Unsupervised Coreference Resolution Using a Graph Labeling Approach

Nafise Sadat Moosavi(✉) and GholamReza GhassemSani

Sharif University of Technology, Tehran, Iran
n_moosavi@ce.sharif.edu, sani@sharif.edu

Abstract. In this paper, we present a new unsupervised coreference resolution method, that models coreference resolution as a graph labeling problem. The proposed approach uses an incremental graph development method that hierarchically deploys coreference features from higher precision to lower ones. Then, a relaxation labeling method is used for solving the graph labeling problem.

Keywords: Unsupervised coreference resolution · Graph labeling · Relaxation labeling · Hierarchical graph development

1 Introduction

Coreferences are relations that hold between expressions which refer to the same entities. Expressions are often called mentions of entities. A coreference is a reflexive, symmetric, and transitive equivalence relation. The reflexive and transitive closure over coreference relations generates equivalence classes of mentions, which are called coreference chains.

Coreference resolution is equivalent to the set partitioning problem in which the search space is the set of all mutually disjoint subsets of mentions. From a language engineering perspective, the accurate identification of the entities that are referred to is an important challenge. Numerous natural language processing (NLP) tasks such as information extraction, question answering, automatic summarization, machine translation, and natural language generation can benefit from availability of a coreference resolution system.

By using coreference information, we can construct a graph G in which each mention is a vertex, and each coreference relation forms an edge between corresponding vertices. In this way, coreference resolution can be formulated as a graph labeling problem: all vertices with the same label are considered to be in a same coreference chain.

Several earlier works modeled coreference resolution as a graph labeling or graph partitioning problem [1–5]. In this paper, we present a new unsupervised coreference resolution system, which also casts coreference resolution as a graph labeling problem. It employs relaxation labeling method for labeling assignment.

Z. Vetulani and J. Mariani (Eds.): LTC 2011, LNAI 8387, pp. 93–103, 2014.
DOI: 10.1007/978-3-319-08958-4_8

The presented system has been inspired by the success of two successful coreference systems [4,6]. However, our system deploys a new hierarchical graph construction method for developing the adjacency graph.

The hierarchical graph construction method helps the labeling algorithm to just consider the most reliable set of neighbors when it tries to label mentions at each pass of the algorithm.

2 Related Work

Recently, the accessibility of annotated coreference data (MUC conferences and ACE evaluations) has brought up the deployment of a wide variety of supervised machine learning approaches for the problem of coreference resolution [7]. The focus of statistical approaches to coreference resolution has been moved from attainment of simple pairwise models (which determine whether two mentions are referring to the same entity [8,9]) to the use of rich linguistic features [10,11], and utilization of advanced learning techniques [12].

Some recent works on coreference resolution have shown that a rich feature set that can model lexical, syntactic, semantic, and discourse aspects of mentions is essential for the success of the coreference task [13–15]. When these rich features are combined with the complexity of coreference models, supervised approaches will be more dependent on annotated data and less appropriate for languages with insufficient or no annotated data.

Because of the increasing importance of multilingual processing in NLP community, developing unsupervised or semi-supervised methods for automatic processing of languages with limited resource has become more essential.

Unsupervised learning methods totally eliminate the need to annotated data, and remarkably, recent unsupervised coreference resolution methods compete with their supervised counterparts [6,7,14–16].

Motivated in part by such observations, in this paper, we present a new unsupervised model for coreference resolution. We model unsupervised coreference resolution as a graph labeling problem which is solved by a relaxation labeling algorithm. The proposed method has been inspired by the success of two earlier coreference systems [17,18], and it benefits from some advantages of both approaches. Sapena et al. [17] can be considered as a supervised counterpart of our approach, and Raghunathan et al. [18] is a rule-based system that deploys coreference features in a sieve architecture. The sieve architecture allows more precise features to be considered before low precision features in coreference decisions. Therefore, the decisions made based on more precise features will not be affected by lower precision ones.

