Anaphora Resolution for Bengali: An Experiment with Domain Adaptation

Utpal Kumar Sikdar[1], Asif Ekbal[1], Sriparna Saha[1], Olga Uryupina[2], and Massimo Poesio[2]

[1]Department of Computer Science & Engineering, Indian Institute of Technology Patna, Patna, India

[2]Department of Information Engineering & Computer Science, University of Trento, Italy

{utpal.sikdar,asif,sriparna}@iitp.ac.in, uryupina@gmail.com, massimo.poesio@unitn.it

Abstract. In this paper we present our first attempt on anaphora resolution for a resource poor language, namely Bengali. We address the issue of adapting a state-of-the-art system, BART, which was originally developed for English. Overall performance of co-reference resolution greatly depends on the high accurate mention detectors. We develop a number of models based on the heuristics used as well as on the particular machine learning employed. Thereafter we perform a series of experiments for adapting BART for Bengali. Our evaluation shows, a language-dependant system (designed primarily for English) can achieve a good performance level when re-trained and tested on a new language with proper subsets of features. The system produces the recall, precision and F-measure values of 56.00%, 46.50% and 50.80%, respectively. The contribution of this work is two-fold, *viz.* (i). attempt to build a machine learning based anaphora resolution system for a resource-poor Indian language; and (ii). domain adaptation of a state-of-the-art English co-reference resolution system for Bengali, which has completely different orthography and characteristics.

Keywords. Anaphora/Co-reference resolution, CRF based mention detection, Bengali, BART.

Resolución de anáfora para el bengalí: un experimento con la aplicación al dominio

Resumen. Este artículo presenta el primer intento de resolución de anáfora para un idioma que tiene escasos recursos lingüísticos, específicamente el idioma bengalí, mediante la adaptación del sistema BART que pertenece al estado del arte y fue desarrollado originalmente para el inglés. El rendimiento general de resolución basada en co-referencias depende en gran medida de los detectores de menciones de alta precisión. Se desarrollaron unos modelos basándose en la heurística usada y en el método de aprendizaje de máquina seleccionado. Se hicieron unos experimentos para adaptar BART al idioma bengalí. La evaluación efectuada muestra que un sistema dependiente del idioma (diseñado principalmente para el inglés) puede lograr un buen rendimiento después de reentrenamiento y prueba, para el idioma nuevo usando conjuntos apropiados de características. El sistema produce los valores de recall, precisión y medida F iguales a 56.00

Palabras clave. Resolución de anáfora/co-referencia, detección de menciones basada en el campos aleatorios condicionales (CRF), bengalí, BART.

1 Introduction

Anaphora/co-reference resolution is the task of identifying noun phrases that are used to refer to the same entity in a text. More precisely, let us assume that C1 and C2 are occurrences of two noun phrases (NPs) and both have a unique referent in the context in which they occur. Here C2 refers to C1 in the context. C1 is called antecedent and C2 is called anaphora. The noun phrases that may participate in co-reference relation are called mentions/markables. Coreference information is needed for solving several natural language processing (NLP) application areas including Information Extraction [6], Text Summarization[19], Question Answering [7] etc. These practical tasks, for example, information extraction and text summarization, can be performed more reliably if it is possible to automatically find parts of the text containing information about a given topic. The summarization task will be helpful if a program can automatically spot all the clauses in the text that contain information about a given topic. Most of these works on supervised co-reference resolution have been developed for English ([17, 9, 26, 5]), due to the availability of large corpora such as ACE (Walker et al., 2006 [23]) and OntoNotes (Weischedel et al., 2008 [24]). BART, the Beautiful Anaphora Resolution Toolkit [21, 12, 11], is the resultant of the project titled ”Exploiting Lexical and Encyclopedic Resources For Entity Disambiguation” carried out at the Johns Hopkins Summer Workshop 2007. It can handle all the preprocessing tasks to perform automatic coreference resolution. A variety of machine learning approaches are used in BART; it mainly uses several machine learning toolkits, including WEKA and MaxEnt.

