Feature Selection in Anaphora Resolution for Bengali: A Multiobjective Approach

Utpal Kumar Sikdar, Asif Ekbal, and Sriparna Saha

Department of Computer Science and Engineering,
Indian Institute of Technology, Patna, India
{utpal.sikdar,asif,sriparna}@iitp.ac.in

Abstract. In this paper we propose a feature selection technique for anaphora resolution for a resource-poor language like Bengali. The technique is grounded on the principle of differential evolution (DE) based multiobjective optimization (MOO). For this we explore adapting BART, a state-of-the-art anaphora resolution system, which is originally designed for English. There does not exist any globally accepted metric for measuring the performance of anaphora resolution, and each of MUC, B^3, CEAF, BLANC exhibits significantly different behaviours. System optimized with respect to one metric often tend to perform poorly with respect to the others, and therefore comparing the performance between the different systems becomes quite difficult. In our work we determine the most relevant set of features that best optimize all the metrics. Evaluation results yield the overall average F-measure values of 66.70%, 59.70%, 51.56%, 33.08%, 72.75% for MUC, B^3, CEAFM, CEAFE and BLANC, respectively.

1 Introduction

The task of anaphora or coreference resolution refers to the task of identifying mentions (basically noun phrases) that denote the same real world objects, or entities. Many crucial applications involving Natural Language Processing (NLP), for example, Information extraction, question-answering, machine translation, text summarization etc. require the task of coreference resolution to be performed. Most of the existing works concern with some of the languages such as English [1,2], due to the availability of different lexical resources and large corpora like as ACE [3] and OntoNotes [4]. In this work we explore how a state-of-the-art English coreference system, BART [5] can be adapted for anaphora resolution in Bengali, a resource-scare language.

India is a multilingual country with great cultural and linguistic diversities. There has not been significant number of works in anaphora resolution involving Indian languages due to the following facts: Indian languages are resource constrained, i.e. annotated corpora, morphological analyzers, part of speech (PoS) taggers, named entity (NE) taggers, parsers etc. are not readily available in the required measure. Literature shows the existence of few works [6,7,8] for anaphora resolution in the languages like Hindi and Tamil. In recent times a generic framework for anaphora resolution in Indian languahas been reported in [9]. However, based on these works it is difficult to get a comprehensive view of the research on anaphora resolution related to Indian languages

A. Gelbukh (Ed.): CICLing 2015, Part I, LNCS 9041, pp. 252–263, 2015.
DOI: 10.1007/978-3-319-18111-0_20

because each of these was developed using the self-generated datasets. Therefore, it is not fair to compare between the algorithms reported in these works.

The first benchmark setup for anaphora resolution involving Indian languages was established in ICON-2011 NLP Tools Contest on Anaphora Resolution[1]. Six teams participated in this shared task with the systems, developed based on either machine learning or rules. Out of these six, four addressed the issues of anaphora resolution in Bengali, and one each for Hindi and Tamil. Apart from these an anaphora resolution system for Bengali is reported in [10], where various models for mention detection were developed, and their impact on anaphora resolution were reported. In another work, authors [11] showed how a off-the-shelf anaphora resolution system can be effectively used for Bengali. A more recent study on anaphora resolution in Bengali can be found in [12]. In contrast to the previous works, here we develop an efficient technique for feature selection in anaphora resolution based on the concept of multiobjective optimization (MOO) that incorporates differential evolution (DE) as an underlying optimization technique. Our approach is able to determine the best optimized feature sets for the five well-known evaluation metrics of coreference resolution. The method also demonstrates how systematic feature selection can help in achieving the reasonable performance with much reduced feature sets.

For anaphora resolution, there have been not much research for explicit automatic optimization except the one proposed in [2], where no significant performance improvement was observed over the baseline that was constructed with all the available features. A systematic effort of manual feature selection on the benchmark datasets was carried out by [13], who evaluated over 600 features. The very first attempt for automatic optimization of anaphora resolution was carried out by [14]. She investigated the usability of evolutionary genetic algorithms for automatic optimization of features and parameters with respect to a machine learning algorithm. She suggested that such a technique may yield significant performance improvements on the MUC-6/7 datasets. (MUC, CEAF or BLANC).

