Adapting a State-of-the-art Anaphora Resolution System for Resource-poor Language

Utpal Kumar Sikdar [1], **Asif Ekbal** [1], **Sriparna Saha** [1]
Olga Uryupina [2], **Massimo Poesio** [3]
[1] Department of Computer Science and Engineering, IIT Patna, India,
{utpal.sikdar,asif,sriparna}@iitp.ac.in
[2] University of Trento, Center for Mind/Brain Sciences, uryupina@unitn.it
[3] University of Essex, Language and Computation Group, poesio@essex.ac.uk

Abstract

In this paper we present our work on adapting a state-of-the-art anaphora resolution system for a resource poor language, namely Bengali. Performance of any anaphoric resolver greatly depends on the quality of a high accurate mention detector. We develop a number of models for mention detection based on heuristics and machine learning. Our experiments show that, a language-dependent system can attain reasonably good performance when re-trained on a new language with a proper subset of features. The system yields the MUC recall, precision and F-measure values of 57.80%, 79.00% and 66.70%, respectively. Our experiments with other available scorers show the F-measure values of 59.47%, 49.83%, 31.81% and 70.82% for BCUB, CEAFM, CEAFE and BLANC, respectively.

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 anaphor. The noun phrases that may participate in co-reference relation are called mentions/markables. Various practical tasks require language technology; 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. Anaphoric information is also needed to solve several other such kinds of Natural Language Processing (NLP) problems. Most of these works on supervised machine learning co-reference resolution have been developed for English (Soon et al., 2001; Ng and Cardie, 2002; Yang et al., 2003; Luo et al., 2004), due to the availability of large corpora such as ACE (Walker et al., 2006) and OntoNotes (Weischedel et al., 2008). BART, the Beautiful Anaphora Resolution Toolkit (Versley et al., 2008), (Ponzetto and Strube, 2006), (Poesio and Kabadjov, 2004), 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, MaxEnt and Support Vector Machine (SVM).

Literature shows the significant amount of works in the area of anaphora resolution. But these (Pradhan et al., 2012; Ng, 2010; Poesio et al., 2010) are mainy in non-Indian languages. The works related to anaphora resolution in Indian languages are still at the nascent stage due to the following facts: Indian languages are resource constrained, i.e. corpus, annotated corpus, morphological analyzers, Part-of-Speech (PoS) taggers, named entity (NE) taggers, parsers etc. are not readily available. There have been few attempts for anaphora resolution in Indian languages. In 2011 a shared task on NLP Tools Contest on Anaphora Resolution in Indian Languages was organized in association with 9th International Conference on Natural Language Processing (ICON 2011) [1]. Four teams participated in this contest with the varying approaches(Chatterji et al., 2011; Dakwale and Sharma, 2011; Senapati and Garain, 2011; Ghosh et al., 2011).

In this paper we propose our work on anaphora

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

resolution in Bengali, a resource poor language. We develop a number of models for mention detection. The mention detector developed with the supervised classifier, Conditional Random Field (Lafferty, 2001) performs best for the anaphora resolution. We identify and implement several features for mention detection as well for anaphora resolution. Detailed experiments were carried out on the development set to identify the most relevant set of features. Later on we use that particular configuration to report the final evaluation results on test data.

2 Brief Description of BART System Architecture

Our starting point of anaphora resolution system is the toolkit from (Versley et al., 2008), originally conceived as a modularized version of previous efforts from (Ponzetto and Strube, 2006; Poesio and Kabadjov, 2004; Versley, 2006; Broscheit et al., 2010). BART's final aim is to bring together state-of-the-art approaches, including syntax-based and semantic features. The state-of-the-art anaphora resolution system, BART has five main components: 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.

2.1 Models for Mention Detection

Robust mention detection is an essential component of anaphora resolution system in any language. 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 very first step of anaphora resolution process tries to identify the occurrence of mentions in the documents. In our original experimental datasets, three information were provided for each token: Part-of-Speech (PoS), phrase (or, chunk) and Named Entity (NE). We develop the following mention detection models:

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

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

3. **Third Model**: In the third model we take only person name or pronoun (PER/PRP) as a candidate of mention. Results in Table 1 show a little improvement in the performance for one document, however the performance for the other documents decrease.

