Multi-metric optimization for coreference: The UniTN / IITP / Essex submission to the 2011 CONLL Shared Task

Olga Uryupina[‡] **Sriparna Saha**[†] **Asif Ekbal**[†] **Massimo Poesio**[*‡]
[‡]University of Trento
[†]Indian Institute of Technology Patna
[*] University of Essex
uryupina@gmail.com, sriparna@iitp.ac.in,
asif@iitp.ac.in, massimo.poesio@unitn.it

Abstract

Because there is no generally accepted metric for measuring the performance of anaphora resolution systems, a combination of metrics was proposed to evaluate submissions to the 2011 CONLL Shared Task (Pradhan et al., 2011). We investigate therefore Multi-objective function Optimization (MOO) techniques based on Genetic Algorithms to optimize models according to multiple metrics simultaneously.

1 Introduction

Many evaluation metrics have been proposed for anaphora resolution (Vilain et al., 1995; Bagga and Baldwin, 1998; Doddington et al., 2000; Luo, 2005; Recasens and Hovy, 2011). Each of these metrics seems to capture some genuine intuition about the the task, so that, unlike in other areas of HLT, none has really taken over. This makes it difficult to compare systems, as dramatically demonstrated by the results of the Coreference Task at SEMEVAL 2010 (Recasens et al., 2010). It was therefore wise of the CONLL organizers to use a basket of metrics to assess performance instead of a single one.

This situation suggests using methods to optimize systems according to more than one metric at once. And as it happens, techniques for doing just that have been developed in the area of Genetic Algorithms—so-called **multi-objective optimization** techniques (MOO) (Deb, 2001). The key idea of our submission is to use MOO techniques to optimize our anaphora resolution system according to three metrics simultaneously: the MUC scorer (a member of what one might call the 'link-based' cluster of metrics) and the two CEAF metrics (representative of the 'entity-based' cluster). In a previous study (Saha et al., 2011), we show that our MOO-based approach yields more robust results than single-objective optimization.

We test two types of optimization: feature selection and architecture–whether to learn a single model for all types of anaphors, or to learn separate models for pronouns and for other nominals. We also discuss how the default mention extraction techniques of the system we used for this submission, BART (Versley et al., 2008), were modified to handle the all-mention annotation in the OntoNotes corpus.

In this paper, we first briefly provide some background on optimization for anaphora resolution, on genetic algorithms, and on the method for multi-objective optimization we used, Non-Dominated Sorting Genetic Algorithm II (Deb et al., 2002). After that we discuss our experiments, and present our results.

2 Background

2.1 Optimization for Anaphora Resolution

There have only been few attempts at optimization for anaphora resolution, and with a few exceptions, this was done by hand.

The first systematic attempt at automatic optimization of anaphora resolution we are aware of was carried out by Hoste (2005), who used genetic algorithms for automatic optimization of both feature selection and of learning parameters, also considering

Proceedings of the 15th Conference on Computational Natural Language Learning: Shared Task, pages 61–65,
Portland, Oregon, 23-24 June 2011.

two different machine learners, TimBL and Ripper. Her results suggest that such techniques yield improvements on the MUC-6/7 datasets. Recasens and Hovy (2009) carried out an investigation of feature selection for Spanish using the ANCORA corpus.

A form of multi-objective optimization was applied to coreference by Munson et al. (2005). Munson et al. (2005) did not propose to train models so as to simultaneously optimize according to multiple metrics; instead, they used ensemble selection to learn to choose among previously trained models the best model for each example. Their general conclusion was negative, stating that "ensemble selection seems too unreliable for use in NLP", but they did see some improvements for coreference.