The concept of MOO for feature selection in anaphora resolution has been addressed in [15] for the English language. A genetic algorithm (GA) based MOO technique was developed for automatic feature selection in [15], where it has been shown how the method can simultaneously optimize more than one objective function, and determine near optimal features that achieve superior performance over the baseline model, developed with all the available features. In contrast to this previous study [15], we don't make use of GA as an optimization technique, and perform feature selection for a non-English language. It is to be noted that GA and DE are two different optimization algorithms. A single objective optimization (SOO) based feature selection method for performing feature selection for Bengali is recently been reported in [12]. The work reported in our current research differs from [10,11] in the sense that this work concerns with the development of a method for automatic feature selection based on a DE based MOO technique. MOO and SOO are fundamentally two different concepts. In SOO, we focus on optimizing only one objective function. But in MOO, our aim is to simultaneously optimize more than one objective function. The output of MOO produces a set of solutions on the Pareto optimal front. Each of these solutions is equally important

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

from the algorithmic points of view. Hence, one interesting aspect of our algorithm is that depending upon the need user can pick up any solution.

The main focus of this work is three-fold, *viz.* (i) building a state-of-the-art anaphora resolution system for a resource-poor language like Bengali; (ii) adapting an existing state-of-the-art English co-reference resolution system for Bengali which has completely different orthography and characteristics; and (iii) multiobjective DE based feature selection technique to optimize features with respect to the evaluation metrics such as MUC, B^3, CEAFM, CEAFE and BLANC.

2 Mention Detection

Mention detection is an important component for anaphora resolution. We develop a mention detector based on the supervised machine learning algorithm, namely Conditional Random Field (CRF)[16]. The classifier is trained with the following set of features: Local contex within the previous two and next two tokens, Prefix and suffix strings of length upto three characters of the current token, Part-of-Speech (PoS) information of the current token, Named entity (NE) information (MUC categories like person, location and organization names) of the current token, noun phrase preceeding a pronoun, morphological constructs (*lemma* and *number information*) and several binary valued features. These binary valued features check whether the token is the first word of the sentence, whether the current token is a pronoun (e.g., *jeMon*[2], *kAro*, *tAhole*, *onnyoKe* etc.) that corresponds to non-anaphoric relations, whether it denotes a defin ite or demonstrative noun. In addition we prepare a list of frequently occurring suffixes that appear with the person names (e.g., *-bAbu*, *-der*, *-dI*, *-rA* etc.) and pronouns (e.g.,*-tI*, *-ke*, *-der* etc.), and define a feature that fires if the current word contains any of these suffixes. Evaluation results of this CRF based mention detector for the test data of ICON-2011 shared task on Anaphora Resolution in Indian Languages[3] are reported in Table 1.

Table 1. Results for mention detection on test data

Document id	precision	recall	F-measure
TestDoc-1	81.32	73.70	77.32
TestDoc-2	81.61	73.76	77.49
TestDoc-3	93.67	51.99	66.87

3 Pre-processing and Features of Anaphora Resolution

In this section we describe BART architecture and the features used for anaphora resolution.

[2] Bengali glosses are written in ITRANS notation.

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

3.1 Brief Description of BART System Architecture

We use BART [5] as our underlying platform for anaphora resolution. It provides the state-of-the-art approaches, including syntax-based and semantic features. The flexibility of BART is that its design is very modular, and this provides effective separation across several tasks, including engineering new features that exploit different sources of knowledge, and improving the way that anaphora resolution is mapped to a machine learning problem. BART has five main components: *preprocessing pipeline*, *mention factory*, *feature extraction module*, *decoder* and *encoder*.

3.2 Markable Extraction

We extract the mentions following the approach described in the previous section. Thereafter we convert the mentions to the particular format required in BART, namely MMAX2s standoff XML format.

3.3 Features for Anaphora Resolution

We view coreference resolution as a binary classification problem. Following similar proposals for English [2], we use the learning framework proposed in [1] 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 used to train a binary classifier. The classifier has to decide, given the features, whether the anaphor and the candidate are co-referent or not. Given BART's flexible architecture, we explore the contribution of some features implemented in BART for coreference resolution in Bengali. We also implement some features specific to the language concerned. Given a potential antecedent RE_i and a anaphor RE_j, we compute the following set of features. Subset of these features were implemented after being motivated from the prior works [10].

