Anaphora Resolution for Bengali, Hindi, and Tamil Using RandomTree Algorithm in Weka

Sanjay Chatterji,Arnab Dhar,Biswanath Barik,Moumita PK,Sudeshna Sarkar,Anupam Basu
Department of Computer Sc. & Engineering, Indian Institute of Technology, Kharagpur, India
e-mail: {sanjaychatter, arnab832007, bn.barik, mpkcel, shudeshna, anupambas}@gmail.com

Abstract

In this paper, we discuss a data driven approach for Anaphora resolution of three Indian languages: Bengali, Hindi, and Tamil. The work consists of two steps: identifying markables and links. Markable identification is done using Conditional Random Field. The identifications of links between markables is done using Decision Tree Algorithm. Both the steps are evaluated and shown results in terms of F-Value.

1 Introduction

Anaphora Resolution is a process of automatically finding the pairs of pronouns or noun phrases in a text that refer to same incidence, thing, person, etc. called referent. The first member of the pair is called antecedent and the next member is called anaphora. Antecedents and anaphoras can co-occur in the same sentence or can be in different sentences. Anaphora Resolution can be done in two steps. First, the pronouns or noun phrases that may be used for indicating a referent are identified. These are called markables. Then, the pairs of such markables that have the $antecedent - anaphora$ relation are identified. The identification of such relations is used in Document Summarization, Machine Translation, Information Extraction, etc.

Anaphora Resolution is a challenging task particularly for resource poor languages like Indian Languages. We have worked on three Indian languages: Bengali, Hindi, and Tamil. All these three languages have $Subject{-}Object{-}Verb$ sentence structure. They have flexible word order. The statistical data used in this work is released by ICON tool contest, 2011. They are from literature domain. Our objective is to identify the antecedent-anaphora pairs of a story.

Co-referential chain is a chain of markables that indicate same referent. Instead of identifying each pair of a chain separately, we wish to find the pairs which come one after another. For the computational simplicity another constraint is added that only the pronouns can be used as anaphora. Consider the following sentences.
"Shahrukh Khan is in tension these days because of the media showing his relations with the founder of Cineyug, who is in jail for his involvement in 2G scam. The actor clarifies that he has nothing to do with it, and the media should not drag him into the matter."
One Co-referential lexical chain of these sentences is: (Shahrukh Khan→his→The actor→he→him)
Putting constraint we get: (Shahrukh Khan → his), and (The actor → he → him)
We wish to identify the pairs (Shahrukh Khan, his), (The actor, he), and (he, him) of the chains.

2 Related Works

Considerable amount of research work has been carried out by many researchers in anaphora resolution related problems. Among them, Pronoun anaphora resolution for various languages is an attractive problem. The work of (Soon et al., 2001) is one of the pioneering works. Many researchers have started working on Indian languages too. Bengali have relatively limited work in this direction. (Dhar and Garain, 2008) have done some work for Bengali. There are many works for Tamil language (Sobha, 2007), (Pattabhi et al., 2007), (Sobha et al., 2009). (Sobha and Patnaik, 2000) have developed an Anaphora resolution system for

Hindi and Malayalam languages.

Waikato Environment for Knowledge Analysis $(Weka)$ (Witten et al., 1999) (Eibe et al., 1999) is a java implementation of various machine learning algorithms such as Decision Tree, Support Vector Machines, Instance-based Classifiers, Bayes Decision Schemes, Neural Networks, etc. (Müller et al., 2002) shown that the J48 decision trees, which are a Weka re-implementation of C4.5 can be used for building a classifier for reference resolution. The MATE guideline of (Poesio., 2004) can be used to do most of the markable identification task automatically if there is syntactic information in the training data.

CRF++: Yet Another CRF++ toolkit (Lafferty, 2001) is a simple, customizable, and open source implementation of Conditional Random Fields (CRFs). It can perform single-best MIRA training and supports semi-Markov CRF.

