Unrestricted Coreference Resolution via Global Hypergraph Partitioning

Jie Cai and **Éva Mújdricza-Maydt** and **Michael Strube**
Natural Language Processing Group
Heidelberg Institute for Theoretical Studies gGmbH
Heidelberg, Germany
(jie.cai|eva.mujdriczamaydt|michael.strube)@h-its.org

Abstract

We present our end-to-end coreference resolution system, *COPA*, which implements a global decision via hypergraph partitioning. In constrast to almost all previous approaches, we do not rely on separate classification and clustering steps, but perform coreference resolution globally in one step. *COPA* represents each document as a hypergraph and partitions it with a spectral clustering algorithm. Various types of relational features can be easily incorporated in this framwork. *COPA* has participated in the *open* setting of the CoNLL shared task on modeling unrestricted coreference.

1 Introduction

Coreference resolution is the task of grouping mentions of entities into sets so that all mentions in one set refer to the same entity. Most recent approaches to coreference resolution divide this task into two steps: (1) a classification step which determines whether a pair of mentions is coreferent or which outputs a confidence value, and (2) a clustering step which groups mentions into entities based on the output of step 1.

In this paper we present an end-to-end coreference resolution system, *COPA*, which avoids the division into two steps and instead performs a global decision in one step. The system presents a document as a hypergraph, where the vertices denote mentions and the edges denote relational features between mentions. Coreference resolution is then performed globally in one step by partitioning the hypergraph into subhypergraphs so that all mentions in one subhypergraph refer to the same entity (Cai and Strube, 2010). *COPA* assigns edge weights by applying simple descriptive statistics on the tranining data. Since *COPA* does not need to learn an explicit model, we used only 30% of the CoNLL shared task training data. We did this not for efficiency reasons, only for convenience.

While *COPA* has been developed originally to perform coreference resolution on MUC and ACE data (Cai and Strube, 2010), the move to the OntoNotes data (Weischedel et al., 2011) required mainly to update the mention detector and the feature set. Since several off-the-shelf preprocessing components are used, *COPA* participated in the *open* setting of the CoNLL shared task on modeling unrestricted coreference (Pradhan et al., 2011). We did not make extensive use of information beyond information from the closed class setting.

2 Preprocessing

COPA is implemented on top of the *BART*-toolkit (Versley et al., 2008). Documents are transformed into the *MMAX2*-format (Müller and Strube, 2006) which allows for easy visualization and (linguistic) debugging. Each document is stored in several XML-files representing different layers of annotations. These annotations are created by a pipeline of preprocessing components. We use the *Stanford MaxentTagger* (Toutanova et al., 2003) for part-of-speech tagging, and the *Stanford Named Entity Recognizer* (Finkel et al., 2005) for annotating named entities. In order to derive syntactic information, we use the *Charniak/Johnson reranking parser* (Charniak and Johnson, 2005) com-

Proceedings of the 15th Conference on Computational Natural Language Learning: Shared Task, pages 56–60,
Portland, Oregon, 23-24 June 2011.

bined with a constituent-to-dependency conversion Tool (`http://nlp.cs.lth.se/software/treebank_converter`). The preprocessing models are not trained on CoNLL data, so we only participated in the open task.

We have implemented an in-house mention detector, which makes use of the parsing output, the part-of-speech tags, as well as the chunks from the *Yamcha Chunker* (Kudoh and Matsumoto, 2000). For the OntoNotes data, the mention detector annotates the biggest noun phrase spans.

3 COPA: Coreference Partitioner

The *COPA* system consists of modules which derive hyperedges from features and assign edge weights indicating a positive correlation with the coreference relation, and resolution modules which create a hypergraph representation for the testing data and perform partitioning to produce subhypergraphs, each of which represents an entity.

3.1 HyperEdgeCreator

COPA needs training data only for computing the hyperedge weights. Hyperedges represent features. Each hyperedge corresponds to a feature instance modeling a simple relation between two or more mentions. This leads to initially overlapping sets of mentions. Hyperedges are assigned weights which are calculated on the training data as the percentage of the initial edges being in fact coreferent. Due to the simple strategy of assigning edge weights, only a reasonable size of training data is needed.

3.2 Coreference Resolution Modules

Unlike pairwise models, *COPA* processes a document globally in one step, taking care of the preference information among all the mentions simultaneously and clustering them into sets directly. A document is represented as a single hypergraph with multiple edges. The hypergraph resolver partitions the hypergraph into several sub-hypergraphs, each corresponding to one set of coreferent mentions.

3.2.1 HGModelBuilder

A single document is represented in a hypergraph with basic relational features. Each hyperedge in a graph corresponds to an instance of one of those features with the weight assigned by the *HyperEdgeLearner*. Instead of connecting nodes with the target relation as usually done in graph models, *COPA* builds the graph directly out of low dimensional features without assuming a distance metric.

3.2.2 HGResolver

In order to partition the hypergraph we adopt a spectral clustering algorithm (Agarwal et al., 2005). All experimental results are obtained using symmetric Laplacians (L_{sym}) (von Luxburg, 2007).