1. **String match**: The feature compares the surface forms, and takes the value true if the candidate anaphor (RE_j) and antecedent (RE_i) have the same surface string forms, otherwise false.
2. **Sentence distance**: This feature denotes the distance between the anaphor and antecedent. The value of this feature is non-negative integer that captures the distance in terms of the number of sentences between a anaphor and its antecedent. The feature takes the value of 0 if both anaphor and antecedent are in the same sentence, the value of 1 is produced if their sentence distance is 1 and so on.
3. **Markable distance**: This non-negative integer feature 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 true 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 preceding three sentences and is not the first person then the corresponding feature is also set to true. 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 is defined for the pair (RE_j, RE_i) that appears in the same quote. If RE_i is not the first person and RE_j corresponds to a second person then this feature is set to true. The feature also fires if RE_j is inside the quotation, but RE_i is outside and ends with the suffix "*ke*".
6. **Third person pronoun**: If both mentions in the pair (RE_j, RE_i) denote the third person prnouns and are outside the quotation then the feature fires.
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.
8. **Number agreement**: This feature checks whether the anaphor and antecedent pair agree in the number information. If both agree in the number then the feature value is set to true, otherwise false. This feature is extracted from the Indian language shallow parser[4].
9. **Semantic class feature**: If the semantic types of both RE_j and RE_i are same then the value of this feature is set to true, otherwise false. The semantic types denote the MUC named entity (NE) categories.
10. **Alias feature**: It checks whether RE_j is an alias of RE_i or not. The feature value is then set accordingly.
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 string subsequence kernel.
13. **Mention Type**: Following [1], 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 pairs denotes proper name.
14. **LeftRightMatch**: If RE_j is a prefix or suffix substring of RE_i or vice versa, then the value of this feature is set to true, otherwise it is false.

3.4 Learning Algorithm

In order to learn coreference decisions, we experiment with WEKA's [17] implementation of the C4.5 decision tree learning algorithm [18], with the features mentioned above. Training instances are created following [1]. Each pair of adjacent coreferent markables denote a positive training instance. A negative instance is created with the pairs of the anaphor and with any markable occurring between the anaphor and the antecedent.

3.5 Decoding

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

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

any preceding markable from right to left, until a pair labelled as coreferent is output, or the beginning of the document is reached. In the this step, the coreference chains are created by 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 Multiobjective Feature Selection for Coreference Resolution

4.1 Overview of Multiobjective Differential Evolution

Differential Evolution (DE) [19] is one of the popular evolutionary optimization techniques, and it performs a parallel direct search in complex, large and multi-modal landscapes, and provides near-optimal solutions. Parameters in the search space are encoded in the form of strings called chromosomes. A set of such strings is called a population denoted by NP. Each string denotes a D-dimensional parameter vector $X_{i,G} = [x_{1,i,G}, x_{2,i,G}, \ldots, x_{D,i,G}]$, $i = 1, 2, \ldots, NP$. The value of D represents the number of real parameters on which optimization or fitness function depends. For multiobjective version more than one objective or fitness function are associated with each string. Each of these fitness functions denotes the goodness of the string. The algorithm generates new parameter vectors by adding the weighted difference between two population vectors to a third vector, and this operation is called mutation. The param eters of the mutated vectors are mixed with the parameters of another predetermined vector, the target vector, to yield a new vector known as the trial vector. The process of parameter mixing is often referred to as crossover. Selection operation refers to the process of selecting the effective solutions. In this process the trial vectors are merged to the current population and then ranked based on the concept of domination and non-domination. In the next generation we select NP number of chromosomes from the ranked solutions using the crowding distance sorting algorithm. The process of selection, crossover and mutation continues for a fixed number of generations or till a termination condition is satisfied.

4.2 Problem Formulation

Suppose, there are D number of available features, and these are denoted by $F_1, \ldots, F_D$. Let, $\mathcal{A} = \{F_i : i = 1; D\}$. The problem of feature selection can then be stated as follows: Determine the appropriate subset of features $\mathcal{A}^{'} \subseteq \mathcal{A}$ such that when the concerned classifier is trained using these features should have optimized some metrics. In our proposed MOO based DE setting, we optimize five objective functions, namely the F-measure values of MUC, B^3, CEAFM, CEAFE and BLANC. All these metrics represent significantly different behaviours.

