# End-to-End Coreference Resolution via Hypergraph Partitioning

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

## Abstract

We describe a novel approach to coreference resolution 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. Our hypergraph-based global model implemented within an end-to-end coreference resolution system outperforms two strong baselines (Soon et al., 2001; Bengtson & Roth, 2008) using system mentions only.

## 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.

The classification steps of most approaches vary in the choice of the classifier (e.g. decision tree classifiers (Soon et al., 2001), maximum entropy classification (Luo et al., 2004), SVM classifiers (Rahman & Ng, 2009)) and the number of features used (Soon et al. (2001) employ a set of twelve simple but effective features while e.g., Ng & Cardie (2002) and Bengtson & Roth (2008) devise much richer feature sets).

The clustering step exhibits much more variation: Local variants utilize a closest-first decision (Soon et al., 2001), where a mention is resolved to its closest possible antecedent, or a best-first decision (Ng & Cardie, 2002), where a mention is resolved to its most confident antecedent (based on the confidence value returned by step 1). Global variants attempt to consider all possible clustering possibilites by creating and searching a *Bell tree* (Luo et al., 2004), by learning the optimal search strategy itself (Daumé III & Marcu, 2005), by building a graph representation and applying graph clustering techniques (Nicolae & Nicolae, 2006), or by employing integer linear programming (Klenner, 2007; Denis & Baldridge, 2009). Since these methods base their global clustering step on a local pairwise model, some global information which could have guided step 2 is already lost. The twin-candidate model (Yang et al., 2008) replaces the pairwise model by learning preferences between two antecedent candidates in step 1 and applies tournament schemes instead of the clustering in step 2.

There is little work which deviates from this two-step scheme. Culotta et al. (2007) introduce a first-order probabilistic model which implements features over sets of mentions and thus operates directly on entities.

In this paper we describe a novel approach to coreference resolution which avoids the division into two steps and instead performs a global decision in one step. We represent a document as a hypergraph, where the vertices denote mentions and the edges denote relational features between mentions. Coreference resolution is performed globally in one step by partitioning the hypergraph into subhypergraphs so that all mentions in one subhypergraph refer to the same entity. Our model out-

*Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 143–151,
Beijing, August 2010

performs two strong baselines, Soon et al. (2001) and Bengtson & Roth (2008).

Soon et al. (2001) developed an end-to-end coreference resolution system for the MUC data, i.e., a system which processes raw documents as input and produces annotated ones as output. However, with the advent of the ACE data, many systems either evaluated only true mentions, i.e. mentions which are included in the annotation, the so-called key, or even received true information for mention boundaries, heads of mentions and mention type (Culotta et al., 2007, inter alia). While these papers report impressive results it has been concluded that this experimental setup simplifies the task and leads to an unrealistic surrogate for the coreference resolution problem (Stoyanov et al., 2009, p.657, p660). We argue that the field should move towards a realistic setting using system mentions, i.e. automatically determined mention boundaries and types. In this paper we report results using our end-to-end coreference resolution system, COPA, without relying on unrealistic assumptions.

## 2 Related Work

Soon et al. (2001) transform the coreference resolution problem straightforwardly into a pairwise classification task making it accessible to standard machine learning classifiers. They use a set of twelve powerful features. Their system is based solely on information of the mention pair anaphor and antecedent. It does not take any information of other mentions into account. However, it turned out that it is difficult to improve upon their results just by applying a more sophisticated learning method and without improving the features. We use a reimplementation of their system as first baseline. Bengtson & Roth (2008) push this approach to the limit by devising a much more informative feature set. They report the best results to date on the ACE 2004 data using true mentions. We use their system combined with our preprocessing components as second baseline.

Luo et al. (2004) perform the clustering step within a Bell tree representation. Hence their system theoretically has access to all possible outcomes making it a potentially global system. However, the classification step is still based on a pairwise model. Also since the search space in the Bell tree is too large they have to apply search heuristics. Hence, their approach loses much of the power of a truly global approach.