3 Coreference Resolution as Graph Labeling

The input of a coreference resolution system includes a document consisting of a set of mentions. Mentions are typically a number of noun phrases that are headed by some pronominal or nominal terminals. An intra-document coreference resolution system partitions such mentions based on their underlying referent entities. Using relations between document's mentions, we can construct an undirected graph in which each mention is represented as a vertex, and each edge corresponds to a coreference relation between two mentions. In other words, assigning mentions to entities can be formulated as a graph labeling problem [4]. As we consider graphs whose vertices represent mentions, here, vertices and mentions are used interchangeably.

It is desired to model the mutual influence between neighboring mentions for simultaneously estimating labels of all mentions in a document. Theoretically, such a model can cover long-range influences between transitively related mentions. Such influences decrease as the distance of two mentions increases. However, for tractability purpose, one should focus on the strongest dependencies between neighbors. Such a model, which is called first-order Markov random Fields [19], cannot be solved in a closed analytic form and is therefore addressed by an iterative technique called relaxation labeling [20].

Relaxation labeling is an iterative optimization process which efficiently solves the problem of assigning a set of labels to a set of variables, while satisfying a set of constraints. Relaxation labeling aims at a label assignment that satisfies as many constraints as possible. In other words, it uses contextual information, which is expressed as a number of constraint functions, for reducing local ambiguities in graph labeling.

One significant feature of relaxation labeling is its ability to deal with any kind of constraints. The algorithm is independent of the complexity of defined constraints (i.e., complexity of modeled application), and it can be improved by using any available constraints. Thus, complex constraints can be used without the need to change the algorithm. Relaxation labeling is applied to various NLP tasks such as POS tagging [21], shallow parsing [22], and supervised coreference resolution [4].

4 System Description

As discussed before, we cast coreference resolution as a graph labeling problem. This was at first inspired by the successful results of [17], which benefits from combining group classification and chain formation methods in a same step. Combination of group classification and chain formation methods in a global method ensures the consistency of solutions [17].

The domain knowledge (i.e., coreference relations) is combined with the model through coefficients of a compatibility matrix. Since the compatibility matrix is a key element of weighted label assignments, the choice of these coefficients is crucial for the success of the algorithm.

These coefficients can be set manually based on the problem specification, or alternatively, they can be learned from a training set. For instance, Sapena et al. [17] uses a decision tree for learning compatibility coefficients. For our method to be unsupervised and therefore independent from any training data, compatibility coefficients should be determined in an unsupervised manner. One possible solution for computing compatibility coefficients is to use Wagstaff and Cardie's approach [23] for deriving incompatibility functions from linguistic features. However, using their method will bring up the concern of setting different heuristic and experimental parameters for weights of compatibility functions [23].

We adopt the idea of sieve architecture presented in [18] for this purpose. The proposed system of Raghunathan et al. [18] is based on the fact that a small number of high precision features is often overwhelmed by a larger number of low precision ones. Thus, Raghunathan et al. [18] proposed a multi-pass system in which higher precision features are deployed at earlier stages of coreference decisions.

We deploy this multi-pass idea in our coreference resolution System. Therefore, our system is a layered system in which each layer is constructed based on different coreference knowledge, and feeds its output forward to the next layer. The layers are organized in a way that highest precision feature is used at the first layer, and successive layers deploy features with decreasing precisions.

The layered architecture is deployed in the graph construction phase; graph is developed incrementally based on different features at each pass, and then the relaxation labeling algorithm is applied to the current partially constructed graph. Therefore, the algorithm will just consider more certain neighborhood relations (i.e., the neighborhood relations that contain higher precision features); unattached vertices will be labeled later at subsequent passes. After determination of weighted label assignments in each partially constructed graph, some of the assignments are determined as being confident enough. These assignments will not change at later passes and therefore will not be affected by weaker features.