4.3 Problem Representation and Population Initialization

The features are encoded as binary valued strings in the chromosomes. Length of the chromosome is set equal to the number of features. The value of 1 in the i^{th} position

of a chromosome denotes the presence of the corresponding feature, and a value of 0 indicates that the respective feature does not participate while training the classifier.

4.4 Fitness Computation

The fitness computation corresponds to determining the values of the objective functions. If there are D features available in the chromosome, the classifier is trained only with these features. The trained model is evaluated on the development set. We compute the F-measure values for all the five objective functions that represent the evaluation scorers, namely MUC, B^3, CEAFM, CEAFE and BLANC. Our goal is to maximize these objective functions.

4.5 Mutation

In multiobjective DE, for each target vector $X_{i,G}$; $i = 1, 2, 3, \ldots, NP$, a mutant vector is generated according to

$$V_{i,G+1} = x_{r1,G} + F \times (x_{r2,G} - x_{r3,G}), \tag{1}$$

where $r1$, $r2$, $r3$ are mutually different random indices and belong to $\{1, 2, \ldots, NP\}$, G is the generation number and $F > 0$. The $r1$, $r2$ and $r3$ are chosen in such a way that they are different from the running current index i, so that the value of NP is at least equal to four. The parameter F controls the amplification of differential variation $(x_{r2,G} - x_{r3,G})$. Its value should be chosen within the range of $[0, 1]$. Here we set its value to 0.5. Mutated vector is denoted by $V_{i,G+1}$. After mutation operation if it is found that the value of $V_{i,G+1}$ is greater or equal to 0.5 then the value is projceted to 1, otherwise 0. A set of such NP mutant vectors is called the mutant population.

4.6 Crossover or Recombination

Crossover or recombination represents the parameter mixing of the target vector $X_{i,G}$ and mutant vector $V_{i,G+1}$. Exchange of information is performed in order to generate a better offspring that represents a promising solution. Diversity of the mutant vector can, thus, be increased. In order to perform this operation, a trial vector is formed as follows:

$$U_{i,G+1} = (u_{1,i,G+1}, u_{2,i,G+1}, \ldots, u_{D,i,G+1}) \tag{2}$$

where

$$u_{j,i,G+1} = v_{j,i,G+1} \text{ if } (r_j \leq CR) \text{ or } j = i_r \tag{3}$$

$$= x_{j,i,G} \text{ if } (r_j > CR) \text{ and } j \neq i_r \tag{4}$$

for $j = 1, 2, \ldots, D$,

In Equation 3, r_j is an uniform random number of the jth evaluation which belongs to $[0, 1]$. User should determine the value of the crossover constant CR that belongs to $[0, 1]$. Here we set its value to 0.5. An index, i_r that belongs to $\{1, 2, \ldots, D\}$, is chosen in such a way that it ensures that the parameters of $U_{i,G+1}$ gets at least one parameter from $V_{i,G+1}$. At the end of this process we obtain the trial population.

4.7 Selection

To select the best NP solutions for the next generation $G + 1$, trial population is merged to the current population, and this yields $2 \times NP$ chromosomes. These solutions are sorted based on the concept of domination and non-domination relations in the objective function space. As an example, the dominated and non-dominated relations are shown in Figure 1. In this figure non-dominated solutions (i.e. ranked solutions) are represented in the Pareto-optimal surface. Thereafter these ranked solutions are added to the population in the next generation until the number of solutions becomes less than or equal to NP. If the number of solutions exceeds NP, then crowding distance sorting algorithm is applied. This algorithm chooses the solutions starting from the beginning of the sorted rank solutions and keeps on including until it becomes equal to NP. This process ultimately determines the best NP chromosomes to be included in the next population.