2.2 Genetic Algorithms

Genetic algorithms (GAs) (Goldberg, 1989) are randomized search and optimization techniques guided by the principles of evolution and natural genetics. In GAs the parameters of the search space are encoded in the form of strings called *chromosomes*. A collection of such strings is called a *population*. An *objective* or *fitness* function is associated with each chromosome that represents the degree of *goodness* of that chromosome. A few of the chromosomes are selected on the basis of the principle of survival of the fittest, and assigned a number of copies that go into the mating pool. Biologically inspired operators like *crossover* and *mutation* are applied on these chromosomes to yield a new generation of strings. The processes of selection, crossover and mutation continues for a fixed number of generations or till a termination condition is satisfied.

2.3 Multi-objective Optimization

Multi-objective optimization (MOO) can be formally stated as follows (Deb, 2001). Find the vectors $\overline{x}^* = [x_1^*, x_2^*, \ldots, x_n^*]^T$ of decision variables that simultaneously optimize the *M* objective values

$$\{f_1(\overline{x}), f_2(\overline{x}), \ldots, f_M(\overline{x})\}$$

while satisfying the constraints, if any.

An important concept in MOO is that of **domination**. In the context of a maximization problem, a solution $\overline{x_i}$ is said to dominate $\overline{x_j}$ if $\forall k \in 1, 2, \ldots, M,\ f_k(\overline{x_i}) \geq f_k(\overline{x_j})$ and $\exists k \in 1, 2, \ldots, M,$ such that $f_k(\overline{x_i}) > f_k(\overline{x_j})$.

Genetic algorithms are known to be more effective for solving MOO than classical methods such as weighted metrics, goal programming (Deb, 2001), because of their population-based nature. A particularly popular genetic algorithm of this type is NSGA-II (Deb et al., 2002), which we used for our runs.

3 Using MOO for Optimization in Anaphora Resolution

We used multi-objective optimization techniques for feature selection and for identifying the optimal architecture for the CONLL data. In this section we briefly discuss each aspect of the methodology.

3.1 The BART System

For our experiments, we use BART (Versley et al., 2008), a modular toolkit for anaphora resolution that supports state-of-the-art statistical approaches to the task and enables efficient feature engineering. BART comes with a set of already implemented features, along with the possibility to design new ones. It also implements different models of anaphora resolution, allowing the choice between single and split classifiers that we explore in our runs, as well as between mention-pair and entity-mention, and between best-first and ranking. It also has interfaces to different machine learners (MaxEnt, SVM, decision trees). It is thus ideally suited for experimenting with feature selection and other aspects of optimization. However, considering all the parameters, it was unfeasible to run an optimization on the amount of data available on CONLL; we focused therefore on feature selection and the choice between single and split classifiers. We considered 42 features, including 7 classifying mention type, 8 for string matching of different subparts and different levels of exactness, 2 for aliasing, 4 for agreement, 12 for syntactic information including also binding constraints, 3 encoding salience, 1 encoding patterns extracted from the Web, 3 for proximity, and 2 for 1st and 2nd person pronouns. Again because of time considerations, we used decision trees as implemented in Weka as our classification model instead of maximum-entropy or SVMs. Finally, we used a simple mention-pair model without ranking as in (Soon et al., 2001).

3.2 Mention detection

BART supports several solutions to the mention detection (MD) task. The users can input precomputed mentions, thus, experimenting with *gold* boundaries or *system* boundaries computed by external modules (e.g., CARAFE). BART also has a built-in mention extraction module, computing boundaries heuristically from the output of a parser.

For the CoNLL shared task, we use the BART internal MD module, as it corresponds better to the mention detection guidelines of the OntoNotes dataset. We have further adjusted this module to improve the MD accuracy. The process of mention detection involves two steps.

First, we create a list of *candidate mentions* by merging basic NP chunks with named entities. NP chunks are computed from the parse trees provided in the CoNLL distribution, Named entities are extracted with the Stanford NER tool (Finkel et al., 2005). For each candidate mention, we store it minimal and maximal span. The former is used for computing feature values (e.g., for string matching); it corresponds to either the basic NP chunk or the NE, depending on the mention type. The latter is used for alignment with CoNLL mentions; it is computed by climbing up the parse tree.

This procedure, combined with the perfect (gold) coreference resolution, gives us an F-score of 91.56% for the mention detection task on the CoNLL development set[1].