4.1 Relaxation Labeling

Suppose that Λ is the set of possible labels for a set of variables V; $V = \{v_1, \dots, v_n\}$ is a set of vertices which, in our modeling, corresponds to the document's mentions, and $R = \{r_{ij}\}$ is a compatibility matrix that defines relations between variables (i.e., adjacency matrix in our problem). Each coefficient r_{ij} corresponds to a constraint regarding to v_i and v_j. A higher value for r_{ij} indicates a higher possibility for v_i and v_j to have the same label.

Relaxation labeling starts by assigning initial labels to all variables. It then iteratively modifies label assignments in a manner that the labeling satisfies as many constraints as possible, where constraints are defined by the compatibility matrix. Information of the compatibility matrix and the current label assignment are used for parallel update of labels. In other words, each variable $v_i \in V$ gets an initial probability vector $\bar{p_i}^0$, which has one element for each possible label of v_i. $p_i^{(t)}(\lambda)$ is an element of $\bar{p_i}^{(t)}$, which corresponds to the probability of assigning label λ to variable v_i at the tth iteration. The whole set of $\bar{p} = \{\bar{p_1}, \dots, \bar{p_n}\}$ is denominated as weighted label assignments.

A support function is defined for each possible label λ of each variable v_i. The compatibility of the current label assignments of neighbors of v_i, and hypothesis "λ is the label of v_i", is measured by this support function. The support function is defined as follows:

$$S_i^{(t)}(\lambda; \bar{p}) = \sum_{j \in neighbors(v_i)} r_{ij} \times p_j^{(t)}(\lambda) \tag{1}$$

Clearly, the higher value of the support function indicates that it is more probable to label v_i with λ. The support function is then used for updating label assignments:

$$p_i^{(t+1)}(\lambda) = \frac{p_i^{(t)}(\lambda) \times (-m + s_i^{(t)}(\lambda, \bar{p}^{(t)}))}{\sum_{\sigma \in \Lambda} p_i^{(t)}(\sigma) \times (-m + s_i^{(t)}(\sigma, \bar{p}^{(t)}))}, \tag{2}$$

where $m = min(\bar{s}_i^{(t)})$.

We use a negative value for r_{ij} when v_i and v_j are incompatible in terms of coreference features (e.g., their gender features are incompatible). Therefore, s_i can have negative values. m is added for negative support values, and the denominator is for normalizing the result, so that $p_i^{(t+1)}(\lambda)$ will remain a probability.

The process of calculating $p_i^{(t+1)}(\lambda)$ continues until the algorithm converges to stable values for p, or it reaches a predefined maximum number of iterations. Relaxation labeling complexity is linear in proportion to the number of variables (i.e., number of mentions in a document).

4.2 Hierarchical Graph Development

In each pass of our hierarchical graph development algorithm, the system processes all mentions of a document. Supposing the algorithm is in pass j, containing feature set $F_j = \{f_1^j, \ldots, f_m^j\}$, where m is the number of features enclosed in pass j. For each mention m_i, the adjacency graph development process will be performed as follows:

Every mention m_k located before m_i is considered as a candidate for graph development. If both m_k and m_i share one of features $\{f_1^j, \ldots, f_m^j\}$, the vertices v_k and v_i corresponding to m_k and m_i, will be attached by a new edge (only if they were unattached before). There is two possible values for edge weights (i.e., r_{ij}): +1 and −1. A weight of +1 represents a preference, and a weight of −1 represents a restriction. The partially constructed graph of each pass contains only the vertices that have at least one edge to some other vertices.

The features that are used at each pass of the system and their corresponding weights are listed in Table 1. It is notable that the first 6 passes mostly consider non-pronoun mentions and the last pass is only for pronouns. A detailed description of the used features can be found in [6,13,14].

4.3 Initialization and Post-processing of Each Pass

We use the same approach as [17] for initializing weighted label assignments. The first non-pronoun mention has no previous mention to be referred to, and it will be considered as the beginning of a new entity. The label assignment of this mention is marked as a first confident assignment in our model. The final label assignments of each pass are considered as the initial label assignments of the next pass. Indeed, if a vertex has a positive neighbor (i.e., a neighbor with a positive weight) with a confident label assignment, it's weighted label assignment will also be marked as a confident assignment at the end of the current pass, and therefore, it would not be changed at later passes.