So far, work on anaphora resolution has covered English, a few other European languages such as Dutch, German, Italian, and Spanish, and few Asian languages including Arabic, Chinese and Japanese; but there is hardly any literature on anaphora resolution in Indian languages even though they include the fourth and sixth most spoken languages in the world, Hindi and Bengali. One reason is that India is a multilingual country with great linguistic and cultural diversities: it counts 22 different official languages from almost all the dominant linguistic families in the world, and not less than 900 languages overall. This huge diversity makes it very difficult to ensure the availability of linguistic resources (corpora, part-of-speech taggers, etc.) for all these languages. In 2011 a shared task on NLP Tools Contest on Anaphora Resolution in Indian Languages is organized in association with 9th International Conference on Natural Language Processing (ICON 2011)[1]. Four teams participated in this contest. But no results or none of these papers [1, 2, 14, 3] are available in the web. In terms of native speakers, Bengali ranks sixth in the world, *second* in India and *first* in Bangladesh. In this paper we attempt to adapt an existing state-of-the-art English anaphora resolution system for a resource poor language, namely Bengali that has a completely different orthography and characteristics.

Our new anaphora resolution system for Bengali was developed in two main steps. Firstly, we develop various models of mention detection systems for Bengali. The identified mentions are then used to produce markables for anaphora resolution. Thereafter we use BART as the underlying platform for detecting anaphoric chains. Decision tree based classifier of Weka machine learning toolkit embedded in BART is used as the classification technique. We experiment with the various subsets of features implemented in BART. The main goal was to come up with a set of features that could be more suitable for Bengali. For evaluation we used the data sets provided in the ICON NLP Tools Contest on Anaphora Resolution in Indian Languages [16]. For training and development datasets, annotations were provided by the organizers. But no annotation was provided for the test data. In line with the annotations of training and development datasets, we manually annotated test dataset. We develop a number of models for mention detection. The mention detector developed with the supervised classifier, conditional random field [4] performs best for the anaphora resolution. The best configuration was obtained based on the development data. We used the configuration that best suited the development set. The anaphora resolution system yields the recall, precision, and F-measure values of 56.00%, 46.50% and 50.80%, respectively.

The rest of the paper is organized as follows. Section II describes mention detection systems for Bengali that includes a brief introduction to CRF, features used for training the CRF and results of the various mention detection models. In Section III, we present our approach for anaphora resolution. Section IV reports the datasets, experiments conducted and the evaluation results of the anaphora resolution system. Finally, Section V concludes the paper.

2 Mention Detection for Bengali

Robust mention detection is an essential component of any anaphora resolution system. BART supports different pipelines for mention detection. The choice of a pipeline depends crucially on the availability of linguistic resources for a given language.

The first stage of the anaphora resolution process tries to identify the occurrence of mentions in the Bengali documents. All the noun phrases

[1] http://ltrc.iiit.ac.in/icon2011/contests.html

present in the corpus may participate in anaphoric relation; these are called mentions/markables. In the original data, three information were provided for each token: Part-of-Spech (PoS), Phrase (or, Chunk) information and Named Entity (NE) information. We develop the following mention detection models:

1. **First approach**: In our first approach we consider each noun phrase (i.e. NP) as a possible candidate of mention. Results of this model are shown in Table 1.

2. **Second approach**: In our second approach we consider each Named Entity (NE) or pronoun (i.e. PRO) as a mention and its results are shown in Table 1.

3. **Third approach**: In the third approach we take only person name (PER) or pronoun (PRO) as a candidate of mention. Results in Table 1 show a little improvement in performance for one document with this model, however the performance for other document decreases.

4. **Fourth approach**: Here we use a supervised classifier named conditional random field (CRF) to detect mentions from a given text. We formulate the mention detection as a classification problem by assigning each token in the text a label, indicating whether it is a mention or not. Hence to learn a classifier at first we have to create a training data and have to derive the class values (either B-mention/I-mention/Others[2]) of all the tokens from the annotated data. We created a training set for mention detection based on the mentions present in the original training data. Evaluation results in Table 1 support that this mention detection system is the best compared to the other three models. Details of this systems are mentioned in the following subsection.