Culotta et al. (2007) introduce a first-order probabilistic model which implements features over sets of mentions. They use four features for their first-order model. The first is an enumeration over *pairs* of noun phrases. The second is the output of a *pairwise* model. The third is the cluster size. The fourth counts mention type, number and gender in each cluster. Still, their model is based mostly on information about pairs of mentions. They assume true mentions as input. It is not clear whether the improvement in results translates to system mentions.

Nicolae & Nicolae (2006) describe a graph-based approach which superficially resembles our approach. However, they still implement a two step coreference resolution approach and apply the global graph-based model only to step 2. They report considerable improvements over state-of-the-art systems including Luo et al. (2004). However, since they not only change the clustering strategy but also the features for step 1, it is not clear whether the improvements are due to the graph-based clustering technique. We, instead, describe a graph-based approach which performs classification and clustering in one step. We compare our approach with two competitive systems using the same feature sets.

## 3 COPA: Coreference Partitioner

The COPA system consists of learning modules which learn hyperedge weights from the training data, and resolution modules which create a hypergraph representation for the testing data and perform partitioning to produce subhypergraphs, each of which represents an entity. An example analysis of a short document involving the two entities, BARACK OBAMA and NICOLAS SARKOZY illustrates how COPA works.

> [US President Barack Obama] came to Toronto today.
> [Obama] discussed the financial crisis with [President Sarkozy].
> [He] talked to him [him] about the recent downturn of the European markets.
> [Barack Obama] will leave Toronto tomorrow.

A hypergraph (Figure (1a)) is built for this document based on three features. Two hyperedges denote the feature *partial string match*, {*US President Barack Obama, Barack Obama, Obama*} and {*US President Barack Obama, President Sarkozy*}. One hyperedge denotes the feature *pronoun match*, {*he, him*}. Two hyperedges denote the feature *all speak*, {*Obama, he*} and {*President Sarkozy, him*}.

On this initial representation, a spectral clustering technique is applied to find two partitions which have the strongest within-cluster connections and the weakest between-clusters relations. The cut found is called *Normalized Cut*, which avoids trivial partitions frequently output by the min-cut algorithm. The two output subhypergraphs (Figure (1b)) correspond to two resolved entities shown on both sides of the bold dashed line. In real cases, recursive cutting is applied to all the subhypergraphs resulting from previous steps, until a stopping criterion is reached.

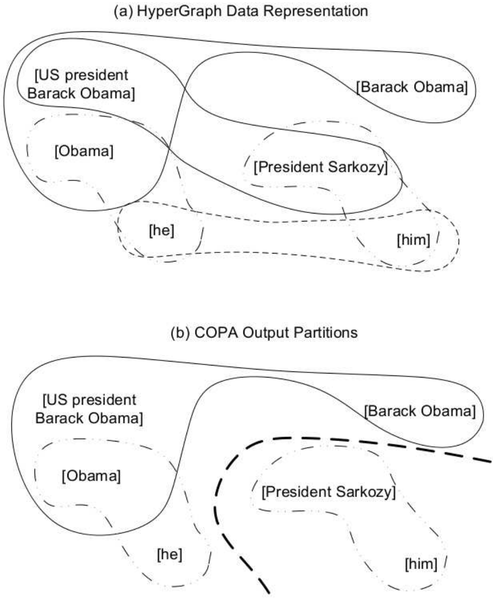
(a) HyperGraph Data Representation
[US president Barack Obama]
[Barack Obama]
[Obama]
[President Sarkozy]
[he]
[him]
(b) COPA Output Partitions
[US president Barack Obama]
[Barack Obama]
[Obama]
[President Sarkozy]
[he]
[him]

Figure 1: Hypergraph-based representation

## 3.1 HyperEdgeLearner

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 based on the training data as the percentage of the initial edges (as illustrated in Figure (1a)) being in fact coreferent. The weights for some of Soon et al. (2001)'s features learned from the ACE 2004 training data are given in Table 1.