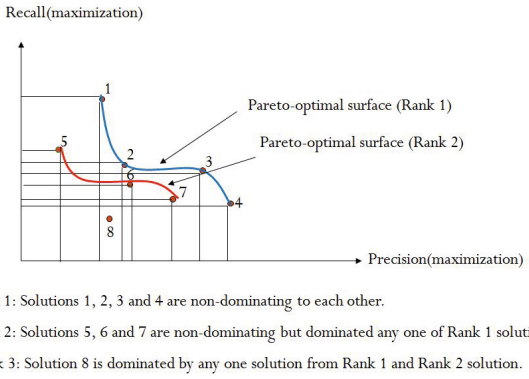

Fig. 1. Representation of dominated and non-dominated solutions

4.8 Termination Condition

The processes of mutation, crossover (or, recombination), fitness computation and selection are executed for a G_{Max} number of generations. Finally we obtain a set of non-dominated solutions on the final Pareto optimal front. Each of these solutions represents a set of (near)-optimal feature combinations.

4.9 Selecting the Best Solution

The MOO based feature selection yields a set of solutions on the Pareto optimal front. None of these solutions is better compared to the others, and therefore all are equally important from the algorithmic point of view. However we may have to select one solution at the end. Here we determine the final solution based on the F-measure value of the individual scores. We consider the top-ranked four solutions. For each of the five

objective functions, namely MUC, B^3, CEAFE, CEAFM and BLANC we select the particular solution that yields the highest F-measure value (for the respective metric) among the four solutions.

5 Experiments and Discussions

Our experiments are based on the datasets provided in the ICON NLP Tools Contest on Anaphora Resolution[5]. 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 the test dataset. Also we re-annotated all these three datasets to prefer longer coreference chains. The statistics of the datasets in terms of number of the sentences and number of tokens present in each set are provided in Table 2. The datasets are of mixed domains, covering *tourism*, *short story*, *news article* and *sports*.

Table 2. Statistics of the datasets

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

Table 3. Results of baseline and manual feature selection models

Scorers	**Manual Feature Selection**			**Baseline**		
	recall	**precision**	**F-measure**	**recall**	**precision**	**F-measure**
MUC	57.80	79.00	66.70	38.80	67.40	49.30
BCUB	51.02	71.27	59.47	27.09	72.95	39.51
CEAFM	49.83	49.83	49.83	31.27	31.27	31.27
CEAFE	48.88	23.58	31.81	48.24	16.8	24.92
BLANC	70.66	70.99	70.82	54.98	63.19	56.77

In order to compare with our proposed method we construct a baseline model using a loose re-implementation of a subset of features defined in [1]. These include *number agreement, alias, string matching, semantic class agreement, sentence distance and appositive* (c.f. Section 3.3). Results of this baseline model are shown in Table 3 that yields the F-measure values of 49.30%, 39.51%, 31.27%, 24.92% and 56.77% for MUC, B^3, CEAFM, CEAFE and BLANC, respectively. Thereafter we train the classifier with all the features as mentioned in Section 3.3. Results of this model are shown in Table 3 that shows the F-measure values of 66.70%, 59.47%, 49.83%, 31.81% and 70.82% for MUC, B^3, CEAFM, CEAFE and BLANC, respectively. Comparisons show that the performance obtained in this model are significantly higher tha n the baseline model.

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

Thereafter we apply our proposed multiobjective DE based feature selection method for determining the most relevant set of features for anaphora resolution. We optimize our algorithm based on the experiments that we perfomed on the developemnt data, and finally the best configuration is used for blind evaluation on the test data. The parameters of DE are set as follows: population size (NP) = 45, number of generations (G_Max) = 100, CR (probability of crossover) = 0.5 and F (mutation factor) = 0.5. The algorithm generates a set of solutions on the Pareto optimal front, and none these is strictly better than the others. We consider the solutions of the first rank, and finally select the best one following the technique as described in Section 4.9. It is to be noted that we select the optimized features from the four solutions of the first rank. The features, thus, selected are shown in Table 4. Detailed evaluation results when the classifier is trained wi th these four feature sets are presented in Table 5. The best performance achieved for each of these scorers correspond to 66.70%, 59.47%, 51.56%, 33.08% and 72.75% for MUC, B^3, CEAFM, CEAFE and BLANC, respectively. We observe performance improvements over the model developed with manual feature selection for all the metrics *except* MUC. The performance with respect to the B^3 metric does not improve, however, it is to be noted that we obtain the similar accuracy with a reduced feature set. This shows the effectiveness of the proposed feature selection technique. We also carried out experiments with the gold mentions, and this showed the F-measure values of 75.71%, 62.38%, 57.52%, 42.31% and 73.75% for MUC, B^3, CEAFM, CEAFE and BLANC, respectively.