4. **Fourth Model**: Here we use Conditional Random Field (CRF) based supervised classifier 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 create a training set for mention detection based on the mentions present in the original training data. Evaluation results in Table 1 clearly show that this mention detection system is the best compared to the other three models. Details of this systems are mentioned in the following subsection.

2.2 Conditional Random Field(CRF) based Mention Detection System

To formulate the problem of mention detection using CRF(Lafferty, 2001), we consider the token

[2] Here B, I and O denote the beginning, internal and outside the entity mention

of a sentence as an element of the observation sequence and the corresponding class label as an element of its state sequence. We have used the C^{++} based CRF^{++} package [3].

2.2.1 Features for Mention Detection

We train CRF with the following set of features.

1. **Context word**: The contextual information of a target entity plays a significant role to decide whether it is a potential candidate for being a mention (or markable). We use the preceding and following few tokens as the features.

2. **Word suffix and prefix**: Fixed length (say, n) word suffixes and prefixes are used as the features for mention detection. These are the fixed length character strings stripped either from the rightmost (for suffix) or from the leftmost positions (for prefix) of the words. We included this feature with the observation that mentions, in general, share some common character sequences either at the beginning or at the end.

3. **Part-of-Speech (PoS) information**: PoS information of the token is effective for mention detection. We consider the PoS classes like NN (Common noun), NNP (Proper noun), PRP (Pronoun) etc. as important for mention detection.

4. **Chunk information**: Each mention belongs to the noun phrase and so its boundary identification is important. We use the chunk information provided with the 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*, *-der*, *-dI*, *-rA* etc.) and pronoun (e.g.,*-tI*, *-ke*, *-der* etc.) names [4]. We prepared such lists from the training data. A binary valued feature is defined that fires if the current word contains any of these suffixes.

6. **Noun phrase preceding pronoun**: We observed that in many cases the pronoun appears immediately after the potential markable candidate. We define a binary-valued feature that is set to 1 for a pronoun (PRP) if it follows a noun phrase (NP).

Sr.	Mentions	DevData	precision	recall	F-measure
1	NP	Doc-1 Doc-2	26.08 25.76	99.16 99.62	41.30 40.93
2	NE /PRP	Doc-1 Doc-2	72.02 47.18	33.80 25.86	46.01 38.35
3	PER /PRP	Doc-1 Doc-2	82.47 92.47	13.13 25.86	22.65 40.42
4	CRF Classifier	Doc-1 Doc-2	88.17 91.77	41.62 70.50	56.55 79.74

Table 1: Results of different approaches for mention detection on development data

7. **Named entity information**: The Named Entity (NE) class is used for identifying mentions. This is a very useful feature as the majority of the mentions belong to the 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.

9. **First word**: Noun phrases often appear at the beginning for the particular datasets that we have used, and these can most likely be the mentions. This feature is used to define whether the token is the first word in the sentence or not.

10. **Morphological features**: We extract morphological features from the shallow parser available at [5]. The features include *lemma* and number *information* (singular/plural) of the words.

11. **Fine−grained noun information**: The fine−grained information of nouns are extracted from the PoS tags. The feature checks whether the token is definite noun or demonstrative noun, and decides accordingly.

We present the results of mention detection module in Table 1. It shows that CRF based classifier attains the best performance. Inspired by these results, we identify mentions in test data using this CRF based classifier. We merge the development

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

[4] Henceforth all the Bengali glosses are written in ITRANS notations available at http://www.aczoom.com/itrans/

[5] http://ltrc.iiit.ac.in/showfile.php?filename=downloads/shallow_parser.php

TestData	precision	recall	F-measure
Doc-1	81.32	73.70	77.32
Doc-2	81.61	73.76	77.49
Doc-3	93.67	51.99	66.87

Table 2: Results for mention detection on test data

data with the training data and create a new training dataset for CRF. Results on the test data are reported in Table 2.

3 Methods for Anaphora Resolution

In this work we extend BART to perform anaphora resolution for Bengali, a resource poor language. We perform systematic study to identify most suitable configuration of BART for anaphora resolution in Bengali. We identify and implement several features for this task. We design and evaluate our system using the Bengali datasets obtained from the NLP Tools Contests on Anaphora Resolution in Indian Languages, organized in ICON-2011 [6]. The Bengali corpus contains three types of datasets- training, development and test.