At the second step, we aim at discarding mentions that are unlikely to participate in coreference chains. We have identified several groups of such mentions: erroneous (”[uh]”), (parts of) multiword expressions (”for [example]”), web addresses, emails (”[http://conll.bbn.com]”), time/date expressions (”two times [a year]”), non-referring pronouns (”[there]”,”[nobody]”), pronouns that are unlikely to participate in a chain (”[somebody]”, ”[that]”), time/date expressions that are unlikely to participate in a chain (”[this time]”), and expletive ”it”.

Our experiments on the development data show that the first five groups can be reliably identified and safely discarded from the processing: even with the perfect resolution, we observe virtually no performance loss (the F-score for our MD module with the gold coreference resolution remains at 91.45% once we discard mentions from groups 1-5).

The remaining groups are more problematic: when we eliminate such mentions, we see performance drops with the gold resolution. The exact impact of discarding those mentions can only be assessed once we have trained the classifier.

In practice, we have performed our optimization experiments, selected the best classifier and then have done additional runs to fine-tune the mention detection module.

3.3 Using NSGA-II

Chromosome Representation of Feature and Architecture Parameters We used chromosomes of length 43, each binary gene encoding whether or not to use a particular feature in constructing the classifier, plus one gene set to 1 to use a split classifier, 0 to use a single classifier for all types of anaphors.

Fitness Computation and Mutations For fitness computation, the following procedure is executed.

1. Suppose there are N number of features present in a particular chromosome (i.e., there are total N number of 1's in that chromosome).

2. Construct the coreference resolution system (i.e., BART) with only these N features.

3. This coreference system is evaluated on the development data. The recall, precision and F-measure values of three metrics are calculated.

For MOO, the objective functions corresponding to a particular chromosome are $F_1 =$ F-measure$_{MUC}$ (for the MUC metric), $F_2 =$ F-measure$_{\phi_3}$ (for CEAF using the ϕ_3 entity alignment function (Luo, 2005)) and $F_3 =$ F-measure$_{\phi_4}$ (for CEAF using the ϕ_3 entity alignment function). The objective is to: $max[F_1, F_2, F_3]$: i.e., these three objective functions are simultaneously optimized using the search capability of NSGA-II.

We use crowded binary tournament selection as in NSGA-II, followed by conventional crossover and mutation for the MOO based optimization. The most characteristic part of NSGA-II is its elitism operation, where the non-dominated solutions (Deb,

[1]Note that, due to the fact that OntoNotes guidelines exclude singleton mentions, it is impossible to evaluate the MD component independently from coreference resolution.

2001) among the parent and child populations are propagated to the next generation. The near-Pareto-optimal strings of the last generation provide the different solutions to the feature selection problem.

Genetic Algorithms Parameters Using the CONLL development set, we set the following parameter values for MOO (i.e., NSGA-II): population size=20, number of generations=20, probability of mutation=0.1 and probability of crossover=0.9.

3.4 Running the Optimization

Considering the size of the OntoNotes corpus, it would be very time-consuming to run an optimization experiment on the whole dataset. We have therefore split the data into 3 sub-samples and performed separate MOO experiments on each one.

The MOO approach provides a set of non-dominated solutions on the final Pareto optimal front. All the solutions are equally important from the algorithmic point of view. We have collected sets of chromosomes for each sub-sample and evaluated them on the whole train/development set, picking the solution with the highest FINAL[2] score for our CoNLL submission.

4 Results

4.1 Development set

Table 1 compares the performance level obtained using all the features with that of loose re-implementations of the systems proposed by Soon et al. (2001) and Ng and Cardie (2002), commonly used as baselines. Our reimplementation of the Ng & Cardie model uses only a subset of features.

The results in Table 1 show that our system with a rich feature set does not outperform simpler baselines (and, in fact, yields poorer results). A similar trend has been observed by Ng and Cardie (2002), where the improvement was only possible after manual feature selection.