2.1 Conditional Random Field based Mention Detection System

In this section we give a brief a introduction to our employed classifier, features used for training the classifier and the results of the different mention detectors.

[2]B, I and O denote the beginning, internal and outside of a mention entity.

2.1.1 Brief Introduction to CRF

Conditional Random Fields (CRFs) [4] are undirected graphical models, a special case of which corresponds to conditionally trained probabilistic finite state automata.

The conditional probability of a state sequence $s =< s_1, s_2, \ldots, s_T >$ given an observation sequence $o =< o_1, o_2, \ldots, o_T >$ is calculated as:

$$P_\wedge(s|o) = \frac{1}{Z_o} \exp(\sum_{t=1}^{T}\sum_{k=1}^{K} \lambda_k \times f_k(s_{t-1}, s_t, o, t)),$$

where, $f_k(s_{t-1}, s_t, o, t)$ is a feature function whose weight λ_k, is to be learned via training. The values of the feature functions may range between $-\infty, \ldots + \infty$, but typically they are binary. To make all conditional probabilities sum up to 1, we must calculate the normalization factor,

$$Z_o = \sum_{s} \exp(\sum_{t=1}^{T}\sum_{k=1}^{K} \lambda_k \times f_k(s_{t-1}, s_t, o, t)),$$

which as in HMMs, can be obtained efficiently by dynamic programming.

To train a CRF, the objective function to be maximized is the penalized log-likelihood of the state sequences given the observation sequences:

$$L_\wedge = \sum_{i=1}^{N} \log(P_\wedge(s^{(i)}|o^{(i)})) - \sum_{k=1}^{K} \frac{\lambda_k^2}{2\sigma^2},$$

where $\{< o^{(i)}, s^{(i)} >\}$ is the labeled training data. The second sum corresponds to a zero-mean, σ^2 -variance Gaussian prior over parameters, which facilitates optimization by making the likelihood surface strictly convex. Here, we set parameters λ to maximize the penalized log-likelihood using Limited-memory BFGS [15], a quasi-Newton method that is significantly more efficient, and which results in only minor changes in accuracy due to changes in λ.

When applying CRFs to the mention detection problem, an observation sequence is a token of a sentence or document of text and the state sequence is its corresponding label sequence. A feature function $f_k(s_{t-1}, s_t, o, t)$ has a value of 0 for most cases and is only set to be 1, when s_{t-1}, s_t are certain states and the observation has certain properties. We have used the C^{++} based CRF^{++} package [3].

[3]http://crfpp.sourceforge.net

Table 1. Results of different approaches for mention detection (we report percentages)

Sr.	Mentions	Datasets	precision	recall	F-measure
1	NP	DEV-1	09.14	99.21	16.74
		DEV-2	13.69	100.00	24.08
2	NE/PRO	DEV-1	14.29	18.90	16.27
		DEV-2	28.02	18.48	22.27
3	PER/PRO	DEV-1	22.81	10.24	14.13
		DEV-2	34.93	18.48	24.17
4	CRF Classifier	DEV-1	33.45	74.02	46.08
		DEV-2	51.62	92.02	66.14

2.1.2 Features for Mention Detection

We train CRF with the following set of features. Most of these features are identified and implemented without using any language dependent resources and/or tools. Thus, these can also be used for developing mention detectors for other languages.

1. Context word: Surrounding tokens are used as the features for mention detection. This is useful for automatic identification of mentions from the text.

2. Word suffix and prefix: Fixed length (say, n) word suffixes and prefixes are very effective to identify mentions and work well for the highly inflective Indian languages. Actually, these are the fixed length character strings stripped either from the rightmost or from the leftmost positions of the words. If the length of the corresponding word is less than or equal to $n-1$ then the feature values are not defined and denoted by ND. The feature value is also not defined (ND) if the token itself is a punctuation symbol or contains any special symbol or digit. This feature is included with the observation that mentions share some common suffixes and/or prefixes.

