Coreference Resolution with Loose Transitivity Constraints

Xinxin Li, Xuan Wang, Shuhan Qi
Shenzhen Graduate School
Harbin Institute of Technology, ShenZhen, China
lixxin2@gmail.com, wangxuan@insun.hit.edu.cn
shuhan_qi@qq.com

Abstract

Our system treats coreference resolution as an integer linear programming (ILP) problem. Extending Denis and Baldridge (2007) and Finkel and Manning (2008)'s work, we exploit loose transitivity constraints on coreference pairs. Instead of enforcing transitivity closure constraints, which brings $O(n^3)$ complexity, we employ a strategy to reduce the number of constraints without large performance decrease, i.e., eliminating coreference pairs with probability below a threshold θ. Experimental results show that it achieves a better performance than pairwise classifiers.

1 Introduction

This paper describes our coreference resolution system participating in the close track of CoNLL 2011 shared task (Pradhan et al., 2011). The task aims to identify all mentions of entities and events and cluster them into equivalence classes in OntoNotes Corpus (Pradhan et al., 2007a). During the last decade, several machine learning methods for coreference resolution have been developed, from local pairwise classifiers (Soon et al., 2001) to global learning methods (Luo et al., 2004; Ng, 2005; Denis and Baldridge, 2007), from simple morphological, grammatical features to more liguistically rich features on syntactic structures and semantic relations (Pradhan et al., 2007b; Haghighi and Klein, 2009).

Our system supports both local classifiers and global learning. Maximum entropy model is used for anaphoricity and coreference, because it assigns probability mass to mentions and coreference pairs directly. In global phase, instead of determining each coreference pair independently in a greedy fashion, we employ an integer linear programming (ILP) formulation for this problem. Extending (Denis and Baldridge, 2007) and (Finkel and Manning, 2008)'s work, we introduce a loose selection strategy for transitivity constraints, attempting to overcome huge computation complexity brought by transitivity closure constraints. Details are described in section 2.3.

2 System Description

2.1 Mention Detection

Mention detection is a method that identifies the anaphoricity and non-anaphoricity mentions before coreference resolution. The non-anaphoric mentions usually influence the performance of coreference resolution as noises. Coreference resolution can benefit from accurate mention detection since it might eliminate the non-anaphoric mentions. We take mention detection as the first step, and then combine coreference classifier into one system.

Total 70 candidate features are used for mention detection, including lexical, syntactic, semantic features (Ng and Cardie, 2002). Features are selected according to the information gain ratio (Han and Kamber, 2006)

$$GainRation(A) = \frac{Gain(A)}{SplitInfo(A)}$$

The top 10 features with highest gain ratio are: string match, head word match, all uppercase, pronoun, starting with article, number, following preposition, nesting in verb phrase, nesting in preposition,

Proceedings of the 15th Conference on Computational Natural Language Learning: Shared Task, pages 107–111,
Portland, Oregon, 23-24 June 2011. ©2011 Association for Computational Linguistics

and starting with definite article. Many string features that cannot be calculated by gain ratio method are also added.

2.2 Coreference Determination

For coreference determination, we first build several baseline systems with different training instance generation methods and clustering algorithms. These strategies are shown below. Detailed description can be found in Ng (2005).

- training instance generation methods: McCarthy and Lehnerts method, Soon et al.'s method, Ng and Cardie's method.
- clustering algorithms: closest-first clustering, best-first clustering, and aggressive merge clustering.

Overall 65 features are considered in our system. Features are extracted from various linguistic information, including:

- distance: sentence distance, minimum edit distance (Strube et al., 2002)
- lexical: string match, partial match, head word match (Daumé III and Marcu, 2005)
- grammar: gender agreement, number agreement(Soon et al., 2001)
- syntactic: same head, path (Yang et al., 2006)
- semantic: semantic class agreement, predicate (Ponzetto and Strube, 2006; Ng, 2007)

Combining different training instance generation methods and clustering algorithms, we get total 9 baseline systems. For each system, we use a greedy forward approach to select features. Starting from a base feature set (Soon et al., 2001), each feature out of the base set is added one by one according to the performance change on development data. Finally, the procedure is ended until the performance is not improved. The baseline system with best performance is selected for further improvement.