5 Experiments

5.1 Data

In this work, the following data sets are used for the evaluation purpose.

- ACE2004-NWIRE: the newswire part of ACE 2004. It consists of 128 documents and 11413 mentions.

Table 1. The feature sets of each pass of the system and their corresponding weights.

Pass	Weight	Feature
1	+1	Exact match
2	+1	Appositive Role appositive Alias Demonym Relative Pronoun
	−1	Gender mismatch Number mismatch Entity type mismatch
3	+1	Head match + same non-stop words + compatible modifiers
4	+1	Head match + same non-stop words Head match + compatible modifiers
5	+1	Head match
6	+1	Substring
7	+1	Gender match Number match Entity type match Animacy match Both speak

- ACE2004-ROTH-DEV: A development set of ACE 2004, which is first utilized in [13]. It consists of 68 documents and 4536 mentions.
- ACE2004-CULOTTA-TEST: A test split of ACE 2004, which is first utilized in [24]. It consists of 107 documents and 5469 mentions.

5.2 Results

The experimental results of our approach are presented in Table 2. Since most of existing evaluations on ACE data sets are based on gold mentions, we also use gold mention boundaries for our experiments. In order to measure the impact of hierarchical graph development, we also present results of a single pass flat variant of our system. This variant constructs the adjacency graph in a single step and uses all features of the multi-pass system in just one step. In this version, edge weights are computed as follow:

$$w_{ij} = \min(1, \sum_{f \in F} \delta_k f_k(m_i, m_j)), \tag{3}$$

where δ_k is a fix weight considered for each feature f_k. Since the first three passes of the multi-pass system contain higher precision features, δ_k is set to $+1$ for such features (δ_k is set to -1 for the features of pass 2 that add a negative edge). δ_k is set to 0.25 for other features. The preprocessing pipelines of both variants of the proposed system are the same as that of [13].

As it is shown, the results of the multi-pass system are considerably higher than that of the single pass variant. However, we still need some further work to reach the performance of more successful unsupervised coreference systems (e.g., [6,14]).

Table 2. Experimental results on ACE 2004 data sets.

	MUC			B^3		
System	R	P	F1	R	P	F1
ACE2004-NWIRE						
Single-pass	53.0	72.1	61.1	41.0	70.2	51.8
Multi-pass	69.1	67.2	68.1	72.6	69.4	71.0
ACE2004-ROTH-DEV						
Single-pass	52.8	76.8	62.6	39.0	78.7	52.2
Multi-pass	69.1	67.1	68.1	73.2	71.6	72.4
ACE2004-CULOTTA-TEST						
Single-pass	52.5	73.5	61.3	41.5	72.6	52.8
Multi-pass	63.5	60.7	62.1	70.4	69.0	69.7

Table 3. Pairwise errors made by our system on the ACE2004-NWIRE data set. Each cell indicates error rate made on the specified configuration.

		Antecedent type		
		Proper	Nominal	Total
Anaphora type	Proper	241/1140	53/171	249/1311
	Nominal	56/257	493/921	549/1178
	Pronoun	286/566	285/451	553/1017

5.3 Error Analysis

Table 3 shows the number of pairwise errors made by the proposed multi-pass system on ACE2004-NWIRE. As it is shown, most errors are made on the nominal anaphora with nominal antecedents. There are several reasons causing a rather high rate for this type of errors. Typically, such errors are caused by wrong head match assumptions, and missing semantic and syntactic compatibility information of the two nominal mentions. Lee et al. [6] uses the first mention of each cluster at most passes; the first mention of each cluster is often more representative than other mentions of that cluster. This can reduce errors such as those made by wrong head match assumptions. Using additional linguistic knowledge such as parse trees, binding theory, salience hierarchy, richer semantic knowledge, and cluster-wise feature sets can further decrease coreference decision errors.