3. Part-of-Speech (PoS) information: PoS information of the current and/or the surrounding tokens are effective for mention identification. Words having PoS classes like common noun (NN), proper noun (NNP), pronoun (PRP) etc. are important for mention detection.

4. Chunk information: Each mention belongs to noun phrase. Thus it is very useful feature to identify mention's boundary. This information was provided with the training, development and test datasets.

5. Suffix List: Variable length suffixes of a word are matched with the predefined list of useful suffixes which are helpful to detect person (e.g., -bAbu [4], -der, -dI, -rA etc.) and pronoun (e.g.,-tI, -ke, -der etc.) names. A binary valued feature is defined that fires if the current word contains any of these suffixes.

6. Noun phrase before Pronoun: For each pronoun, the binary valued feature is set to 1 if it follows a noun phrase (NP).

7. Named Entity Information: The Named Entity (NE) information of the current and/or the surrounding token(s) are used for identifying the mentions. It is very helpful because most of the mentions belong to different NE categories.

8. Pronoun List: We manually prepare a list of pronoun names (e.g., jeMon, kAro, tAhole, onnyoKe etc.) that do not participate in anaphora resolution. This discards pronouns that are not co-referent mentions.

Results of the different mention detection models are reported in Table 1. From the results it is evident that CRF based classifier achieves the best performance for the development data. Based on these observations, finally we select the

[4]Glosses are written in ITRANS notation available, at http://www.aczoom.com/itrans/

CRF based mention detection model for further experiments. In order to identify the mentions from the test data, we create a training set by merging the development data with the training data. The CRF classifier is trained on this resultant training data using the features described above. Evaluation results on the test data are reported in Table 2.

Table 2. Results for mention detection on test data

Document id	precision	recall	F-measure
DOC-TEST-1	78.49	57.48	66.36
DOC-TEST-2	69.75	69.75	69.75
DOC-TEST-3	88.53	49.47	63.47

3 Methods for Anaphora Resolution

In this section at first we describe BART system in brief which is used as the underlying framework for anaphora resolution. Later on we describe our proposed approach for anaphora resolution in Bengali.

3.1 Brief Description of BART System Architecture

Our starting point of coreference resolution system is the toolkit from [21], originally conceived as a modularized version of previous efforts from [12, 11, 20]. BART's final aim is to bring together state-of-the-art approaches, including syntax-based and semantic features. The design of BART is very modular, and this design provides effective separation across several tasks, including engineering new features that exploit different sources of knowledge, and improving the way that co-reference resolution is mapped to a machine learning problem.

BART has five main components, *viz. preprocessing pipeline*, *mention factory*, *feature extraction module*, *decoder* and *encoder*. In addition, an independent language plugin module handles all the language specific information and is accessible from any component. Each module can be accessed independently and thus adjusted to leverage the system's performance on a particular language or domain. The preprocessing pipeline converts an input document into a set of linguistic layers, represented as separate XML files. The mention factory uses these layers to extract mentions and assign their basic properties (number, gender etc.). The feature extraction module describes pairs of mentions $M_i, M_j, i < j$ as a set of features. The decoder generates training examples through a process of sample selection and trains a binary classifier. Finally, the encoder generates testing examples through a (possibly distinct) process of sample selection, runs the classifier and partitions the mentions into coreference chains.

3.2 Approach for Bengali Anaphora Resolution

In this work we extend BART to perform anaphora resolution for Bengali, a resource poor language that has completely different characteristics and orthography compared to English. We perform systematic study to identify most suitable configuration for anaphora resolution. We observe that only a small subset of features are actually useful for Bengali. We design and evaluate our system using the datasets obtained from the shared task of ICON-2011[16]. The available datasets contain all the three types of datasets-training, development and test.

Preprocessing and Markable Extraction. At first the mentions are extracted using the CRF based mention detection model. These markables are then converted to the data format used by BART, namely MMAX2s standoff XML format [8].