3 Description of the Proposed Approach

We have developed a two stage pronoun reference resolution system for Bengali, Hindi and Tamil. In the first stage we have used the Bengali Markable annotation data as training data. We have used Conditional Random Field for segmenting and labeling sequential markable data. In other words, Conditional Random firld is used to to find the boundary of a markable item. According to our observation it may be a single noun chunk or a group of noun, counjunct or adjectival chunk.

In the second stage we have used Markable annotated Training data of both the three languages for training three models. Each model is used to test the corresponding markable annotated test files. We have used Decision Tree algorithm for statistically identifying the pairs of markables that have the antecedent-anaphora relation. A $K-Fold$ decision tree algorithm, called RandomTree, is used here for constructing trees at each node by considering R random features. The best result is observed for $K-Fold=2$. We have used RandomTree implementation of Weka to test our anaphora annotations.

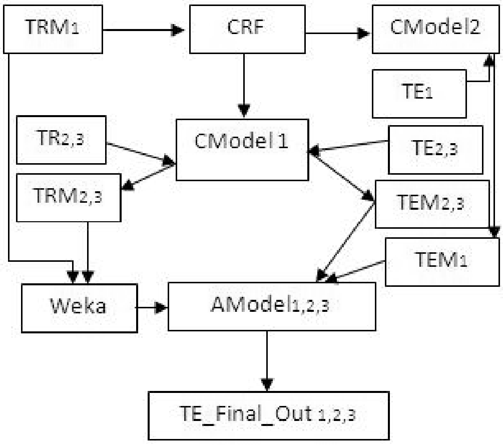

Figure 1: System Architecture
TR: Train Data, TRM: Train Data with Markables
TE: Test Data, TEM: Test Data with Markables
Suffix 1: Bengali, 2: Hindi, 3: Tamil
AModel: Anaphora Model, CModel⋆: CRF Models
TE_ Final_Out: Anaphora Output for Test Data

3.1 System Architecture

In Figure 1, the architecture of the proposed scheme is shown. Using Bengali markable training data(TRM_1) we have created a CRF model$(CModel2)$ for Bengali raw test data and a CRF model$(CModel1)$ for Hindi and Tamil data. Hindi and Tamil training data $(TR_{2,3})$ is annotated by markable tag in $(TRM_{2,3})$. Similarly, Hindi and Tamil test data $(TE_{2,3})$ is annotated by markable tag in $(TEM_{2,3})$. At the same time, Bengali test data (TE_1) is annotated by markable tag in (TEM_1). In the next step, Bengali markable annotated training data (TRM_1) and Hindi and Tamil markable annotated training data $(TRM_{2,3})$ is used to build the anaphora models for three different languages$(AModel_{1,2,3})$. Finally, each language model is used to test the corresponding language test markable data. And, final output$(TE_Final_Out_{1,2,3})$ is the anaphora annotated markable test data. During, these steps we need to convert the format of the data and those steps are not shown in this architecture to avoid the complicacy.

3.2 Data Format

The data used for training the system was provided in ICON, 2011 NLP Tool Contest. The annotation data uses CONLL format. In this format, there is one word per line, with extra attributes of a word (POS tag, chunk, Named Entity tag) separated by tab. Sentences are separated by a blank line. However, there are also sentence end markers to indicate the end of the sentence.

Each word has a separate POS tag. But, chunks or named entity tags may take more than one words. These tags are in BIO format, in which a $B - XXX$ tag indicates the first word of a tag type XXX and $I - XXX$ is used for subsequent words of that tag type. The tag O indicates the word is outside of the tag. However, as all the words must be in any one chunk tagset, the concept of O is not valid for chunk annotation. Each word also has a serial number.

A pronoun of a sentence can refer to one referent of another sentence. But, both the sentences must be in the same story. Keeping this in mind, story name is added before each line of the training and test data.

The training data for Bengali, Hindi and Tamil contains 17.7K, 30K and 69K words respectively. The corresponding test data for these languages are 7.5K, 15K and 25K.

