, for relative pronouns, we move up the dependency tree from the pronoun, and select the parent NP of the clause that is attached to that parent by the relation '*nmod-relc'*.

First Person: First person pronouns generally refer to the speaker of a conversation. These include मैं, हम, and their inflected forms like मेरा, हमारा, etc. Since they refer to the speaker of the Narrative or Attributional clauses, these refer to the *k1 (karta)* of the main clause. Therefore, we go up the dependency tree until we find the complementizer कि, which marks the ending of the main clause, and go up in the dependency tree within the main clause (which ends at कि) until we reach the root (verb) node. Now we select the constituent that is attached to the root by the relation *k1 (karta)* as the referent.

Second Person: They refer to the listener of the utterance and include pronouns such as तू, तुम, आप and their inflected forms. Since they refer to the listener of the Narrative or Attributional clauses, these refer to the *k4 (sampradan)* of the main clause. Similar to First person pronouns, we go up the dependency tree until we find the complementizer कि, which marks the ending of the main clause, and go up in the dependency tree within the main clause (which ends at कि) until we reach the root (verb) node. Now we select the constituent that is attached to the root by the relation *k4 (sampradan)* as the referent.

Third Person: They are used more often than first-person and second-person pronouns because they refer to persons, places, or things that are not the reader or the writer. Similar observations were made by us that the instances of third person pronouns were much higher than others. These include 'यह', 'वह', 'वे', 'इनसे', 'उन्होंनें', etc - and are divided into two categories: Proximal and Distal.
As mentioned afterwards in 'Challenges', The Proximal Pronouns weren't referring to concrete entities in most of the cases, but were rather being used to refer to events and

for deixis. Thus, because of their rather low count as Entity Pronouns, they have been removed from Analysis.
Thus for this project we have excluded the Proximal third person pronouns and are checking just for Distals.
As they can refer to any object, we used the saliency order *k1>r6>k2>k4>k3*. Once such a node is found, it is sent for matching of gender, number, person.
We also thought of taking into consideration the property of animacy, but many pronouns and their correct referent did not have the same animacy feature in the dataset.

Locatives: These pronouns are used to refer to spaces, and Hindi has two such pronouns - यहां and वहां and their inflected forms.
Using the 'Location' tag as the NER category, if the nearest noun phrase has this feature, it can be considered as the referent. Earlier, we were also using animacy features, but it became redundant when the NER was applied. In the CPG Framework, we resolved the Locative Pronoun when a mention(nearest NP) has 'k2p' or 'k7p' as the dependency relation with the main verb.

In agreement with the observation regarding locatives in [2], we also observed that if there was another locative pronoun occurring before the current one, that itself was many times the referent of the current locative pronoun.

Classifier: A decision tree classifier was used from the sci-kit learn python library. Positive training instances were created by pairing the pronoun with the preceding heads of the Noun Phrases which were coreference entities in its coreference chain, while negative training instances consisted of the pronoun and all previous NP heads which were not the members of the coreference chain.
The features that were used for training the Classifier include:

1. **Number of Pronoun and NP:** Singular and Plural were given 1 and 2 values, while those which were unlabelled or were *any* were given 0 value.
2. **Chunk Distance Between Pronoun and NP:** This feature uses the number of chunks between the candidate NP and the current Pronoun
3. **Sentence Distance Between Pronoun and NP:** This feature uses the number of sentences between the candidate NP and the current Pronoun
4. **NE category of NP:** This feature considers the Named Entity categories which were given values : Person: 1, Location: 2, Organisation: 3, Number: -1, Measure: -2, Time: -3, None: 0.
5. **Pronoun itself:** A list of pronouns in the text was made, and they were given corresponding numbers based on which pronoun was being used.
6. **Dependency relation tag of NP:** Dependency tags of NP was also added as a feature by assigning values to the dependency tags that are used with NPs.
7. **Pronoun Category:** The category of the Pronoun in Reflexive, Locative,

First Person, Second Person, Third Person and Relative were used for this feature.

The code for this project is hosted on github[2].

OBSERVATIONS

RESULTS / ANALYSIS

Rule-Based:

Following the above methods, we got decent accuracy for our rule-based approach.

