Inference Protocols for Coreference Resolution

Kai-Wei Chang Rajhans Samdani
Alla Rozovskaya Nick Rizzolo
Mark Sammons Dan Roth
University of Illinois at Urbana-Champaign
{kchang10|rsamdan2|rozovska|rizzolo|mssammon|danr}@illinois.edu

Abstract

This paper presents *Illinois-Coref*, a system for coreference resolution that participated in the CoNLL-2011 shared task. We investigate two inference methods, *Best-Link* and *All-Link*, along with their corresponding, pairwise and structured, learning protocols. Within these, we provide a flexible architecture for incorporating linguistically-motivated constraints, several of which we developed and integrated. We compare and evaluate the inference approaches and the contribution of constraints, analyze the mistakes of the system, and discuss the challenges of resolving coreference for the OntoNotes-4.0 data set.

1 Introduction

The coreference resolution task is challenging, requiring a human or automated reader to identify denotative phrases ("mentions") and link them to an underlying set of referents. Human readers use syntactic and semantic cues to identify and disambiguate the referring phrases; a successful automated system must replicate this behavior by linking mentions that refer to the same underlying entity.

This paper describes *Illinois-Coref*, a coreference resolution system built on Learning Based Java (Rizzolo and Roth, 2010), that participated in the "closed" track of the CoNLL-2011 shared task (Pradhan et al., 2011). Building on elements of the coreference system described in Bengtson and Roth (2008), we design an end-to-end system (Sec. 2) that identifies candidate mentions and then applies one of two inference protocols, *Best-Link* and *All-Link* (Sec. 2.3), to disambiguate and cluster them. These protocols were designed to easily incorporate domain knowledge in the form of constraints. In Sec. 2.4, we describe the constraints that we develop and incorporate into the system. The different strategies for mention detection and inference, and the integration of constraints are evaluated in Sections 3 and 4.

2 Architecture

Illinois-Coref follows the architecture used in Bengtson and Roth (2008). First, candidate mentions are detected (Sec. 2.1). Next, a pairwise classifier is applied to each pair of mentions, generating a score that indicates their compatibility (Sec. 2.2). Next, at inference stage, a coreference decoder (Sec. 2.3) aggregates these scores into mention clusters. The original system uses the *Best-Link* approach; we also experiment with *All-Link* decoding. This flexible decoder architecture allows linguistic or knowledge-based constraints to be easily added to the system: constraints may force mentions to be coreferent or non-coreferent and can be optionally used in either of the inference protocols. We designed and implemented several such constraints (Sec. 2.4). Finally, since mentions that are in singleton clusters are not annotated in the OntoNotes-4.0 data set, we remove those as a post-processing step.

2.1 Mention Detection

Given a document, a mention detector generates a set of mention candidates that are used by the subsequent components of the system. A robust mention detector is crucial, as detection errors will propagate to the coreference stage. As we show in Sec. 3, the system that uses gold mentions outperforms the system that uses predicted mentions by a large margin, from 15% to 18% absolute difference.

Proceedings of the 15th Conference on Computational Natural Language Learning: Shared Task, pages 40–44,
Portland, Oregon, 23-24 June 2011.

For the ACE 2004 coreference task, a good performance in mention detection is typically achieved by training a classifier e.g., (Bengtson and Roth, 2008). However, this model is not appropriate for the OntoNotes-4.0 data set, in which (in contrast to the ACE 2004 corpus) singleton mentions are not annotated: a specific noun phrase (NP) may correspond to a mention in one document but will not be a mention in another document. Therefore, we designed a high recall ($\sim$ 90%) and low precision ($\sim$ 35%) rule-based mention detection system that includes all phrases recognized as Named Entities (NE's) and all phrases tagged as NPs in the syntactic parse of the text. As a post-processing step, we remove all predicted mentions that remain in singleton clusters after the inference stage.

The best mention detection result on the DEV set[1] is 64.93% in F1 score (after coreference resolution) and is achieved by our best inference protocol, *Best-Link* with constraints.