| Edge Name | Weight |
|---|---|
| Alias | 0.777 |
| StrMatch_Pron | 0.702 |
| Appositive | 0.568 |
| StrMatch_Npron | 0.657 |
| ContinuousDistAgree | 0.403 |

Table 1: Hyperedge weights for ACE 2004 data

## 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 at the same time and clustering them into sets directly. A raw document is represented as a single hypergraph with multiple edges. The hypergraph resolver partitions the simple hypergraph into several subhypergraphs, each corresponding to one set of coreferent mentions (see e.g. Figure (1b) which contains two subhypergraphs).

### 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 a set of low dimensional features without any assumptions for a distance metric.

### 3.2.2 HGResolver

In order to partition the hypergraph we adopt a spectral clustering algorithm. Spectral clustering techniques use information obtained from the eigenvalues and eigenvectors of the graph Laplacian to cluster the vertices. They are simple to implement and reasonably fast and have been shown to frequently outperform traditional clustering algorithms such as k-means. These techniques have

**Algorithm 1** R2 partitioner

```
Note: { $L = I - D_v{}^{-\frac{1}{2}} HW D_e{}^{-1} H^T D_v{}^{-\frac{1}{2}}$ }
Note: { $Ncut(S) := vol\partial S(\frac{1}{volS} + \frac{1}{volS^c})$}
input: target hypergraph $HG$, predefined $\alpha^\star$
Given a $HG$, construct its $D_v$, $H$, $W$ and $D_e$
Compute $L$ for $HG$
Solve the $L$ for the second smallest eigenvector $V_2$
for each splitting point in $V_2$ do
    calculate $Ncut_i$
end for
Choose the splitting point with $\underset{i}{min}(Ncut_i)$
Generate two sub$HG$s
if $\underset{i}{min}(Ncut_i) < \alpha^*$ then
    for each sub$HG$ do
        Bi-partition the sub$HG$ with the R2 partitioner
    end for
else
    Output the current sub$HG$
end if
output: partitioned $HG$
```

many applications, e.g. image segmentation (Shi & Malik, 2000).

We adopt two variants of spectral clustering, *recursive 2-way partitioning (R2 partitioner)* and *flat-K partitioning*. Since flat-K partitioning did not perform as well we focus here on recursive 2-way partitioning. In contrast to flat-K partitioning, this method does not need any information about the number of target sets. Instead a stopping criterion $\alpha^\star$ has to be provided. $\alpha^\star$ is adjusted on development data (see Algorithm 1).

In order to apply spectral clustering to hypergraphs we follow Agarwal et al. (2005). All experimental results are obtained using symmetric Laplacians ($L_{sym}$) (von Luxburg, 2007).

Given a hypergraph *HG*, a set of matrices is generated. $D_v$ and $D_e$ denote the diagonal matrices containing the vertex and hyperedge degrees respectively. $|V| \times |E|$ matrix $H$ represents the *HG* with the entries $h(v, e) = 1$ if $v \in e$ and $0$ otherwise. $H^T$ is the transpose of $H$. $W$ is the diagonal matrix with the edge weights. $S$ is one of the subhypergraphs generated from a cut in the *HG*, where $Ncut(S)$ is the cut's value.

Using Normalized Cut does not generate singleton clusters, hence a heuristic singleton detection strategy is used in COPA. We apply a threshold $\beta$ to each node in the graph. If a node's degree is below the threshold, the node will be removed.

### 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 insubstantial. 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

The *HGModelBuilder* allows hyperedges with a degree higher than two to grow throughout the building process. This type of edge is *mergeable*. Edges with a degree of two describe pairwise relations. Thus these edges are *non-mergeable*. This way any kind of relational features can be incorporated into the hypergraph model.