Features for Anaphora Resolution. We view anaphora resolution as a binary classification problem. Following similar proposals for English [10], we use the learning framework proposed by Soon et al. [18] as a baseline. Each classification instance consists of two markables, i.e. an anaphora and potential antecedent. Instances are modeled as feature vectors and are used to train a binary classifier. The classifier has to decide, given the features, whether the anaphora and the candidate are coreferent or not. Given BARTs flexible architecture, we explore the contribution of different features implemented in BART for anaphora resolution in Bengali. Given a potential antecedent RE_i and a potential anaphora RE_j , we compute the following features:

1. String Match: This feature holds non-negative integer values. If one candidate is a sub-string of another, its value is 0, else the value is 0 plus the edit distance.

2. Distance: A non-negative integer feature capturing the distance between anaphora and antecedent; if they are in the same sentence, then value of 0 is produced else if their sentence distance is 1 the value of 1 is produced.

3. CorefChain: It is the size of the coreference chain computed for the antecedent so far. This dynamically computed feature gives a boost to central entities making them likely candidates for coreference.

Learning algorithm. In order to learn coreference decisions, we experiment with WEKA's [25] implementation of the C4.5 decision tree learning algorithm [13], with the above mentioned feature combinations. Instances are created following Soon et al. (2001) [18]. We generate a positive training instance from each pair of adjacent coreferent markables. Negative instances are created by pairing the anaphora with any markable occurring between the anaphora and the antecedent. During testing, we perform a closest first clustering of instances deemed coreferent by the classifier. Each text is processed from left to right: each markable is paired with any preceding markable from right to left, until a pair labeled as coreferent is output, or the beginning of the document is reached.

Decoding. In the decoding step, the coreference chains are created by the best-first clustering. Each mention is compared with all of its previous mentions with a probability greater than a fixed threshold value, and is clustered with the highest probability. If none has probability greater than the threshold, the mention becomes a new cluster.

4 Evaluation Results

In this section we report the details of the datasets, evaluation metrics, experiments, results along with the necessary discussions.

4.1 Dataset

For evaluation we used the data sets provided in the ICON NLP Tools Contest on Anaphora Resolution in Indian Languages [16]. For training and development datasets, annotations were provided by the organizers. But no annotation was provided for the test data. In line with the annotations of training and development datasets, we manually annotated test dataset. The statistics of the data sets in terms of number of sentences and number of tokens present in each set are provided in Table 3.

Table 3. Statistics of the datasets

Dataset	#sentences	#tokens
Training	881	10,504
Development	598	5,785
Test	572	6,985

4.2 Experiments and Discussions

In order to evaluate the anaphora resolution system we used the MUC score [22]. We experiment with the different mention detectors for anaphora resolution. Table 4 shows recall, precision and F-measure values of the system trained using the training data and evaluated using the development data when mentions are taken as noun phrases.

Table 5 reports the results of the system for the development data when NE or pronoun is considered as mentions. Comparisons between Table 4 and Table 5 show that the later model performs better. We then present the results in Table 6 for the system when *person names* or *pronouns* are considered as mentions.

Finally, we extract the markables using a CRF based classifier and its results are presented in Table 7. Comparisons show that the system performs the best with this configuration.

Results of these tables reveal the fact that if we determine the mentions from the CRF based classifier then we achieve the best results for anaphora resolution. Based on these results on development data, we evaluate the system on the test using the mentions extracted by the CRF model. Results on the test data are reported in Table 8, and it shows the recall, precision and F-measure values of 56.00%, 46.50% and 50.80%, respectively. We don't have any scope for

Table 4. Results on development data using only noun phrases as mentions

Datasets ID	recall	precision	F-measure
DEV-1	9.00	6.20	7.30
DEV-2	25.40	23.70	24.60
MUC-TOTAL	21.20	18.10	19.50

Table 5. Results on development data using NE or pronoun as mentions

Datasets ID	recall	precision	F-measure
DEV-1	47.80	11.80	18.90
DEV-2	71.60	32.30	44.50
MUC-TOTAL	65.40	24.30	35.40

Table 6. Results of the system on development data using person name or pronoun as mentions

Datasets ID	recall	precision	F-measure
DEV-1	35.90	13.20	19.30
DEV-2	72.10	34.20	46.40
MUC-TOTAL	62.70	27.70	38.40

Table 7. Results on development data using mentions generated by a CRF-based classifier

Datasets ID	recall	precision	F-measure
DEV-1	65.70	16.80	26.80
DEV-2	89.20	35.50	50.80
MUC-TOTAL	83.10	29.00	42.90

comparing these figures due to the non-availability of any published work that was carried out on these shared task datasets.