Table 4. Optimized set of features. Here, the following abbreviations are used: 'SM': String match, 'SD': Sentence distance, 'MD': Markable distance, 'FPP': First person pronoun, 'SPP': Second person pronoun, 'TPP': Third person pronoun, 'RP': Reflexive pronoun, 'NA': Number agreement , 'SCF': Semantic class feature, 'AF': Alias feature, 'MT': Mention type, 'APF': Appositive feature, 'SK': String kernel, 'MT': Mention type,'LRM': LeftRightMatch, '$Rank_{1_{Soln}}$': Solutions of rank one, 'X': Denotes the presence of the corresponding feature.

$Rank_{1_{Soln}}$	LRM	SK	NA	FPP	SPP	TPP	RP	AF	SM	SCF	MT	APF	SD	MD
$Rank_{1.1}$		X		X	X	X	X	X	X	X	X		X	X
$Rank_{1.2}$	X	X	X	X	X	X	X	X		X	X			
$Rank_{1.3}$	X	X		X	X	X	X			X	X		X	X
$Rank_{1.4}$	X	X	X	X	X	X	X		X	X	X		X	X

Our statistical significance tests using ANOVA [20] show that the performance gains in our proposed model are actually significant. In order to perform this analysis we executed our algorithm three times. Comparisons with the works reported in the ICON-2011 shared tasks show that the performance achieved in our proposed model is better compared to the others for some of the metrics. In particular we obtain much higher accuracy for the MUC scorer. The performance obtained for the BLANC scorer is also at par the state-of-the-art method, and often better in few points over most of the works carried out thereafter. However, the performance for the other three scorers need further attention. The lower performance in these three metrics may be attributed to the fact that

Table 5. Optimized F-measure values for the first-ranked solutions. Here, the following abbreviations are used: 'H_{Val}': Highest F-measure values for the corresponding scorer.

$Rank_{1_{Soln}}$	MUC	BCUB	CEAFM	CEAFE	BLANC
$Rank_{1.1}$	65.90	59.10	**51.56**	31.89	**72.75**
$Rank_{1.2}$	**66.70**	**59.70**	49.83	31.81	70.82
$Rank_{1.3}$	66.06	58.32	50.90	**33.08**	71.37
$Rank_{1.4}$	65.96	58.51	50.84	33.04	71.38
H_{Val}	**66.70**	**59.70**	**51.56**	**33.08**	**72.75**

we re-annotated the training, development and test datasets to include the longer coreference chains. For example, the coreference pairs like (*SachIn*, *Se*), (*SachIn*, *tAr*) are merged into a single coreference chain like (*SachIn*, *Se*, *tAr*). In contrast in the original datasets of ICON-11 shared task these were treated as two separate instances. This is one of the possible explanations why the link-based metric(s) such as MUC exhibits better performance and the others suffer. The method proposed in [11] is developed based on the benchmark setup of ICON-2011. They developed three models and obtained the average F-measure values of 66.6%, 68.9% and 77.1% in these three systems, respectively. However it is to be noted that along with the ICON-11 datasets, they also used additional four documents that contain 4,923 tokens. Hence, the performance reported here can't be directly compared with the method proposed in [11]. The method proposed in [12] deals with a SOO based feature selection. As we have already mentioned, here we present a MOO based feature selection technique, which is a conceptually different from SOO. The performance figures obtained in the MOO based approach are higher compared to SOO, and these are achieved even with a set of relatively l ess number of features.