3.3 CRF Models and features

Many of the researchers (Soon et al., 2001) (Poesio and Artstein, 2008) have considered all the noun phrases, nested noun phrases, and named entities as markables. For Indian languages, where chunks are used instead of phrases, considering all Noun chunks as markable is not always correct. There are many exceptions in both the two cases. There are many noun chunks that are not markables. There are many markables that are not noun chunks. Consider the following Bengali example in ITrans format.

bombAi dilli kimbA mAdrAja yekhAnei tini thAkuna tAte AmAra kichhu karAra nei.
Bombay Delhi or Madras wherever he stays that my anything do not
(Bombay Delhi or Madras wherever he stays I can not do anything.)

Here, “tAte” is a noun chunk(NP) but not markable. The conjunct chunk(CCP) “kimbA” is also markable. The phrase “bombAi dilli kimbA mAdrAja” contains three noun chunks and a conjunct chunk. The whole phrase is considered as a single markable.

Therefore, We have used CRF models to build the Markable annotator system. The linguistic feature that is there with us are POS, Chunk, NE and the words itself. The only markable training data available is in Bengali. The feature set that gives best result for Bengali test data is selected experimentally. But, this feature set does not give best result for other language test data. Therefore, another set of features is created and used to find the markables of Hindi and Tamil sentences.

The features used for markable annotation are as follows.

Word feature: Previous m words $(w_{i-m}...w_{i-1})$ to next n words $(w_{i+1}...w_{i+n})$ are used as features.

Tag features: The POS, chunk, Named Entity tags of previous m words to next n words are used as features.

Root Information of Word: Indian languages are morphologically rich. Words are inected in various forms depending on its morphological features. Therefore, word feature can not help much. Therefore, the roots of previous m words to next n words are also used.

Morphological features: Morphological information (such as, gander, number, person, tense, aspect, etc) of previous m words to next n words are used as features.

For the last two features of the words, we have analyzed the words of Bengali sentences.

3.4 Features for RandomTree of Weka

For finding whether a pair of markables is referring the same referent or not we have used the decision tree algorithm implemented in Weka. The features used for this task are as follows.

Anaphora: Each of Bengali, Hindi, and Tamil has a closed set of pronouns. Each of the pronouns are treated as anaphora.

Antecedent: Antecedent can be any group of words that come one after another in a sentence. As Weka do not accept open set of strings, we have given an unique value for each markables.

Distance: Its value is the number of sentences between antecedent and anaphora. If antecedent and anaphora come in the same sentence then its value is 0.

Antecedent is Name/Pronoun: Its value is 1 if the antecedent is a Name, 2 if the antecedent is a Pronoun and otherwise 0.

4 Evaluation

Thee performance of the CRF++ tool on the Bengali data using various features are presented in Table 1 and for Hindi data are presented in Table 2. The performance is measures in terms of F-Value which is the weighted harmonic mean of precision and recall.

In all the cases we have considered $m = n$. This value is written in first bracket. So, experiment number 1 of Table 1 shows that the Word, Morph, POS, Chunk, NE of previous 2 words and next 2 words including the current word in unigram and bigram level is taken as features. Corresponding F-Value is 85.31.

Exp.	Feature	Overall F-value
	TR size 21K	
1.	W, R, M, POS, C, NE (≤ 2) Unigram and Bigram	85.31
2	W, R, M, POS, C, NE (≤ 2) Uni and Bi (and Tri for POS)	86.56
3.	W, R, M, POS, C, NE (≤ 2) Uni and Bi (and Tri for C)	87.23
4.	W, R, M, POS, C, NE (≤ 2) Uni and Bi (and Tri for W)	**87.70**
5.	W, R, M, POS, C, NE (≤ 2) Uni and Bi (and Tri for R)	87.27
6.	TR size 24K for Exp. 4	87.72
7.	TR size 26K for Exp. 4	**88.38**
8.	TR size 28K for Exp. 4	88.15

Table 1: Bengali Markable annotation F-values for different features

W:Word, R:Root, M:Morph, POS:Parts of Speech
C:Chunk, NE:Named Entity, TR: Training data
Uni:Unigram, Bi:Bigram, Tri:Trigram, Exp: Experiment

The best feature of the Table 1 is the experiment number 7. We have used this feature to build the model CModel2 of Figure 1. Similarly, The best($3rd$) feature of Table 2 is used to build the model CModel1 of Figure 1.