We compare our approach with a baseline model, developed following the Soon et al. [17] implementation. Some of the features include [5] *mention type*, *gender agreement*, *number agreement*, *alias*, *appositive*, *string matching*, *semantic class compatibility*, *sentence distance*. Results obtained are shown in Table 9. Results of the baseline exhibits much inferior performance.

Discussions. We explore different models for mention detections. We observed that the mention detection performs best with the supervised machine learner, CRF. This module is thereafter integrated with BART to develop a anaphora resolution system for Bengali. Mentions generated using the outputs of a CRF based classifier are utilized to generate markables. These system mentions are then used for the encoding and decoding modules in BART. Experimental results shown in Tables 2-6 show that mention detection plays an important role in anaphora resolution. Our results show that only a small subset of features is actually helpful for anaphora resolution in Bengali. The proposed approach performs reasonably better compared to the baseline. Detailed analysis suggest that a morphological analyzer and shallow parser might be more useful to improve the performance further.

[5] http://www.sfs.uni-tuebingen.de/ versley/BART/BART-intro.pdf

Table 8. Results on test data by the proposed approach

Datasets ID	recall	precision	F-measure
DOC-TEST-1	54.50	40.70	46.60
DOC-TEST-2	68.30	48.90	57.00
DOC-TEST-3	47.50	49.40	48.40
MUC-TOTAL	56.00	46.50	50.80

Table 9. Results of baseline model on test data

Datasets ID	recall	precision	F-measure
DOC-TEST-1	20.6	22.6	21.5
DOC-TEST-2	22.0	22.0	22.0
DOC-TEST-3	18.4	27.9	22.2
MUC-TOTAL	20.2	24.1	21.9

5 Conclusion

In this paper we have presented our first attempt for anaphora resolution in resource-scarce language like Bengali. We have adapted BART, a state-of-the-art coreference resolution model originally developed for English for performing the same task for Bengali. We explore many models for markable identification, and observed that a supervised CRF based classifier produces the best results. We have identified a set of language independent features for mention detection, and so these can also be used for other languages. The contribution of this work is two-fold, *viz.* (i). attempt to build a machine learning based anaphora resolution system for a resource-poor Indian language; and (ii). domain adaptation of a state-of-the-art English co-reference resolution system for Bengali which has completely different orthography and characteristics. Due to the lack of published works, we could not compare our results with the existing systems. Currently our focus is on developing methods for capturing the missing markables; and identifying features related to morphology and shallow parsing information. These might require either the development of such engines or tuning of such systems, if available. Future work will concentrate on porting the systems to other Indian languages, e.g. Hindi and Telugu, as well as investigating the portability and usefulness of syntactic, morphological and semantic information across different languages.

Acknowledgment

The authors acknowledge a partial support of the "European Community's Seventh Framework Programme (FP7/2007-2013) under the grant #288024: LiMoSINe – Linguistically Motivated Semantic aggregation engiNes".