Type	Total	Correct	Unidentified	Accuracy
FP-SP	16	12	4	75.000%
Third Person	226	91	0	40.265%
Reflexive	84	55	24	65.476%
Relative	50	35	1	70.000%
Locative	64	18	5	28.125%
Total	440	211	34	47.955%

We can see that the system gives quite decent accuracy for all pronouns except Locatives and third person pronouns. This is because Locatives might not always be marked for NEs or k7p, k2p labels.

Classifier-Based Approach:

We divided the dataset into 2:1 ratio on the basis of documents for training and testing respectively.

Training Pronouns : 1143
Testing Pronouns : 440
Correct: 150
Incorrect: 290
Accuracy: 34.090%

The Classifier, with the limited number of features and the limited training data available, gave a decent accuracy.

Hybrid Approach:

The rule based approach was observed to give accuracy better than classifier-based itself. However, there were some referents which were out of bounds for the range that was chosen. These were decided to be passed to the Classifier, thus creating a Hybrid model where the Pronouns that were unresolved by the Rule-Based module were resolved by the classifier. This gave a boost of 2.727% to the overall resolution of the Rule-based system. Thus, our Hybrid resolution approach gives an accuracy of **50.682.**

The number of unresolved Pronouns by the Rule-Based system, which were then passed to the Classifier, along with their accuracy are given in Table 3.

Type	Total	Correct	Accuracy
FP-SP	16	14	87.5000%
Third Person	226	91	40.265%
Relative	50	35	70.000%
Reflexive	84	65	77.381%
Locative	64	18	28.125%
Total	440	223	50.682%

Table 3 : Accuracy of decision-tree class

[2] https://github.com/av-dx/cl2-project

CHALLENGES FACED

Our main challenge faced in the rule based was regarding third person pronouns and locatives as discussed earlier.

- Many times, the pronoun is not actually referring to a discourse object. There is a use of deixis.
- Though we have considered only entity-anaphora resolution, as talked out about later, certain pronouns like 'इस', 'उस', 'जिस', 'जिसके', etc. referred to both entity and events. Our system was not able to successfully recognize event cases. Hence, it hampered the accuracy.
- The proximal third person pronouns was hit badly with not so good results as compared to Distals.
- Our main challenge for locatives was NER necessity. Though we implemented a new way (as described above) for it.
- The annotations in the data collected were a bit irregular, and incorrect at places. Thus many times even though the system has predicted the answer acceptable by us, according to the data, the referent was not present in the coreference chain.
- There is a possibility to detect more "wrong resolution" cases at the rule level itself, so it could be sent to the classifier.
- Of course not to forget, the ambiguities were always there, and though we tried our best, it is still not fully explored, as we would have then had to look into context and other details.

CONCLUSION

The heuristic approach gave good results for Anaphora Resolutions, with accuracy ~48%. While the accuracy of decision tree classifier was comparatively lower than the rule-based module when tested on its own, adding it to the resolve the unresolved cases after the rule based resolution helped in improving the accuracy of the heuristic approach further in the form of **Hybrid Approach**. The Hybrid Approach gave a better accuracy (~51%), and given the very few number of unresolved cases present in the given dataset (34) that were passed to the classifier, it showed a decent improvement of 2.727% in accuracy over the Rule based approach. Hybrid Approach was the only suitable way for resolving these unresolved cases, because looking in the discourse further than 3 sentences behind would be Computationally expensive, and not possible for practical usage.

REFERENCES

[1] P. Dakwale, V. Mujadia, and D. M. Sharma, "A hybrid approach for anaphora resolution in Hindi," in Proceedings of the Sixth International Joint Conference on Natural Language Processing, 2013, pp. 977–981.

[2] V. Mujadia, D. Agarwal, R. Mamidi, and D. M. Sharma, "Paninian grammar based hindi dialogue anaphora resolution," in Asian Language Processing (IALP), 2015 International Conference on, 2015, pp. 53-56.

[3] A. Bharati, R. Sangal, and D. M. Sharma, "Ssf: Shakti standard format guide," Language Technologies Research Centre, International Institute of Information Technology, Hyderabad, India, 2007, pp. 1–25.