2.3 ILP with Loose Transitivity Constraints

Previous systems usually take coreference resolution as binary classification problem, and build the coreference chain by determining each coreference pair indepedently. The binary classifier is easily implemented, but may cause inconsistency between coreference pairs. Several work have been developed to overcome the problem, e.g., Bell trees (Luo et al., 2004), conditional random fields (McCallum and Wellner, 2004) and reranker (Ng, 2005).

Denis and Baldridge (2007) proposed an ILP formulation to find the optimal solution for the problem. It utilizes the output of other local classifiers and performs global learning. The objective function for their conference-only model takes the form:

$$min \sum_{\langle i,j \rangle \in M^2} c_{\langle i,j \rangle} * x_{\langle i,j \rangle} + \bar{c}_{\langle i,j \rangle} * (1 - x_{\langle i,j \rangle})$$

where $c_{\langle i,j \rangle} = -\log(P_C)$, $\bar{c}_{\langle i,j \rangle} = -\log(1 - P_C)$. M is the candidate mention set for each document. P_C refers to the probability of coreference link between two mentions produced by our maximum entropy model, and $x_{\langle i,j \rangle}$ is a binary variable that is set to 1 if two mentions are coreferent, 0 otherwise.

However, as Finkel and Manning showed, D&B's coreference-only model without transitivity constraints is not really necessary, because they only select the coreference links with probability $P_C > 0.5$. Klenner (2007) and Finkel and Manning (2008)'s work extended the ILP framework to support transitivity constraints. The transitivity constraints are formulated as

$$\forall i,j,k \in M (i < j < k)$$
$$x_{\langle i,j \rangle} \geq x_{\langle j,k \rangle} + x_{\langle i,k \rangle} - 1$$
$$x_{\langle j,k \rangle} \geq x_{\langle i,j \rangle} + x_{\langle i,k \rangle} - 1$$
$$x_{\langle i,k \rangle} \geq x_{\langle i,j \rangle} + x_{\langle j,k \rangle} - 1$$

These constraints ensure that when any two corefrent links (e.g., $x_{\langle i,j \rangle}$, $x_{\langle i,k \rangle}$) among three mentions exist, the third one $x_{\langle j,k \rangle}$ must also be a link. However, these constraints also bring huge time and space complexity with n^3 constraints (n is number of candidate mention set M, which is larger than 700 in some documents), and cannot be solved in a restricted time and memory environment. We introduce a loose method to eliminate conference links

Ratio	Recall	Precision	F-value
0.4	**84.03**	43.75	57.54
0.6	70.6	70.85	**70.72**
0.8	64.24	74.35	68.93
1.0	58.63	**76.13**	66.25

Table 1: Results of mention dection

below a probability threshold θ. The constraints are transformed as

$$x_{\langle i,k \rangle} + x_{\langle j,k \rangle} \leq 1 \qquad (1)$$

$$x_{\langle i,j \rangle} = 0 \qquad (2)$$

when $P_C(i,j) < \theta$. The threshold θ is tuned on development data for faster computation without large performance decrease.

3 Experiments and Analysis

In the paper we mainly take noun phrases (NPs) and pronouns as candidate mentions, and ignore other phrases since more than 91% of the mentions are NPs and pronouns.

3.1 Mention Detection

We observe that the ratio of positive examples and negative examples is about 1:3 in training data. To balance the bias, we propose a ratio control method which sets a ratio to limit the number of negative examples. Our system will select all positive examples, and part of negative examples according to the ratio. By tuning the ratio, we can control the proportion of positive and negative examples. With different ratios for negative feature selection, the results on development data are shown in table 1.

From table 1, we can see that as the ratio increases, recall becomes smaller and precision becomes larger. Small threshold means less negative examples are generated in training procedure, and the classifier tends to determine a mention as positive. Finally, we choose the ratio 0.6 for our model because it gets the best F-value on the development data.