2.2 Pairwise Mention Scoring

The basic input to our inference algorithm is a pairwise mention score, which indicates the compatibility score of a pair of mentions. For any two mentions u and v, the compatibility score w_{uv} is produced by a pairwise scoring component that uses extracted features $\phi(u, v)$ and linguistic constraints c:

$$w_{uv} = \mathbf{w} \cdot \phi(u, v) + c(u, v) + t, \qquad (1)$$

where $\mathbf{w}$ is a weight vector learned from training data, $c(u, v)$ is a compatibility score given by the constraints, and t is a threshold parameter (to be tuned). We use the same features as Bengtson and Roth (2008), with the knowledge extracted from the OntoNotes-4.0 annotation. The exact use of the scores and the procedure for learning weights $\mathbf{w}$ are specific to the inference algorithm and are described next.

2.3 Inference

In this section, we present our inference techniques for coreference resolution. These clustering techniques take as input a set of pairwise mention scores over a document and aggregate them into globally consistent cliques representing entities. We investigate the traditional *Best-Link* approach and a more intuitively appealing *All-Link* algorithm.

2.3.1 Best-Link

Best-Link is a popular approach to coreference resolution. For each mention, it considers the best mention on its left to connect to (best according the pairwise score w_{uv}) and creates a link between them if the pairwise score is above some threshold. Although its strategy is simple, Bengtson and Roth (2008) show that with a careful design, it can achieve highly competitive performance.

Inference: We give an integer linear programming (ILP) formulation of *Best-Link* inference in order to present both of our inference algorithms within the same framework. Given a pairwise scorer $\mathbf{w}$, we can compute the compatibility scores — w_{uv} from Eq. (1) — for all mention pairs u and v. Let y_{uv} be a binary variable, such that $y_{uv} = 1$ *only if* u and v are in the same cluster. For a document d, *Best-Link* solves the following ILP formulation:

$$\begin{aligned} \arg\max_y \quad & \sum_{u,v} w_{uv} y_{uv} \\ \text{s.t} \quad & \sum_{u<v} y_{uv} \leq 1 \; \forall v, \\ & y_{uw} \in \{0, 1\}. \end{aligned} \qquad (2)$$

Eq. (2) generates a set of connected components and all the mentions in each connected component constitute an entity.

Learning: We follow the strategy in (Bengtson and Roth, 2008, Section 2.2) to learn the pairwise scoring function $\mathbf{w}$. The scoring function is trained on:

- Positive examples: for each mention u, we construct a positive example (u, v), where v is the closest preceding mention in u's equivalence class.
- Negative examples: all mention pairs (u, v), where v is a preceding mention of u and u, v are not in the same class.

As a result of the singleton mentions not being annotated, there is an inconsistency in the sample distributions in the training and inference phases. Therefore, we apply the mention detector to the training set, and train the classifier using the union set of gold and predicted mentions.

[1] In the shared task, the data set is split into three sets: TRAIN, DEV, and TEST.

2.3.2 All-Link

The *All-Link* inference approach scores a clustering of mentions by including all possible pairwise links in the score. It is also known as correlational clustering (Bansal et al., 2002) and has been applied to coreference resolution in the form of supervised clustering (Mccallum and Wellner, 2003; Finley and Joachims, 2005).

Inference: Similar to *Best-Link*, for a document d, *All-Link* inference finds a clustering *A*ll-Link$(d; w)$ by solving the following ILP problem:

$$\begin{aligned} \arg\max_y \quad & \sum_{u,v} w_{uv} y_{uv} \\ \text{s.t} \quad & y_{uw} \geq y_{uv} + y_{vw} - 1 \; \forall u, w, v, \\ & y_{uw} \in \{0, 1\}. \end{aligned} \tag{3}$$

The inequality constraints in Eq. (3) enforce the transitive closure of the clustering. The solution of Eq. (3) is a set of cliques, and the mentions in the same cliques corefer.