6 Conclusion

In this paper we propose a multiobjective DE based feature selection technique for anaphora resolution. The proposed model is evaluated for a resource-poor language like Bengali. We adapted BART, a state-of-the-art coreference resolution model originally developed for English for the task. Our feature selection model was developed by simultaneously optimizing five evaluation metrics, namely MUC, B^3, CEAFM, CEAFE and BLANC. We conducted our experiments on a benchmark dataset that was created as part of a shared task. Our proposed multiobjective DE based method attains significant performance gains over the baseline model and the model developed with all the available features. Comparisons show that our system achieves encouraging performance with respect to the available systems. In the current setting we used only decision tree as the machine learning algorithm. Experiments with other machine learning algorithms such as maximum entropy and support vector machine will be the another direction for future work. We would also like to concentrate on porting the systems to other Indian languages, e.g. Hindi and Telugu, and domains(e.g. biomedical texts).

References

1. Soon, W.M., Chung, D., Lim, D.C.Y., Lim, Y., Ng, H.T.: A machine learning approach to coreference resolution of noun phrases (2001)
2. Ng, V., Cardie, C.: Improving machine learning approaches to coreference resolution. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 104–111 (2002)
3. Walker, C., Strassel, S., Medero, J., Maeda, K.: Ace 2005 multilingual training corpus: Ldc2006t06 philadelphia penn.: Linguistic data consortium (2006)
4. Weischedel, R., Pradhan, S., Ramshaw, L., Palmer, M., Xue, N., Marcus, M., Taylor, A., Greenberg, C., Hovy, E., Belvin, R., Houston, A.: Ontonotes release 2.0:ldc2008t04 philadelphia penn.: Linguistic data consortium (2008)
5. Versley, Y., Ponzetto, S.P., Poesio, M., Eidelman, V., Jern, A., Smith, J., Yang, X., Moschitti, A.: Bart: A modular toolkit for coreference resolution. In: HLT-Demonstrations 2008 Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies, pp. 9–12 (2008)
6. Sobha, L., Patnaik, B.N.: Vasisth: An anaphora resolution system for indian languages. In: Proceedings Artificial and Computational Intelligence for Decision, Control and Automation in Engineering and Industrial Applications (ACIDCA), Monastir, Tunisia (2000)
7. Agarwal, S., Srivastava, M., Agarwal, P., Sanyal, R.: Anaphora resolution in hindi documents. In: Proceedings of Natural Language Processing and Knowledge Engineering (IEEE NLP-KE), Beijing, China (2007)
8. Uppalapu, B., Sharma, D.: Pronoun resolution for hindi. In: Proceedings of DAARC (2009)
9. Devi, S.L., Ram, V.S., Rao, P.R.: A generic anaphora resolution engine for indian languages. In: Proceedings of COLING 2014, pp. 1824–1833 (2014)
10. Sikdar, U., Ekbal, A., Saha, S., Uryupina, O., Poesio, M.: Adapting a state-of-the-art anaphora resolution system for resource-poor language. In: Proceedings of the Sixth International Joint Conference on Natural Language Processing, pp. 815–821. Asian Federation of Natural Language Processing (2013)
11. Senapati, A., Garain, U.: Guitar-based pronominal anaphora resolution in bengali. In: Proceedings of ACL, Sofia, Bulgaria (2013)
12. Sikdar, U.K., Ekbal, A., Saha, S., Uryupina, O., Poesio, M.: Differential evolution-based feature selection technique for anaphora resolution. Soft Computing, 1–13 (2014)
13. Uryupina, O.: Knowledge Acquisition for Coreference Resolution. PhD thesis, University of the Saarland (2007)
14. Hoste, V.: Optimization Issues in Machine Learning of Coreference Resolution. PhD thesis, Antwerp University (2005)
15. Saha, S., Ekbal, A., Uryupina, O., Poesio, M.: Single and multi-objective optimization for feature selection in anaphora resolution. In: Proceedings of the fifth International Joint Conference in Natural Langauge Processing (IJCNLP 2011), pp. 93–101 (2011)
16. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: ICML, pp. 282–289 (2001)
17. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann Publishers Inc., San Francisco (2005)
18. Quinlan, J.R.: Programs for machine learning. Morgan Kaufmann Publishers Inc., San Francisco (1993)
19. Storn, R., Price, K.: Differential evolution a simple and efficient heuristic for global optimization over continuous spaces. J. of Global Optimization 11(4), 341–359 (1997)
20. Anderson, T.W., Scolve, S.: Introduction to the Statistical Analysis of Data. Houghton Mifflin (1978)