3.2 Coreference Resolution

Our system participates in the close track with auto mention and gold boundary annotation. The the performance is evaluated on MUC, B-CUBED, CEAF(M), CEAF(E), BLANC metrics. The official metric is calculated as $^{(MUC+B^3+CEAF)}/_3$.

TIGM	Soon	Soon	Soon	Ng
CA	A	B	C	B
MUC	44.29	**46.18**	**46.18**	45.33
B^3	59.76	**61.39**	60.03	60.93
CEAF(M)	42.77	**44.43**	43.01	44.41
CEAF(E)	35.77	36.37	36.08	**36.54**
BLANC	60.22	63.94	59.9	**63.96**
Official	46.6	**47.98**	46.76	47.6

Table 2: Results of baseline systems

Table 2 summarizes the performance of top 4 of 9 baseline systems with different training instance generation methods and clustering algorithms on development data. In the table, TIGM means training instance generation method, and CA denotes clustering algorithm, which includes C as closest-first, B as best-first, and A as aggressive-merge clustering algorithm. The results in Table 2 show that the system with Soon's training instance generation method and best-first clustering algorithm achieves the best performance. We take it as baseline for further improvement.

In ILP model, we perform experiments on documents with less than 150 candidate mentions to find the suitable probability threshold θ for loose transitivity constraints. There are totol 181 documents meeting the condition in development data. We take two strategies to loose transitivity constraints: (I) formula 1 and 2, and (II) formula 2 only. Glpk package is used to solve our ILP optimization problems.[1]

Table 3 shows that as threshold θ increases, the running time reduces dramatically with a small performance decrease from 49.06 to 48.88. Strategy I has no benefit for the performance. Finally strategy II and $\theta = 0.06$ are used in our system.

We also combine mentions identified in first phase into coreference resolution. Two strategies are used: feature model and cascaded model. For feature model, we add two features which indicate whether the two candidate mentions of a coreference pair are mentions identified in first phase or not. For cascaded model, we take mentions identified in first phase as inputs for coreference resolution. For ILP

[1] http://www.gnu.org/software/glpk/

θ	0	0.02	0.02	0.04	0.04	0.06	0.06	0.08	0.08	0.1	0.1
Strategy		I	II	I	II	I	II	I	II	I	II
MUC	40.95	40.64	40.92	40.64	40.83	40.64	40.8	40.64	40.75	40.64	40.68
B^3	65.6	65.47	65.59	65.47	65.58	65.47	65.57	65.47	65.5	65.47	65.49
CEAF(M)	48.62	48.39	48.59	48.39	48.56	48.39	48.54	48.39	48.42	48.39	48.39
CEAF(E)	40.62	40.47	40.62	40.47	40.63	40.47	40.61	40.47	40.5	40.47	40.47
BLANC	61.87	61.76	61.85	61.76	61.84	61.76	61.83	61.76	61.79	61.76	61.78
Official	49.06	48.88	49.04	48.88	49.01	48.88	48.99	48.88	48.92	48.88	48.88
Time(s)	1726	1047	913	571	451	361	264	253	166	153	109

Table 3: Results on different probability thresholds and strategies

Model	Feature	Cascade	ILP
MUC	41.08	47.41	45.89
B^3	59.74	57.67	61.85
CEAF(M)	41.9	42.04	44.52
CEAF(E)	34.72	32.33	36.85
BLANC	61.1	62.99	63.92
Official	45.18	45.81	**48.19**

Table 4: Results of coreference resolution systems.

Data	Dev	Dev	Test	Test
Mention	Auto	Gold	Auto	Gold
MUC	45.89	46.75	46.62	44.00
B^3	61.85	61.48	61.93	57.42
CEAF(M)	44.52	45.17	44.75	42.36
CEAF(E)	36.85	37.19	36.83	34.22
BLANC	63.92	63.83	64.27	62.96
Official	48.19	**48.47**	**48.46**	45.21

Table 5: Results for development and test data

model, we perform experiments on coreference-only system with our loose transitivity constraints. The results on development data are shown in Table 4.