Learning: We present a structured perceptron algorithm, which is similar to supervised clustering algorithm (Finley and Joachims, 2005) to learn w. Note that as an approximation, it is certainly possible to use the weight parameter learned by using, say, averaged perceptron over positive and negative links. The pseudocode is presented in Algorithm 1.

Algorithm 1 Structured Perceptron like learning algorithm for All-Link inference

Given: Annotated documents D and initial weight w_{init}
Initialize $w \leftarrow w_{init}$
for Document d in D **do**
  Clustering $y \leftarrow$ *A*ll-Link$(d; w)$
  for all pairs of mentions u and v **do**
    $\mathcal{I}^1(u, v) = [u, v \text{ coreferent in } D]$
    $\mathcal{I}^2(u, v) = [y(u) = y(v)]$
    $w \leftarrow w + \big(\mathcal{I}^1(u, v) - \mathcal{I}^2(u, v)\big)\, \phi(u, v)$
  end for
end for
return w

For the *All-Link* clustering, we drop one of the three transitivity constraints for each triple of mention variables. Similar to Pascal and Baldridge (2009), we observe that this improves accuracy — the reader is referred to Pascal and Baldridge (2009) for more details.

2.4 Constraints

The constraints in our inference algorithm are based on the analysis of mistakes on the DEV set[2]. Since the majority of errors are mistakes in recall, where the system fails to link mentions that refer to the same entity, we define three high precision constraints that improve recall on NPs with definite determiners and mentions whose heads are NE's.

The patterns used by constraints to match mention pairs have some overlap with those used by the pairwise mention scorer, but their formulation as constraints allow us to focus on a subset of mentions to which a certain pattern applies with high precision. For example, the constraints use a rule-based string similarity measure that accounts for the inferred semantic type of the mentions compared. Examples of mention pairs that are correctly linked by the constraints are: *Governor Bush* ⇒ *Bush*; *a crucial swing state , Florida* ⇒ *Florida*; *Sony itself* ⇒ *Sony*; *Farmers* ⇒ *Los Angeles - based Farmers*.

3 Experiments and Results

In this section, we present the performance of the system on the OntoNotes-4.0 data set. A previous experiment using an earlier version of this data can be found in (Pradhan et al., 2007). Table 1 shows the performance for the two inference protocols, with and without constraints. *Best-Link* outperforms *All-Link* for both predicted and gold mentions. Adding constraints improves the performance slightly for *Best-Link* on predicted mentions. In the other configurations, the constraints either do not affect the performance or slightly degrade it.

Table 2 shows the results obtained on TEST, using the best system configurations found on DEV. We report results on predicted mentions with predicted boundaries, predicted mentions with gold boundaries, and when using gold mentions[3].

[2]We provide a more detailed analysis of the errors in Sec. 4.

[3]Note that the *gold boundaries* results are different from the *gold mention* results. Specifying gold mentions requires coreference resolution to exclude singleton mentions. Gold boundaries are provided by the task organizers and also include singleton mentions.

Method	Pred. Mentions w/Pred. Boundaries					Gold Mentions			
	MD	MUC	BCUB	CEAF	AVG	MUC	BCUB	CEAF	AVG
Best-Link	64.70	55.67	69.21	43.78	56.22	80.58	75.68	64.69	**73.65**
Best-Link W/ Const.	64.69	55.8	69.29	43.96	**56.35**	80.56	75.02	64.24	73.27
All-Link	63.30	54.56	68.50	42.15	55.07	77.72	73.65	59.17	70.18
All-Link W/ Const.	63.39	54.56	68.46	42.20	55.07	77.94	73.43	59.47	70.28

Table 1: The performance of the two inference protocols on both gold and predicted mentions. The systems are trained on the TRAIN set and evaluated on the DEV set. We report the F1 scores (%) on mention detection (MD) and coreference metrics (MUC, BCUB, CEAF). The column AVG shows the averaged scores of the three coreference metrics.