Features are represented as types of hyperedges (in Figure (1b) the two hyperedges marked by "–··" are of the same type). 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 reweighted by the distance feature (Section 4.4).

In the following Subsections we describe the features used in our experiments. We use the entire set for obtaining the final results. We restrict ourselves to Soon et al. (2001)'s features when we compare our system with theirs in order to assess the impact of our model regardless of features (we use features 1., 2., 3., 6., 7., 11., 13.).

### 4.1 Hyperedges With a Degree $> 2$

High degree edges are the particular property of the hypergraph which allows to include all types of relational features into our model. The edges are built through pairwise relations and, if consistent, get incrementally merged into larger edges.

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

High degree edges are not sensitive to positional information from the documents.

**(1) StrMatch_Npron & (2) 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.

**(3) Alias:** After discarding stop words, if mentions are aliases of each other (i.e. proper names with partial match, full names and acronyms of organizations, etc.), they are put into edges of the *Alias* type.

**(4) Synonym:** If, according to WordNet, mentions are synonymous, they are put into an edge of the *Synonym* type.

**(5) AllSpeak:** Mentions which appear within a window of two words of a verb meaning *to say* form an edge of the *AllSpeak* type.

**(6) Agreement:** If mentions agree in *Gender*, *Number* and *Semantic Class* they are put in edges of the *Agreement* type. Because *Gender*, *Number* and *Semantic Class* are strong negative coreference indicators – in contrast to e.g. *StrMatch* – and hence weak positive features, they are combined into the one feature *Agreement*.

### 4.2 Hyperedges With a Degree $= 2$

Features which have been used by pairwise models are easily integrated into the hypergraph model by generating edges with only two vertices. Information sensitive to relative distance is represented by pairwise edges.

**(7) Apposition & (8) RelativePronoun:** If two mentions are in a appositive structure, they are put in an edge of type *Apposition*. If the latter mention is a relative pronoun, the mentions are put in an edge of type *RelativePronoun*.

**(9) HeadModMatch:** If the syntactic heads of two mentions match, and if their modifiers do not contradict each other, the mentions are put in an edge of type *HeadModMatch*.

**(10) SubString:** If a mention is the substring of another one, they are put into an edge of type *SubString*.

### 4.3 MentionType and EntityType

In our model **(11) mention type** can only reasonably be used when it is conjoined with other features, since mention type itself describes an attribute of single mentions. In COPA, it is conjoined with other features to form hyperedges, e.g. the *StrMatch_Pron* edge. We use the same strategy to represent **(12) entity type**.

### 4.4 Distance Weights

Our hypergraph model does not have any obvious means to encode distance information. However, the distance between two mentions plays an important role in coreference resolution, especially for resolving pronouns. We do not encode distance as feature, because this would introduce many two-degree-hyperedges which would be computationally very expensive without much gain in performance. Instead, we use distance to reweight two-degree-hyperedges, which are sensitive to positional information.

We experimented with two types of distance weights: One is **(13) sentence distance** as used in Soon et al. (2001)'s feature set, while the other is **(14) compatible mentions distance** as introduced by Bengtson & Roth (2008).

## 5 Experiments

We compare COPA's performance with two implementations of pairwise models. The first baseline is the BART (Versley et al., 2008) reimplementation of Soon et al. (2001), with few but effective features. Our second baseline is Bengtson & Roth (2008), which exploits a much larger feature set while keeping the machine learning approach simple. Bengtson & Roth (2008) show that their system outperforms much more sophisticated machine learning approaches such as Culotta et al. (2007), who reported the best results on true mentions before Bengtson & Roth (2008). Hence, Bengtson & Roth (2008) seems to be a reasonable competitor for evaluating COPA.

In order to report realistic results, we neither assume true mentions as input nor do we evaluate only on true mentions. Instead, we use an inhouse mention tagger for automatically extracting mentions.