We apply the recursive variant of spectral clustering, *recursive 2-way partitioning (R2 partitioner)* (Cai and Strube, 2010). This method does not need any information about the number of target sets (the number k of clusters). Instead a stopping criterion $\alpha^\star$ has to be provided which is adjusted on development data.

3.3 Complexity of HGResolver

Since edge weights are assigned using simple descriptive statistics, the time HGResolver needs for building the graph Laplacian matrix is not substantial. For eigensolving, we use an open source library provided by the Colt project[1]which implements a Householder-QL algorithm to solve the eigenvalue decomposition. When applied to the symmetric graph Laplacian, the complexity of the eigensolving is given by $O(n^3)$, where n is the number of mentions in a hypergraph. Since there are only a few hundred mentions per document in our data, this complexity is not an issue. Spectral clustering gets problematic when applied to millions of data points.

4 Features

In our system, features are represented as types of hyperedges. Any realized edge is an instance of the corresponding edge type. All instances derived from the same type have the same weight, but they may get reweighed by the distance feature (see Cai and Strube (2010)). We use three types of features:

negative: prevent edges between mentions;

positive: generate strong edges between mentions;

weak: add edges to an existing graph without introducing new vertices;

[1] `http://acs.lbl.gov/~hoschek/colt/`

In the following subsections we describe the features used in our experiments. Some of the features described in Cai and Strube (2010) had to be changed to cope with the OntoNotes data. We also introduced a few more features (in particular in order to deal with the dialogue section in the data).

4.1 Negative Features

Negative features describe pairwise relations which are most likely not coreferent. While we implemented this information as weak positive features in Cai and Strube (2010), here we apply these features before graph construction as global variables.

When two mentions are connected by a negative relation, no edges will be built between them in the graph. For instance, no edges are allowed between the mention *Hillary Clinton* and the mention *he* due to incompatible gender.

(1) N_Gender, (2) N_Number: Two mentions do not agree in gender or number.

(3) N_SemanticClass: Two mentions do not agree in semantic class (only the *Object*, *Date* and *Person* top categories derived from WordNet (Fellbaum, 1998) are used).

(4) N_Mod: Two mentions have the same syntactic heads, and the anaphor has a pre-modifier which does not occur in the antecedent and does not contradict the antecedent.

(5) N_DSPrn: Two first person pronouns in direct speeches assigned to different speakers.

(6) N_ContraSubjObj: Two mentions are in the subject and object positions of the same verb, and the anaphor is a non-possesive pronoun.

4.2 Positive Features

The majority of well studied coreference features (e.g. Stoyanov et al. (2009)) are actually positive coreference indicators. In our system, the mentions which participate in positive relations are included in the graph representation.

(7) StrMatch_Npron & (8) StrMatch_Pron: After discarding stop words, if the strings of mentions completely match and are not pronouns, they are put into edges of the *StrMatch_Npron* type. When the matched mentions are pronouns, they are put into the *StrMatch_Pron* type edges.

(9) Alias: After discarding stop words, if mentions are aliases of each other (i.e. proper names with partial match, full names and acronyms, etc.).

(10) HeadMatch: If the syntactic heads of mentions match.

(11) Nprn_Prn: If the antecedent is not a pronoun and the anaphor is a pronoun. This feature is restricted to a sentence distance of 2. Though it is not highly weighted, it is crucial for integrating pronouns into the graph.

(12) Speaker12Prn: If the speaker of the second person pronoun is talking to the speaker of the first person pronoun. The mentions contain only first or second person pronouns.

(13) DSPrn: If one of the mentions is the subject of a *speak* verb, and other mentions are first person pronouns within the corresponding direct speech.

(14) ReflexivePrn: If the anaphor is a reflexive pronoun, and the antecedent is subject of the sentence.

(15) PossPrn: If the anaphor is a possesive pronoun, and the antecedent is the subject of the sentence or the subclause.

(16) GPEIsA: If the antecedent is a Named Entity of GPE entity type (i.e. one of the ACE entity type (NIST, 2004)), and the anaphor is a definite expression of the same type.

(17) OrgIsA: If the antecedent is a Named Entity of Organization entity type, and the anaphor is a definite expression of the same type.

4.3 Weak Features

Weak features are weak coreference indicators. Using them as positive features would introduce too much noise to the graph (i.e. a graph with too many singletons). We apply weak features only to mentions already integrated in the graph, so that weak information provides it with a richer structure.

(18) W_Speak: If mentions occur with a word meaning *to say* in a window size of two words.

(19) W_Subject: If mentions are subjects.

(20) W_Synonym: If mentions are synonymous as indicated by WordNet.

5 Results

We submitted *COPA*'s results to the *open* setting in the CoNLL shared task on modeling unrestricted coreference. We used only 30% of the training data

(randomly selected) and the 20 features described in Section 4.

The stopping criterion $\alpha*$ (see Section 3) is tuned on development data to optimize the final coreference scores. A value of 0.06 is chosen for testing.

COPA's results on development set (which consists of 202 files) and on testing set are displayed in Table 1 and Table 2 respectively. The *Overall* numbers in both tables are the average scores of MUC, $BCUBED$ and $CEAF(E)$.