Task	MD	MUC	BCUB	CEAF	AVG
Pred. Mentions w/ Pred. Boundaries	64.88	57.15	67.14	41.94	55.96
Pred. Mentions w/ Gold Boundaries	67.92	59.79	68.65	41.42	56.62
Gold Mentions	-	82.55	73.70	65.24	73.83

Table 2: The results of our submitted system on the TEST set. The system uses *Best-Link* decoding with constraints on predicted mentions and *Best-Link* decoding without constraints on gold mentions. The systems are trained on a collection of TRAIN and DEV sets.

4 Discussion

Most of the mistakes made by the system are due to not linking co-referring mentions. The constraints improve slightly the recall on a subset of mentions, and here we show other common errors for the system. For instance, the system fails to link the two mentions, *the Emory University hospital in Atlanta* and *the hospital behind me*, since each of the mentions has a modifier that is not part of the other mention. Another common error is related to pronoun resolution, especially when a pronoun has several antecedents in the immediate context, appropriate in gender, number, and animacy, as in " *E. Robert Wallach* was sentenced by *a U.S. judge* in New York to six years in prison and fined $ 250,000 for *his* racketeering conviction in the Wedtech scandal .": both *E. Robert Wallach* and *a U.S. judge* are appropriate antecedents for the pronoun *his*. Pronoun errors are especially important to address since 35% of the mentions are pronouns.

The system also incorrectly links some mentions, such as: "*The suspect* said *it* took months to repackage..." ("it" cannot refer to a human); "*They* see *them*." (subject and object in the same sentence are linked); and "Many freeway accidents occur simply because people stay inside *the car* and sort out..." (the NP *the car* should not be linked to any other mention, since it does not refer to a specific entity).

5 Conclusions

We have investigated a coreference resolution system that uses a rich set of features and two popular types of clustering algorithm.

While the *All-Link* clustering seems to be capable of taking more information into account for making clustering decisions, as it requires each mention in a cluster to be compatible with all other mentions in that cluster, the *Best-Link* approach still outperforms it. This raises a natural algorithmic question regarding the inherent nature of clustering style most suitable for coreference and regarding possible ways of infusing more knowledge into different coreference clustering styles. Our approach accommodates infusion of knowledge via constraints, and we have demonstrated its utility in an end-to-end coreference system.

Acknowledgments This research is supported by the Defense Advanced Research Projects Agency (DARPA) Machine Reading Program under Air Force Research Laboratory (AFRL) prime contract no. FA8750-09-C-0181 and the Army Research Laboratory (ARL) under agreement W911NF-09-2-0053. Any opinions, findings, and conclusion or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the view of the DARPA, AFRL, ARL or the US government.

References

N. Bansal, A. Blum, and S. Chawla. 2002. Correlation clustering. In *Proceedings of the 43rd Symposium on Foundations of Computer Science*.

E. Bengtson and D. Roth. 2008. Understanding the value of features for coreference resolution. In *EMNLP*, 10.

T. Finley and T. Joachims. 2005. Supervised clustering with support vector machines. In *Proceedings of the International Conference on Machine Learning (ICML)*.

A. Mccallum and B. Wellner. 2003. Toward conditional models of identity uncertainty with application to proper noun coreference. In *The Conference on Advances in Neural Information Processing Systems (NIPS)*.

D. Pascal and J. Baldridge. 2009. Global joint models for coreference resolution and named entity classification. In *Procesamiento del Lenguaje Natural*.

S. Pradhan, L. Ramshaw, R. Weischedel, J. MacBride, and L. Micciulla. 2007. Unrestricted Coreference: Identifying Entities and Events in OntoNotes. In *in Proceedings of the IEEE International Conference on Semantic Computing (ICSC)*, September 17-19.

S. Pradhan, L. Ramshaw, M. Marcus, M. Palmer, R. Weischedel, and N. Xue. 2011. Conll-2011 shared task: Modeling unrestricted coreference in ontonotes. In *Proceedings of the Annual Conference on Computational Natural Language Learning (CoNLL)*.

N. Rizzolo and D. Roth. 2010. Learning Based Java for Rapid Development of NLP Systems. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, Valletta, Malta, 5.