### 5.1 Data

We use the MUC6 data (Chinchor & Sundheim, 2003) with standard training/testing divisions (30/30) as well as the MUC7 data (Chinchor, 2001) (30/20). Since we do not have access to the official ACE testing data (only available to ACE participants), we follow Bengtson & Roth (2008) for dividing the ACE 2004 English training data (Mitchell et al., 2004) into training, development and testing partitions (268/76/107). We randomly split the 252 ACE 2003 training documents (Mitchell et al., 2003) using the same proportions into training, development and testing (151/38/63). The systems were tuned on development and run only once on testing data.

### 5.2 Mention Tagger

We implement a classification-based mention tagger, which tags each NP chunk as ACE mention or not, with neccessary post-processing for embeded mentions. For the ACE 2004 testing data, we cover $75.8\%$ of the heads with $73.5\%$ accuracy.

### 5.3 Evaluation Metrics

We evaluate COPA with three coreference resolution evaluation metrics: the $B^3$-algorithm (Bagga & Baldwin, 1998), the *CEAF*-algorithm (Luo, 2005), and, for the sake of completeness, the *MUC*-score (Vilain et al., 1995).

Since the *MUC*-score does not evaluate singleton entities, it only partially evaluates the performance for ACE data, which includes singleton entities in the keys. The $B^3$-algorithm (Bagga & Baldwin, 1998) addresses this problem of the *MUC*-score by conducting calculations based on mentions instead of coreference relations. However, another problematic issue emerges when system mentions have to be dealt with: $B^3$ assumes the mentions in the key and in the response to be identical, which is unlikely when a mention tagger is used to create system mentions. The *CEAF*-algorithm aligns entities in key and response by means of a similarity metric, which is motivated by $B^3$'s shortcoming of using one entity multiple times (Luo, 2005). However, although *CEAF* theoretically does not require to have the same number of mentions in key and response, the algorithm still cannot be directly applied to end-to-end coreference resolution systems, because the similarity metric is influenced by the number of mentions in key and response.

Hence, both the $B^3$- and *CEAF*-algorithms have to be extended to deal with system mentions which are not in the key and true mentions not extracted by the system, so called *twinless mentions* (Stoyanov et al., 2009). Two variants of the $B^3$-algorithm are proposed by Stoyanov et al. (2009), $B^3_{all}$ and $B^3_0$. $B^3_{all}$ tries to assign intuitive precision and recall to the twinless system mentions and twinless key mentions, while keeping the size of the system mention set and the key mention set unchanged (which are different from each other). For twinless mentions, $B^3_{all}$ discards twinless key mentions for precision and twinless system mentions for recall. Discarding parts of the key mentions, however, makes the fair comparison of precision values difficult. $B^3_0$ produces counter-intuitive precision by discarding all twinless system mentions. Although it penalizes the recall of all twinless key mentions, so that the F-scores are balanced, it is still too lenient (for further analyses see Cai & Strube (2010)).

We devise two variants of the $B^3$- and *CEAF*-algorithms, namely $B^3_{sys}$ and *CEAF*$_{sys}$. For computing precision, the algorithms put all twinless true mentions into the response even if they were not extracted. All twinless system mentions which were deemed not coreferent are discarded. Only twinless system mentions which were mistakenly resolved are put into the key. Hence, the system is penalized for resolving mentions not found in the key. For recall the algorithms only consider mentions from the original key by discarding all the twinless system mentions and putting twinless true mentions into the response as singletons (algorithm details, simulations and comparison of different systems and metrics are provided in Cai & Strube (2010)). For *CEAF*$_{sys}$, $\phi_3$ (Luo, 2005) is used. $B^3_{sys}$ and *CEAF*$_{sys}$ report results for end-to-end coreference resolution systems adequately.