References

1. **Chatterji, S., Dhar, A., Barik, B., PK, M., Sarkar, S., & Basu, A.** (**2011**). Anaphora resolution for bengali, hindi, and tamil using random tree algorithm in weka. In *In Proceedings of the ICON-2011*.
2. **Dakwale, P. & Sharma, H.** (**2011**). Anaphora resolution in indian languages using hybrid approaches. In *In Proceedings of the ICON-2011*.
3. **Ghosh, A., Neogi, S., Chakrabarty, S., & Bandyopadhyay., S.** (**2011**). Anaphora resolution in bengali: ju_cse_nlp system at icon 2011. In *In Proceedings of the ICON-2011*.
4. **Lafferty, J. D., McCallum, A., & Pereira, F. C. N.** (**2001**). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *In Proceedings of the ICML*. 282–289.
5. **Luo, X., Ittycheriah, A., Jing, H., Kambhatla, N., & Roukos, S.** (**2004**). A mention synchronous coreference resolution algorithm based on the bell tree. In *Proceedings of ACL-04*. 136–143.
6. **McCarthy, J. F. & Lehnert, W. G.** (**2009**). Using decision trees for coreference resolution. In *Proceedings of IJCAI 1995*. 1050–1055.
7. **Morton, T. S.** (**1999**). Using coreference in question answering. In *Proceedings of TREC-8*. 85–89.
8. **Muller, C. & Strube, M.** (**2000**). Multi-level annotation of linguistic data with mmax2. In *Sabine Braun, Kurt Kohn, and Joybrato Mukherjee, editors, Corpus Technology and Language Pedagogy: New Resources, New Tools, New Methods*. Germany.
9. **Ng, V. & Cardie, C.** (**2002**). Improving machine learning approaches to coreference resolution. In *Proceedings of ACL-02*. 104–111.
10. **Ng, V. & Cardie, C.** (**2002**). Improving machine learning approaches to coreference resolution. In *In Proceedings of ACL-02*. 104–111.
11. **Poesio, M. & Kabadjov, M. A.** (**2004**). A general purpose, off-the-shelf anaphora resolution module: Implementation and preliminary evaluation. In *In Proceedings of LREC-04*.
12. **Ponzetto, S. P. & Strube, M.** (**2008**). Exploiting semantic role labeling, wordnet and wikipedia for coreference resolution. In *In Proceedings of HLT-NAACL-06*.
13. **Quinlan, J. R.** C4.5: Programs for machine learning. *Morgan Kaufmann, San Francisco, Cal*.
14. **Senapati, A. & Garain, U.** (**2011**). Anaphora resolution system for bengali by pronoun emitting approach. In *In Proceedings of the ICON-2011*.
15. **Sha, F. & Pereira, F.** (**2003**). Shallow Parsing with Conditional Random Fields. In *In Proceedings of the NAACL-03*. 134–141.
16. **Sobha et. al.** (**2011**). Nlp tools contest on anaphora resolution in indian languages. In *In Proceedings of the ICON-2011*.
17. **Soon, W. M., Ng, H. T., & Lim, D. C. Y.** (**2001**). A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4), 521–544.
18. **Soon, W. M., Ng, H. T., & Lim, D. C. Y.** (**2001**). A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4), 521–544.
19. **Steinberger, J., Poesio, M., Kabadjov, M. A., & Jeek, K.** (**2007**). Two uses of anaphora resolution in summarization. *Information Processing and Management: an International Journal*, 43(6), 166–1680.
20. **Versley, Y.** (**2006**). A constraint-based approach to noun phrase coreference resolution in german newspaper text. In *In Proceedings of Konferenz zur Verarbeitung Nat rlicher Sprache*. 143–150.
21. **Versley, Y., Ponzetto, S. P., Poesio, M., Eidelman, V., Jern, A., Smith, J., Yang, X., & Moschitti, A.** (**2008**). Bart: A modular toolkit for coreference resolution. In *In Proceedings of LREC-2008*.
22. **Vilain, M., Burger, J., Aberdeen, J., Connolly, D., & Hirschman, L.** (**1995**). A model-theoretic coreference scoring scheme. In *In Proceedings of the 6th Message Understanding Conference (MUC-6)*. 45–52.

23. **Walker, C., Strassel, S., Medero, J., & Maeda, K.** (**2006**). Ace 2005 multilingual training corpus. In *LDC2006T06*. Penn.: Linguistic Data Consortium.

24. **Weischedel, R., Pradhan, S., Ramshaw, L., Palmer, M., Xue, N., Marcus, M., Taylor, A., Greenberg, C., Hovy, E., Belvin, R., & Houston, A.** (**2008**). Ontonotes release 2.0. In *LDC2008T0*. Penn.: Linguistic Data Consortium.