Metric	R	P	F1
MUC	52.69	57.94	55.19
$BCUBED$	64.26	73.39	68.52
$CEAF(M)$	54.44	54.44	54.44
$CEAF(E)$	45.73	40.92	43.19
$BLANC$	69.78	75.26	72.13
$Overall$			55.63

Table 1: *COPA*'s results on CoNLL development set

Metric	R	P	F1
MUC	56.73	58.90	57.80
$BCUBED$	64.60	71.03	67.66
$CEAF(M)$	53.37	53.37	53.37
$CEAF(E)$	42.71	40.68	41.67
$BLANC$	69.77	73.96	71.62
$Overall$			55.71

Table 2: *COPA*'s results on CoNLL testing set

6 Mention Detection Errors

As described in Section 2, our mention detection is based on automatically extracted information, such as syntactic parses and basic noun phrase chunks. Since there is no *minimum span* information provided in the OntoNotes data (in constrast to the previous standard corpus, ACE), exact mention boundary detection is required. A lot of the spurious mentions in our system are generated due to mismatches of ending or starting punctuations, and the OntoNotes annotation is also not consistent in this regard. Our current mention detector does not extract verb phrases. Therefore it misses all the *Event* mentions in the OntoNotes corpus.

We are planning to include idiomatic expression identification into our mention detector, which will help to avoid detecting a lot of spurious mentions, such as *God* in the phrase *for God's sake*.

7 *COPA* Errors

Besides the fact that the current *COPA* is not resolving any *event coreferences*, our in-house mention detector performs weakly in extracting *date* mentions too. As a result, the system outputs several spurious coreference sets, for instance a set containing the *September* from the mention *15th September*.

A large amount of the recall loss in our system is due to the lack of the world knowledge. For example, *COPA* does not resolve the mention *the Europe station* correctly into the entity *Radio Free Europe*, for it has no knowledge that the entity is a station.

Some more difficult coreference phenomena in *OntoNotes* data might require a reasoning mechanism. To be able to connect the mention *the victim* with the mention *the groom's brother*, the event of the brother being killed needs to be intepreted by the system.

We also observed from the experiments that the resolution of the *it* mentions are quite inaccurate. Although our mention detector takes care of discarding pleonastic *it*'s, there are still a lot of them left which introduce wrong coreference sets. Since the *it*'s do not contain enough information by themselves, more features exploring their local syntax are necessary.

8 Conclusions

In this paper we described a coreference resolution system, *COPA*, which implements a global decision in one step via hypergraph partitioning. *COPA*'s hypergraph-based strategy is a general preference model, where the preference for one mention depends on information on all other mentions.

The system implements three types of relational features — negative, positive and weak features, and assigns the edge weights according to the statitics from the training data. Since the weights are robust with respect to the amount of training data we used only 30% of the training data.

Acknowledgements. This work has been funded by the Klaus Tschira Foundation, Heidelberg, Germany. The first author has been supported by a HITS PhD. scholarship.

References

Sameer Agarwal, Jonwoo Lim, Lihi Zelnik-Manor, Pietro Perona, David Kriegman, and Serge Belongie. 2005. Beyond pairwise clustering. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 838–845.

Jie Cai and Michael Strube. 2010. End-to-end coreference resolution via hypergraph partitioning. In *Proceedings of the 23rd International Conference on Computational Linguistics*, Beijing, China, 23–27 August 2010, pages 143–151.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and MaxEent discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, Ann Arbor, Mich., 25–30 June 2005, pages 173–180.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, Mass.

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*, Ann Arbor, Mich., 25–30 June 2005, pages 363–370.

Taku Kudoh and Yuji Matsumoto. 2000. Use of Support Vector Machines for chunk identification. In *Proceedings of the 4th Conference on Computational Natural Language Learning*, Lisbon, Portugal, 13–14 September 2000, pages 142–144.

Christoph Müller and Michael Strube. 2006. Multi-level annotation of linguistic data with MMAX2. In Sabine Braun, Kurt Kohn, and Joybrato Mukherjee, editors, *Corpus Technology and Language Pedagogy: New Resources, New Tools, New Methods*, pages 197–214. Peter Lang: Frankfurt a.M., Germany.

NIST. 2004. The ACE evaluation plan: Evaluation of the recognition of ACE entities, ACE relations and ACE events. http://www.itl.nist.gov/iad/mig//tests/ace/2004/doc/ace04-evalplan-v7.pdf.

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 Shared Task of 15th Conference on Computational Natural Language Learning*, Portland, Oreg., 23–24 June 2011.

Veselin Stoyanov, Nathan Gilbert, Claire Cardie, and Ellen Riloff. 2009. Conundrums in noun phrase coreference resolution: Making sense of the state-of-the-art. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing*, Singapore, 2–7 August 2009, pages 656–664.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, Edmonton, Alberta, Canada, 27 May –1 June 2003, pages 252–259.

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 *Companion Volume to the Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, Columbus, Ohio, 15–20 June 2008, pages 9–12.

Ulrike von Luxburg. 2007. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. 2011. OntoNotes release 4.0. LDC2011T03, Philadelphia, Penn.: Linguistic Data Consortium.