### 5.4 Baselines

We compare COPA's performance with two baselines: *SOON* – the BART (Versley et al., 2008) reimplementation of Soon et al. (2001) – and

| | | *SOON* | | | COPA with R2 partitioner | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | R | P | F | R | P | F | $\alpha^\star$ | $\beta$ |
| *MUC* | MUC6 | 59.4 | 67.9 | 63.4 | 62.8 | 66.4 | **64.5** | 0.08 | 0.03 |
| | MUC7 | 52.3 | 67.1 | 58.8 | 55.2 | 66.1 | **60.1** | 0.05 | 0.01 |
| | ACE 2003 | 56.7 | 75.8 | 64.9 | 60.8 | 75.1 | **67.2** | 0.07 | 0.03 |
| | ACE 2004 | 50.4 | 67.4 | 57.7 | 54.1 | 67.3 | **60.0** | 0.05 | 0.04 |
| $B^3_{sys}$ | MUC6 | 53.1 | 78.9 | 63.5 | 56.4 | 76.3 | **64.1** | 0.08 | 0.03 |
| | MUC7 | 49.8 | 80.0 | 61.4 | 53.3 | 76.1 | **62.7** | 0.05 | 0.01 |
| | ACE 2003 | 66.9 | 87.7 | 75.9 | 71.5 | 83.3 | **77.0** | 0.07 | 0.03 |
| | ACE 2004 | 64.7 | 85.7 | 73.8 | 67.3 | 83.4 | **74.5** | 0.07 | 0.03 |
| *CEAF*$_{sys}$ | MUC6 | 56.9 | 53.0 | 54.9 | 62.2 | 57.5 | **59.8** | 0.08 | 0.03 |
| | MUC7 | 57.3 | 54.3 | 55.7 | 58.3 | 54.2 | 56.2 | 0.06 | 0.01 |
| | ACE 2003 | 71.0 | 68.7 | 69.8 | 71.1 | 68.3 | 69.7 | 0.07 | 0.03 |
| | ACE 2004 | 67.9 | 65.2 | 66.5 | 68.5 | 65.5 | 67.0 | 0.07 | 0.03 |

Table 3: *SOON* vs. COPA R2 (*SOON* features, system mentions, bold indicates significant improvement in F-score over *SOON* according to a paired-t test with $p < 0.05$)

| | *SOON* | | | *B&R* | | |
|---|---|---|---|---|---|---|
| | R | P | F | R | P | F |
| $B^3_{sys}$ | 64.7 | 85.7 | 73.8 | 66.3 | 85.8 | 74.8 |

Table 2: Baselines on ACE 2004

*B&R* – Bengtson & Roth (2008)[2]. All systems share BART's preprocessing components and our in-house ACE mention tagger.

In Table 2 we report the performance of *SOON* and *B&R* on the ACE 2004 testing data using the BART preprocessing components and our in-house ACE mention tagger. For evaluation we use $B^3_{sys}$ only, since Bengtson & Roth (2008)'s system does not allow to easily integrate *CEAF*.

*B&R* considerably outperforms *SOON* (we cannot compute statistical significance, because we do not have access to results for single documents in *B&R*). The difference, however, is not as big as we expected. Bengtson & Roth (2008) reported very good results when using true mentions. For evaluating on system mentions, however, they were using a too lenient variant of $B^3$ (Stoyanov et al., 2009) which discards all twinless mentions. When replacing this with $B^3_{sys}$ the difference between *SOON* and *B&R* shrinks.

### 5.5 Results

In both comparisons, COPA uses the same features as the corresponding baseline system.

#### 5.5.1 COPA vs. *SOON*

In Table 3 we compare the *SOON*-baseline with COPA using the R2 partitioner (parameters $\alpha^\star$ and $\beta$ optimized on development data). Even though COPA and *SOON* use the same features, COPA consistently outperforms *SOON* on all data sets using all evaluation metrics. With the exception of the MUC7, the ACE 2003 and the ACE 2004 data evaluated with *CEAF*$_{sys}$, all of COPA's improvements are statistically significant. When evaluated using *MUC* and $B^3_{sys}$, COPA with the R2 partitioner boosts recall in all datasets while losing in precision. This shows that global hypergraph partitioning models the coreference resolution task more adequately than Soon et al. (2001)'s local model – even when using the very same features.