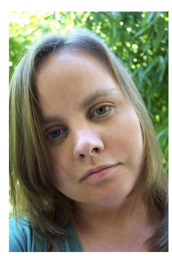
25. **Witten, I. H. & Frank, E.** Data mining: Practical machine learning tools and techniques. *Morgan Kaufmann, San Francisco, Cal*.

26. **Yang, X., Zhou, G., Su, J., & Tan, C. L.** (**2003**). Coreference resolution using competition learning approach. In *Proceedings of ACL-03*. 176–183.

Utpal Kumar Sikdar is a Senior Research Fellow in the Department of Computer Science and Engineering, IIT Patna, India. He received his Bachelors' and Masters' degrees in Information Technology and Software Engineering from Vidyasagar University and Jadavapur University in 2004 and 2008, respectively. Prior to joining to his Ph.D. programme he served as a specialist at Tata Elxsi Ltd., India. His research interests include anaphora resolution and information extraction.

Asif Ekbal is an Assistant Professor in the Department of Computer Science and Engineering, IIT Patna. He received his Masters' and PhD in Computer Science and Engineering from Jadavpur University in 2004 and 2009, respectively. His master's and PhD theses were related to the broad areas of Natural Language Processing. Before joining IITP, he worked as a postdoctoral research fellow at the University of Trento, Italy and Heidelberg University, Germany. His broad areas of research include Natural Language Processing (NLP), Information Extraction, Bio-text Mining etc. He has authored/co-authored more than 80 technical articles in international journals, book chapters, and conference/workshop proceedings. He received the Best Innovative Project Award from the Indian National Academy of Engineering in the year 2000.

Sriparna Saha is an Assistant Professor in the Department of Computer Science and Engineering, IIT Patna. She received his Masters' and PhD in Computer Science from Indian Statistical Institute, Kolkata in 2005 and 2011, respectively. Her master's and PhD theses were related to the areas of Pattern Recognition, Multiobjective Optimization and Clustering. Before joining IITP, she was working as a postdoctoral research fellow in the Department of Information Engineering and Computer Science, University of Trento, Italy. Before that she worked as a postdoctoral research fellow in Heidelberg University, Germany for a period of one year. Her broad areas of research include Pattern Recognition, Optimization Techniques, Clustering, NLP, Bio-informatics etc. She has authored/co-authored around 100 technical articles in international journals, book chapters, and conference/workshop proceedings. She is the recipient of Google India Women in Engineering Award (2008). She has also received ISIAA Rashi Ray Memorial Medal from Indian Statistical Institute, Kolkata, India, for being adjudged the First Class First in M. Tech (2005).



Olga Uryupina is a postdoctoral fellow at the Department of Information Engineering and Computer Science, University of Trento, Italy (disi.unitn.it/). She received her PhD in Computational Linguistics from the University of Saarland (Germany) in 2007. She was a visiting researcher at the University of Edinburgh (2009) and a postdoctoral fellow at the Center for Mind and Brain Sciences, University of Trento. Her research interests include natural language processing, in particular, discourse analysis and coreference resolution. She is investigating possibilities of building complex deep semantic models of discourse, combining various sources of information, from surface-based features to automatically extracted relations and background knowledge. She worked for the LiveMemories project (www.livememories.org) and is currently involved in the LiMoSINe project (http://www.limosine-project.eu).

Massimo Poesio obtained his PhD in Computer Science from the University of Rochester (USA) in 1994, and was a EPSRC Advanced Research Fellow at the University of Edinburgh (UK) from 1994-2001. He is now a Professor at the University of Essex, where he has been since 2001. He is a cognitive scientist with a particular focus on human language technology; his research interests include computational models of semantic interpretation, particularly anaphora resolution/coreference; relation extraction, particularly applied to entity disambiguation; deception detection; the creation of large corpora of semantically annotated data, also using games-with-a-purpose (www.phrasedetectives.org); and the study of conceptual knowledge using a combination of methods from human language technology and from neuroscience.

Article received on 21/12/2012; accepted on 16/01/2013.