Feature	Overall F-value
POS, C, NE (≤ 2) Uni, Bi, and Tri	73.81
POS, C, NE (≤ 3) Uni, Bi, and Tri	**74.38**
POS, C, NE (≤ 4) Uni, Bi, and Tri	73.69

Table 2: Hindi Markable annotation F-values for different features

C:Chunk, NE:Named Entity
Uni:Unigram, Bi:Bigram, Tri:Trigram

For evaluating the Anaphora links we have used two methods. Firstly, we kept aside $1K$ word test data from different places of the training data. And, remaining part is used for training. For 5 such cases, the results are noted down. The average of the $F - values$ for Bengali, Hindi and Tamil are 55%, 39% and 44% respectively. In second way, we kept aside $1K$, $2K$, $3K$, $4K$ and $5K$ word test data for these three languages. The results for these words are shown in the figure 2, 3, and 4, respectively. Here, vertical axis gives $F - Value$ and Horizontal axis gives number of words in test data.

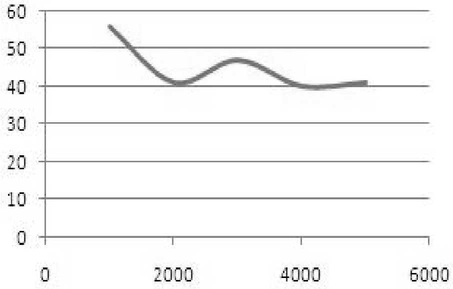

Figure 2: F-Value for Bengali words

The work is also evaluated in 5 other evaluation schemes: MUC, BCUB, CEAFM, CEAFE and BLANC [(Recasens and Hovy, 2011) (Luo, 2005) (Bagga and Baldwin, 1998) (Vilain et al., 1995)] as in Table 3. The average F-value of those evaluation results for Bengali, Tamil and Hindi are 66.87, 36.73 and 37.48, respectively.

Metric	MUC			BCUB			CEAFM			CEAFE			BLANC			
Langs	R	P	F1	R	P	F1	R	P	F1	R	P	F1	R	P	F1	Average F1
Bengali	45.2	27	33.88	96.66	70.39	81.46	86.29	72.73	78.93	73.2	77.67	75.37	63.92	65.58	64.74	66.87
Tamil	1.72	2.54	2.05	68.66	78.42	73.21	56.67	54.48	55.56	56.92	46.41	51.13	1.23	2.75	1.7	36.73
Hindi	1.74	2.46	2.04	68.06	76.98	72.24	58.94	55.82	57.34	60.21	48.22	53.55	1.56	4.04	2.25	37.48
Comb	18	15.83	16.85	80.58	73.5	76.88	68.51	61.81	64.99	62.89	57.48	60.06	20.87	36.58	26.58	49.07

Table 3: Anaphora Annotation F-values using 5 Standard Schemes.
Comb: Combined score for all the three language.
R:Recall, P:Precision, F1: F-Value