3.1 Preprocessing and Markable Extraction

For the anaphora resolution system, mentions are identified from the datasets based on the gold annotations. These are treated as the markables. Thereafter we convert the markables to the data format used by BART, namely MMAX2s standoff XML format.

3.2 Features for anaphora resolution

We view coreference resolution as a binary classification problem. We use the learning framework proposed by (Soon et al., 2001) as a baseline. Each classification instance consists of two markables, i.e. an anaphor and its potential antecedent. Instances are modelled as feature vectors and are used to train a binary classifier. The classifier has to decide, given the features, whether the anaphor and the candidate antecedent are coreferent or not. To improve the performance we define some features specific to the language. Given BART's flexible architecture, we explore the contribution of some features implemented in BART for co-reference resolution in Bengali. Given a potential antecedent RE_i and a anaphor RE_j, we compute the following features:

1. **String match**: This feature compares between the two mentions. The value of this feature is true if the candidate anaphor (RE_j) and antecedent (RE_i) have the same surface strings forms, otherwise false.

2. **Sentence distance**: A non-negative integer feature capturing the distance between anaphor 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. **Markable distance**: This is also a non-negative integer feature that captures the distance in terms of the number of mentions between the two markables.

4. **First person pronoun**: This feature is defined based on the direct and indirect speech. For a given anaphor-antecedent pair (RE_j, RE_i) a feature is set to high if RE_j is a first person pronoun found within a quotation and RE_i is a mention immediately preceding it within the same quote. If RE_i is outside the quote and appears either in the same sentence or in any of the previous three sentences and is not first person then the corresponding feature is also set to high. The feature also behaves in a similar way if the pair (RE_j, RE_i) appears outside the quotation.

5. **Second person pronoun**: This feature checks whether the pair (RE_j, RE_i) is in the same quote and fires the feature accordingly. It is true if RE_j is second person and RE_i is other than the first person. If RE_j is inside the quotation, and RE_i ends with the suffix ”*ke* and is outside the quote then the feature fires.

6. **Third person pronoun**: This feature checks whether the pair (RE_j, RE_i) appears inside or outside the quotation. It feature fires if both the mentions either appear within or outside the quotation.

7. **Reflexive pronoun**: For a given pair (RE_j, RE_i), this feature checks whether RE_j is a reflexive pronoun and fires accordingly. This means if any antecedent is immediately followed by a reflexive pronoun then the feature is true, otherwise false.

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

8. **Number agreement**: If both anaphor(RE_j) and antecedent(RE_i) agree in their number information then the feature value is set to true, otherwise false. We extract this feature from the shallow parser available at [7]. The parser was not able to take longer sentences as inputs and so we had to pre-process the data before running the parser.

9. **Semantic class feature**: If both RE_j and RE_i agree in their semantic classes then this feature is set to true, otherwise false. In particular this feature checks whether the pair either belongs to *person class* or *organization class* or *location class*.

10. **Alias feature**: It checks whether RE_j is an alias of RE_i or not.

11. **Appositive feature**: If RE_j is in apposition to RE_i then the value of this feature is set to true, otherwise it is false.

12. **String kernel**: String kernel similarity is used to estimate the similarity between two strings based on the string subsequence kernel.

13. **Mention type**: Following (Soon et al., 2001), we have encoded mention types (name, nominal or pronoun) of the anaphor and the antecedent. In addition, we check whether the anaphor RE_j is a definite pronoun or demonstrative pronoun or merely a pronoun. We also check whether each of the entities in the mention pair denotes proper name.

3.3 Learning algorithm

In order to learn coreference decisions, we experiment with WEKA's (Witten and Frank, 2005) implementation of the C4.5 decision tree learning algorithm (Quinlan, 1993), with the above mentioned feature combinations. Instances are created following (Soon et al., 2001). We generate a positive training instance from each pair of adjacent coreferent markables. Negative instances are created by pairing the anaphor with any markable occurring between the anaphor 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 labelled as coreferent is output, or the beginning of the document is reached.