The last line of Table 1 shows the performance level of the best chromosome found through the MOO technique. As it can be seen, it outperforms all the baselines according to all the measures, leading to an improvement of 2-5 percentage points in the FINAL score.

[2]The FINAL score is an average of F_{MUC}, F_{B3} and F_{CEAFE}.

This suggests that automatic feature selection is essential to improve performance – i.e., that an efficient coreference resolution system should combine rich linguistic feature sets with automatic feature selection mechanisms.

4.2 Test set

We have re-trained our best solution on the combined train and development set, running it on the test data. This system has showed the following performance in the official evaluation (open track): the FINAL score of 54.32, $F_{MUC} = 57.53\%$, $F_{B3} = 65.18\%$, $F_{CEAFE} = 40.16\%$.

5 Conclusion

Our results on the development set suggest that a linguistically-rich system for coreference resolution might benefit a lot from feature selection. In particular, we have investigated Non-Dominated Sorting Genetic Algorithm II (Deb et al., 2002) for multi-objective optimization.

In subsequent work, we plan to expand the optimization technique to consider also learning parameters optimization, classifier selection, and learning model selection.

Acknowledgments

This work was in part supported by the Provincia di Trento Grande Progetto LiveMemories, in part by an Erasmus Mundus scholarship for Asif Ekbal and Sriparna Saha.

Features	F_{MUC}	F_{CEAFE}	F_{B3}	FINAL
following Soon et al. (2001)	54.12	41.08	66.67	53.42
-*-, with splitting	53.81	41.03	66.70	53.31
following Ng & Cardie (2002)	52.97	42.40	66.18	53.31
-*-, with splitting	53.28	40.46	66.03	52.72
All features	50.18	38.54	63.79	50.33
-*-, with splitting	50.19	39.47	65.38	51.16
Optimized feature set (splitting)	57.05	42.61	67.46	55.15

Table 1: Performance on the development set

References

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

Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):181–197.

Kalyanmoy Deb. 2001. *Multi-objective Optimization Using Evolutionary Algorithms*. John Wiley and Sons, Ltd, England.

G. Doddington, A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassell, and R. Weischedel. 2000. The automatic content extraction (ACE) program–tasks, data, and evaluation. In *Proc. of LREC*.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 363–370.

D. E. Goldberg. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, New York.

Veronique Hoste. 2005. *Optimization Issues in Machine Learning of Coreference Resolution*. Ph.D. thesis, Antwerp University.

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

Art Munson, Claire Cardie, and Rich Caruana. 2005. Optimizing to arbitrary NLP metrics using ensemble selection. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pages 539–546.

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

Sameer Pradhan, Lance Ramshaw, Mitchell Marcus, Martha Palmer, Ralph Weischedel, and Nianwen Xue. 2011. Conll-2011 shared task: Modeling unrestricted coreference in ontonotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning (CoNLL 2011)*, Portland, Oregon, June.

M. Recasens and E. Hovy. 2009. A deeper look into features for coreference resolution. In S. Lalitha Devi, A. Branco, and R. Mitkov, editors, *Anaphora Processing and Applications (DAARC 2009*, number 5847 in LNAI, pages 29–42, Berlin / Heidelberg. Springer-Verlag.

M. Recasens and E. Hovy. 2011. Blanc: Implementing the rand index for coreference evaluation. *Natural Language Engineering*.

M. Recasens, L. Màrquez, E. Sapena, M. A. Martí, M. Taulé, V. Hoste, M. Poesio, and Y. Versley. 2010. Semeval-2010 task 1: Coreference resolution in multiple languages. In *Proc. SEMEVAL 2010*, Uppsala.

Sriparna Saha, Massimo Poesio, Asif Ekbal, and Olga Uryupina. 2011. Single and multi-objective optimization for feature selection in anaphora resolution. Submitted.

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

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 *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies*, pages 9–12.

M. Vilain, J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proc. of the Sixth Message Understanding Conference*, pages 45–52.