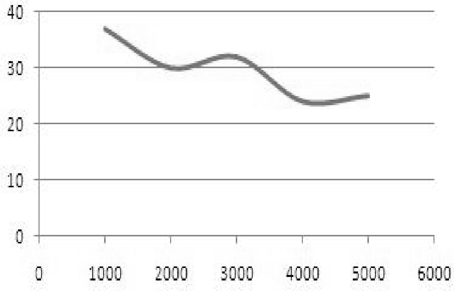

Figure 3: F-Value for Hindi words

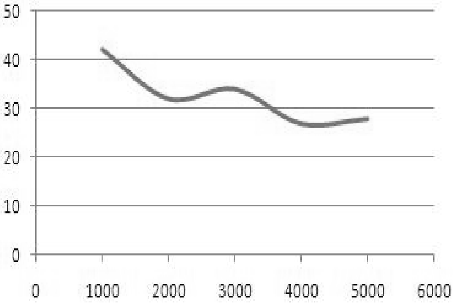

Figure 4: F-Value for Tamil words

5 Conclusion and Future Work

The work described in this paper is a two stage pronoun reference resolution system for Bengali, Hindi and Tamil. The effort we have given in markable annotation task is supposed to work well for a language where all NP chunks are not markable and also all markables are not NP chunks. Even, where there are many NP chunks that together form a markable, this statistical approach may be used.

However, more experiments are required to find the effect of statistical markable annotation task in anaphora resolution. The result shows that the system can be used as a baseline system. Further research may be started to find the effect of more features in Weka. Further experiment can also be done to see the effect of language dependent rules in the anaphora annotation task.

ACKNOWLEDGEMENT

This work is partially supported by The Bangla Treebank project sponsored by CIIL, Mysore and the ILMT project sponsored by TDIL program of MCIT, Govt of India. We would like to thank all the lab members in Communication Empowerment Lab, IIT Kharagpur, for their active participation in the development of the corpus.

References

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *In The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*, pages 563–566.

A. Dhar and U. Garain. 2008. A method for pronominal anaphora resolution in bengali. In *Proceedings of the 6th Int. Conf. on Natural Language Processing (ICON)*, Pune, India.

Ian Witten Eibe, Ian H. Witten, Eibe Frank, Len Trigg, Mark Hall, Geoffrey Holmes, and Sally Jo Cunningham. 1999. Weka: Practical machine learning tools and techniques with java implementations. In *Proc ICONIP/ ANZIIS/ANNES99 Future Directions for Intelligent Systems and Information Sciences*, pages 192–196. Morgan Kaufmann.

John Lafferty. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. pages 282–289. Morgan Kaufmann.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *In Proc. of HLT/EMNLP*, pages 25–32. URL.

Christoph Müller, Stefan Rapp, and Michael Strube. 2002. Applying co-training to reference resolution. In *ACL*, pages 352–359.

R K Rao Pattabhi, L. Sobha, and Amit Bagga. 2007. Multilingual cross-document co-referencing. In *Proceedings of 6th Discourse Anaphora and*

Anaphor Resolution Colloquium (DAARC), pages 115–119, Portugal.

Massimo Poesio and Ron Artstein. 2008. Anaphoric annotation in the arrau corpus. In *LREC*. European Language Resources Association.

M. Poesio. 2004. Design of the moses decoder for statistical machine translation. In *Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue*, pages 154–162, Morristown, NJ, USA.

Marta Recasens and Eduard Hovy. 2011. Blanc: Implementing the rand index for coreference evaluation. In *Natural Language Engineering*. URL.

L. Sobha and B.N. Patnaik. 2000. Vashisht: An anaphora resolution system for malayalam and hindi. In *International Conference ACIDCA2000*, Monastir,Tunisia.

L. Sobha, Sankar Kuppan, Kavitha Venkataswamy, and Pattabhi R.K. Rao. 2009. “identification of similar documents using coherent chunks“, anaphora processing and applications. In *Lecture Notes in Artificial Intelligence, Springer*, volume 5847, pages 54–68, Berlin/Heidelberg.

L. Sobha. 2007. Resolution of pronominals in tamil. In *Computing: Theory and Applications, 2007. ICCTA '07. International Conference on*, pages 475 – 479, march.

W. M. Soon, H. T. Ng, and D. C. Y. Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, pages 45–52. Morgan Kaufmann, California.

Ian H. Witten, Eibe Frank, Len Trigg, Mark Hall, Geoffrey Holmes, and Sally Jo Cunningham. 1999. Weka: Practical machine learning tools and techniques with java implementations.