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

Table 3: Statistics of the datasets

3.4 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

4.1 Dataset

For our experiments we use the data sets provided in the ICON NLP Tools Contest on Anaphora Resolution in Indian Languages. The datasets were taken from the Bengali literature (mostly from the short stories). All the datasets were provided with PoS, chunk and NE information. For training and development datasets, anaphoric annotations were provided by the organizers. However for test set there was no annotation available. In line with the annotations of training and development datasets, we manually annotated test dataset. Some statistics of the datasets are presented in Table 3.

4.2 Evaluation metrics and results

In order to evaluate the anaphora resolution system we use different scorers such as MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), CEAF (Luo, 2005) and BLANC (Recasens and Hovy, October 2011). We experiment with the different mention detectors for anaphora resolution. Table 4 shows the MUC recall, precision and F-measure values of the system trained using the training data and evaluated using the development data. Experiments were carried out on a high performance computing facility with the following configuration: Dell machine, 216 cores, 2.66 GHZ Intel Xeon processors, 4 GB RAM/core, and 10 TB storage.

[7] http://ltrc.iiit.ac.in/showfile.php?filename=downloads/shallow_parser.php

Mentions	recall	precision	F-measure
NP	52.50	40.40	45.60
NE/PRP	45.20	69.40	54.80
PER/PRP	45.20	66.30	53.80
CRF Classifier	52.20	78.80	62.80

Table 4: Results with MUC scorer on development data

Scorers	recall	precision	F-measure
MUC	57.80	79.00	66.70
BCUB	51.02	71.27	59.47
CEAFM	49.83	49.83	49.83
CEAFE	48.88	23.58	31.81
BLANC	70.66	70.99	70.82

Table 5: Overall results on test data

Results of Table 4 reveal the fact that the proposed anaphora resolution system achieves the best performance when CRF based classifier is used for mention detection. Based on these results on development data, we evaluate the system for the test data using the mentions extracted by the CRF based machine learner. Results on the test data are reported in Table 5. Results show the F-measure values of 66.70%, 59.47%, 49.83%, 31.81% and 70.82% for MUC, BCUB, CEAFM, CEAFE and BLANC, respectively.

4.3 Discussion

We explore different models for mention detections. We observed that the mention detection performs best with the supervised machine learner, CRF. These system mentions are then used for the encoding and decoding modules in BART. Experimental results shown in Table 4 show that mention detection plays an important role in anaphora resolution. We implement the baseline model using a subset of the features reported in (Soon et al., 2001). These include number agreement, alias, string matching, semantic class agreement, sentence distance, appositive and several features (c.f. Section 3.2). This model showed the MUC recall, precision and F-measure values of 38.8%, 67.4% and 49.3%, respectively. This is clearly much less compared to our proposed model. Comparisons with the available related works (specific to the language) show that our proposed system achieves state-of-the-art accuracy.

5 Conclusion

We present a anaphora resolution system for Bengali, a resource-poor language based on BART, a state-of-the-art coreference resolution model originally developed for English. We explore many models for markable identification, and observed that a supervised CRF based classifier produces the best results. The main focus of this work is to build a machine learning based anaphora resolution system for a resource-poor Indian language. Our system attains the state-of-the-art accuracy level. Currently our focus is on developing methods for capturing the missing markables; and identifying more syntactic and semantic features. Future work will also concentrate on porting the systems to other Indian languages, e.g. Hindi and Telugu, as well as investigating the portability and usefulness of more syntactic, morphological and semantic information across different languages. We also aim to perform systematic feature selection for mention detection and anaphora resolution.

6 Acknowledgments

The research described in this paper has been partially supported by the European Community's Seventh Framework Programme (FP7/2007-2013) under the grant #288024: LiMoSINe – Linguistically Motivated Semantic aggregation engiNes.

References

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *Proc. of the LREC workshop on Linguistic Coreference*, pages 563–566, Granada.

Samuel Broscheit, Massimo Poesio, Simone Paolo Ponzetto, Kepa Joseba Rodriguez, Lorenza Romano, Olga Uryupina, Yannick Versley, and Roberto Zanoli. 2010. BART: A multilingual anaphora resolution system. In *Proceedings of the 5th International Workshop on Semantic Evaluations (SemEval-2010),* Uppsala, Sweden, 15–16 July 2010, pages 104–107.