6 Post Conference Section

In this section, we propose an alternative approach for the hierarchical graph development. The main purpose of the hierarchical graph development is the appropriate selection of neighbors in Eq. 1. Graph is at first constructed based on more precise features. Therefore, at each pass of the algorithm, the *neighbors* function just returns more important neighbors of each mention, and the label assignments will be determined based on the label assignments of those neighbors.

As an alternative way for this hierarchical graph development method, we can use a different *neighbors* function in Eq. 1, which provides the same benefit for the labeling algorithm.

Suppose that $F = \{F_1, \ldots, F_m\}$ is a set of feature sets in which each F_i contains one or more binary coreference features, and all F_is are ordered based on their precisions. This ordering can be done manually based on some linguistic knowledge [6], or it can be done based on an automatic feature ordering method. Given F, we can define the *neighbors* function as follows:

$$neighbors(v_i) = \{v_j | \exists_{f_l \in F_k} f_l(v_i, v_j) = true\} \tag{4}$$

where k is the first index for which there exist a $f_l \in F_k$ with a *true* value for v_i and at least one other vertex. In this way, less precise features will just be considered in the absence of more precise features.

Using this new *neighbors* function, all mention can be labeled simultaneously in just one pass, while the labeling algorithm considers more precise neighbors of each mention for determining label assignments. In this way, the resolution process will be less time consuming while it still benefits from the hierarchical use of coreference features.

7 Conclusions

In this paper, we examine and evaluate the applicability of relaxation labeling in unsupervised coreference resolution, which has been inspired by the earlier work of [17], where relaxation labeling technique is used for supervised coreference resolution.

In comparison to [17], our model is totally unsupervised (i.e., it does not need any labeled data for determining edge weights), and it uses a hierarchical graph development algorithm. This hierarchical graph construction method prevents the small numbers of high precision features to be overwhelmed by a larger number low precision ones. In the hierarchical graph construction, instead of considering the whole set of neighbors, the labeling algorithm just considers the most reliable set of neighbors for labeling a mention at each pass.

We also present a new *neighbors* function as an alternative way to the hierarchical graph development method that provides the same benefits, while being less time consuming.

Although the presented system underperforms the state-of-the-art systems, it shows promising results and can be further improved in several ways. A natural way to extend the model is to incorporate more linguistic knowledge sources, such as those used in [6,14].

References

1. McCallum, A., Wellner, B.: Toward conditional models of identity uncertainty with application to proper noun coreference. In: Proceedings of the IJCAI-03 Workshop on Information Integration on the Web, Acapulco, Mexico, 9–10 August 2003, pp. 79–86 (2003)
2. Lang, J., Qin, B., Liu, T., Li, S.: Unsupervised coreference resolution with hypergraph partitioning. Comput. Inf. Sci. **2**, 55–63 (2009)
3. Cai, J., Mújdricza-Maydt, É., Strube, M.: Unrestricted coreference resolution via global hypergraph partitioning. In: Proceedings of the Shared Task of the 15th Conference on Computational Natural Language Learning, Portland, Oregon, 23–24 June 2011, pp. 56–60 (2011)
4. Sapena, E., Padró, L., Turmo, J.: RelaxCor participation in CoNLL shared task on coreference resolution. In: Proceedings of the Shared Task of the 15th Conference on Computational Natural Language Learning, Portland, Oregon, 23–24 June 2011, pp. 35–39 (2011)