In Core Quad 2.40G CPU and 2G memory machine, our ILP model can optimize one document per minute on average. From table 4, we can see that the ILP model achieves the best F-value, implying the benefit of our algorithm. It also shows that traditional coreference resolution methods combining mention detection decrease the performance. For restricted time deadline, other constraints strategies (Klenner, 2007) and joint anaphoricity-coreference ILP model are not used in our system. It would be in our future work.

3.3 Test

Table 5 shows the performance of our system for both development and test data, with auto mention and gold boundary annotation.

The results in table 5 show that in auto mention annotation, the performance on test data is a little bit better than development data. The reason might be that the system on test data uses more data to train, including development data. A phenomenon surprises us is that the performance on test data with gold annotation is less than on development data, even than auto annotation. It turns out that the mistake is made because we confuse the the definition of gold bourdaries as gold mentions, which are ”all” and ”only” mentions in coreference chains.

4 Conclusion

In this paper, we present a coreference resolution system which employs an ILP formulation for global optimization. To reduce computation complexity, our system employs loose transitivity constraints to the ILP model. Experimental results show that it achieves a better performance than pairwise classifiers.

References

Hal Daumé III and Daniel Marcu. 2005. A large-scale exploration of effective global features for a joint entity detection and tracking model. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 97–104, Vancouver, British Columbia, Canada, October. Association for Computational Linguistics.

Pascal Denis and Jason Baldridge. 2007. Joint determination of anaphoricity and coreference resolution using integer programming. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 236–243, Rochester, New York, April. Association for Computational Linguistics.

Jenny Rose Finkel and Christopher D. Manning. 2008. Enforcing transitivity in coreference resolution. In *Proceedings of ACL-08: HLT, Short Papers*, pages 45–48, Columbus, Ohio, June. Association for Computational Linguistics.

Aria Haghighi and Dan Klein. 2009. Simple coreference resolution with rich syntactic and semantic features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1152–1161, Singapore, August. Association for Computational Linguistics.

J. Han and M. Kamber. 2006. *Data mining: concepts and techniques*. Morgan Kaufmann.

Manfred Klenner. 2007. Enforcing consistency on coreference sets. In *Recent Advances in Natural Language Processing (RANLP)*, pages 323–328.

Xiaoqiang Luo, Abe Ittycheriah, Hongyan Jing, Nanda Kambhatla, and Salim Roukos. 2004. A mention-synchronous coreference resolution algorithm based on the bell tree. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 135–142, Barcelona, Spain, July.

Andrew McCallum and Ben Wellner. 2004. Conditional models of identity uncertainty with application to noun coreference. In *NIPS 2004*.

Vincent Ng and Claire Cardie. 2002. Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution. In *Proceedings of the 19th international conference on Computational linguistics - Volume 1*, COLING '02, pages 1–7, Stroudsburg, PA, USA. Association for Computational Linguistics.

Vincent Ng. 2005. Machine learning for coreference resolution: From local classification to global ranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 157–164, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Vincent Ng. 2007. Shallow semantics for coreference resolution. In *Proceedings of IJCAI*, pages 1689–1694.

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, Eduard Hovy, Mitch Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2007a. Ontonotes: A unified relational semantic representation. *International Journal of Semantic Computing (IJSC)*, 1(4):405–419.

Sameer Pradhan, Lance Ramshaw, Ralph Weischedel, Jessica MacBride, and Linnea Micciulla. 2007b. Unrestricted coreference: Identifying entities and events in ontonotes. In *in Proceedings of the IEEE International Conference on Semantic Computing (ICSC)*, September 17-19.

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.

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

Michael Strube, Stefan Rapp, and Christoph Müller. 2002. The influence of minimum edit distance on reference resolution. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*, EMNLP '02, pages 312–319, Stroudsburg, PA, USA. Association for Computational Linguistics.

Xiaofeng Yang, Jian Su, and Chew Lim Tan. 2006. Kernel-based pronoun resolution with structured syntactic knowledge. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 41–48, Sydney, Australia, July. Association for Computational Linguistics.