Sanjay Chatterji, Arnab Dhar, Biswanath Barik, Moumita PK, Sudeshna Sarkar, and Anupam Basu. 2011. Anaphora resolution for Bengali, Hindi, and Tamil using random tree algorithm in weka. In *9th International Conference on Natural Language Processing, Anna University-MIT Campus, Chromepet, Chennai, India 16-19 December, 2011, http://ltrc.iiit.ac.in/icon2011/contests.html.*

Praveen Dakwale and Himanshu Sharma. 2011. Anaphora resolution in Indian languages using hybrid approaches. In *9th International Conference on Natural Language Processing, Anna University-MIT Campus, Chromepet, Chennai, India 16-19 December, 2011, http://ltrc.iiit.ac.in/icon2011/contests.html.*

Aniruddha Ghosh, Snehasis Neogi, Saikat Chakrabarty, and Sivaji Bandyopadhyay. 2011. Anaphora resolution in Bengali. In *9th International Conference on Natural Language Processing, Anna University-MIT Campus, Chromepet, Chennai, India 16-19 December, 2011, http://ltrc.iiit.ac.in/icon2011/contests.html.*

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

Xiaoqiang Luo, Abe Ittycheriah, Hongyan Jing, A Kambhatla, and Salim Roukos. 2004. A mention-synchronous coreference resolution algorithm based on the bell tree. In *Proc. of the ACL*, pages 135–142.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proc. NAACL / EMNLP*, Vancouver.

Vincent Ng and Claire Cardie. 2002. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 104–111.

Vincent Ng. 2010. Supervised noun phrase coreference research: The first fifteen years. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1396–1411.

Massimo Poesio and Mijail A. Kabadjov. 2004. A general-purpose, off-the-shelf anaphora resolution module: Implementation and preliminary evaluation. In *Proceeding of LREC*, pages 663–666.

Massimo Poesio, Simone Paolo Ponzetto, and Yannick Versley. 2010. Computational models of anaphora resolution: A survey.

Simone Paolo Ponzetto and Michael Strube. 2006. Exploiting semantic role labeling, wordnet and wikipedia for coreference resolution. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 192–199, New York City, USA, June. Association for Computational Linguistics.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea, July. Association for Computational Linguistics.

J. Ross Quinlan. 1993. *programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Marta Recasens and Eduard Hovy. October, 2011. BLANC: Implementing the rand index for coreference evaluation. *Natural Language Engineering*, 17(04):485–510.

Apurbalal Senapati and Utpal Garain. 2011. Anaphora resolution system for Bengali by pronoun emitting approach. In *9th International Conference on Natural Language Processing, Anna University-MIT Campus, Chromepet, Chennai, India 16-19 December, 2011, http://ltrc.iiit.ac.in/icon2011/contests.html.*

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Comput. Linguist.*, 27(4):521–544, December.

Yannick Versley, Simone Paolo Ponzetto, Massimo Poesio, Vladimir Eidelman, Alan Jern, Jason Smith, Xiaofeng Yang, and Alessandro Moschitti. 2008. Bart: A modular toolkit for coreference resolution. In *HLT-Demonstrations '08 Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies*, pages 9–12.

Yannick Versley. 2006. A constraint-based approach to noun phrase coreference resolution in german newspaper text. In *Proceedings of Konferenz zur Verarbeitung Nat rlicher Sprache*, pages 143–150.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proc. of the Sixth Message Understanding Conference*, pages 45–52.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus, linguistic data consortium, philadelphia.

Ralph Weischedel, Sameer Pradhan, Lance Ramshaw, Martha Palmer, Nianwen Xue, Mitchell Marcus, Ann Taylor, Craig Greenberg, Eduard Hovy, Robert Belvin, and Ann Houston. 2008. Ontonotes release 2.0:ldc2008t04 philadelphia penn.: Linguistic data consortium.

Ian H. Witten and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Xiaofeng Yang, Guodong Zhou, Jian Su, and Chew Lim Tan. 2003. Coreference resolution using competition learning approach. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 176–183.