5. Martschat, S., Cai, J., Broscheit, S., Mújdricza-Maydt, É., Strube, M.: A multigraph model for coreference resolution. In: Proceedings of the Shared Task of the 16th Conference on Computational Natural Language Learning, Jeju Island, Korea, 12–14 July 2012, pp. 100–106 (2012)
6. Lee, H., Chang, A., Peirsman, Y., Chambers, N., Surdeanu, M., Jurafsky, D.: Deterministic coreference resolution based on entity-centric, precision-ranked rules. Computat. Linguist. **39**, 885–916 (2013)
7. Ng, V.: Supervised noun phrase coreference research: the first fifteen years. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden, 11–16 July 2010, pp. 1396–1411 (2010)
8. Soon, W.M., Ng, H.T., Lim, D.C.Y.: A machine learning approach to coreference resolution of noun phrases. Comput. Linguist. **27**(4), 521–544 (2001)
9. Yang, X., Zhou, G., Su, J., Tan, C.L.: Coreference resolution using competition learning approach. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, Sapporo, Japan, 7–12 July 2003, pp. 176–183 (2003)
10. Ji, H., Westbrook, D., Grishman, R.: Using semantic relations to refine coreference decisions. In: Proceedings of the Human Language Technology Conference and the 2005 Conference on Empirical Methods in Natural Language Processing, Vancouver, B.C., Canada, 6–8 October 2005 pp. 17–24 (2005)
11. Ponzetto, S.P., Strube, M.: Semantic role labeling for coreference resolution. In: Companion Volume to the Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics, Trento, Italy, 3–7 April 2006, pp. 143–146 (2006)
12. Denis, P., Baldridge, J.: Specialized models and ranking for coreference resolution. In: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, Waikiki, Honolulu, Hawaii, 25–27 October 2008, pp. 660–669 (2008)
13. Bengtson, E., Roth, D.: Understanding the value of features for coreference resolution. In: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, Waikiki, Honolulu, Hawaii, 25–27 October 2008, pp. 294–303 (2008)
14. Haghighi, A., Klein, D.: Simple coreference resolution with rich syntactic and semantic features. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Singapore, 6–7 August 2009, pp. 1152–1161 (2009)
15. Haghighi, A., Klein, D.: Coreference resolution in a modular, entity centered model. In: Proceedings of Human Language Technologies 2010: The Conference of the North American Chapter of the Association for Computational Linguistics, Los Angeles, Cal., 2–4 June 2010, pp. 385–393 (2010)
16. Martschat, S.: Multigraph clustering for unsupervised coreference resolution. In: Proceedings of the Student Research Workshop at the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, 5–7 August 2013 (2013, to appear)
17. Sapena, E., Padró, L., Turmo, J.: A global relaxation labeling approach to coreference resolution. In: Proceedings of Coling 2010: Poster Volume, Beijing, China, 23–27 August 2010, pp. 1086–1094 (2010)
18. Raghunathan, K., Lee, H., Rangarajan, S., Chambers, N., Surdeanu, M., Jurafsky, D., Manning, C.: A multi-pass sieve for coreference resolution. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Cambridge, Massachusetts, 9–11 October 2010, pp. 492–501 (2010)
19. Pelkowitz, L.: A continuous relaxation labeling algorithm for markov random fields. IEEE Trans. Syst. Man Cybern. **20**, 709–715 (1990)

20. Hummel, R., Zucker, S.W.: On the foundations of relaxation labeling processes. IEEE Trans. Pattern Anal. Mach. Intell. **5**, 267–287 (1983)
21. Marquez, L., Padro, L.: A flexible pos tagger using an automatically acquired language model. In: Proceedings of the Eighth Conference on European Chapter of the Association for Computational Linguistics, pp. 238–245 (1997)
22. Voutilainen, A., Padro, L.: Developing a hybrid NP parser. In: Proceedings of the Fifth Conference on Applied Natural Language Processing, pp. 80–87 (1997)
23. Cardie, C., Wagstaff, K.: Noun phrase coreference as clustering. In: Proceedings of the 1999 SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, College Park, Maryland, 21–22 June 1999, pp. 82–89 (1999)
24. Culotta, A., Wick, M., McCallum, A.: First-order probabilistic models for coreference resolution. In: Proceedings of Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics, Rochester, New York, 22–27 April 2007, pp. 81–88 (2007)