#### 5.5.2 COPA vs. *B&R*

In Table 4 we compare the *B&R* system (using our preprocessing components and mention tagger), and COPA with the R2 partitioner using *B&R* features. COPA does not use the learned features from *B&R*, as this would have implied to embed a pairwise coreference resolution system in COPA. We report results for ACE 2003 and ACE 2004. The parameters are optimized on the ACE 2004 data. COPA with the R2 partitioner outperforms *B&R* on both datasets (we cannot compute statistical significance, because we do not have access to results for single documents in *B&R*). Bengtson & Roth (2008) developed their system on ACE 2004 data and never exposed it to ACE 2003 data. We suspect that the relatively poor result of *B&R* on ACE 2003 data is caused by overfitting to ACE

[2] `http://l2r.cs.uiuc.edu/~cogcomp/asoftware.php?skey=FLBJCOREF`

|  |  | *B&R* |  |  | COPA with R2 partitioner |  |  |
|---|---|---|---|---|---|---|---|
|  |  | R | P | F | R | P | F |
| $B^3_{sys}$ | ACE 2003 | 56.4 | 97.3 | 71.4 | 70.3 | 86.5 | 77.5 |
|  | ACE 2004 | 66.3 | 85.8 | 74.8 | 68.4 | 84.4 | 75.6 |

Table 4: *B&R* vs. COPA R2 (*B&R* features, system mentions)

2004. Again, COPA gains in recall and loses in precision. This shows that COPA is a highly competetive system as it outperforms Bengtson & Roth (2008)'s system which has been claimed to have the best performance on the ACE 2004 data.

### 5.5.3 Running Time

On a machine with 2 AMD Opteron CPUs and 8 GB RAM, COPA finishes preprocessing, training and partitioning the ACE 2004 dataset in 15 minutes, which is slightly faster than our duplicated *SOON* baseline.

## 6 Discussion and Outlook

Most previous attempts to solve the coreference resolution task globally have been hampered by employing a local pairwise model in the classification step (step 1) while only the clustering step realizes a global approach, e.g. Luo et al. (2004), Nicolae & Nicolae (2006), Klenner (2007), Denis & Baldridge (2009), lesser so Culotta et al. (2007). It has been also observed that improvements in performance on true mentions do not necessarily translate into performance improvements on system mentions (Ng, 2008).

In this paper we describe a coreference resolution system, COPA, which implements a global decision in one step via hypergraph partitioning. COPA looks at the whole graph at once which enables it to outperform two strong baselines (Soon et al., 2001; Bengtson & Roth, 2008). COPA's hypergraph-based strategy can be taken as a general preference model, where the preference for one mention depends on information on all other mentions.

We follow Stoyanov et al. (2009) and argue that evaluating the performance of coreference resolution systems on true mentions is unrealistic. Hence we integrate an ACE mention tagger into our system, tune the system towards the real task, and evaluate only using system mentions. While Ng (2008) could not show that superior models achieved superior results on system mentions, COPA was able to outperform Bengtson & Roth (2008)'s system which has been claimed to achieve the best performance on the ACE 2004 data (using true mentions, Bengtson & Roth (2008) did not report any comparison with other systems using system mentions).

An error analysis revealed that there were some cluster-level inconsistencies in the COPA output. Enforcing this consistency would require a global strategy to propagate constraints, so that constraints can be included in the hypergraph partitioning properly. We are currently exploring constrained clustering, a field which has been very active recently (Basu et al., 2009). Using constrained clustering methods may allow us to integrate negative information as constraints instead of combining several weak positive features to one which is still weak (e.g. our *Agreement* feature). For an application of constrained clustering to the related task of database record linkage, see Bhattacharya & Getoor (2009).

Graph models cannot deal well with positional information, such as distance between mentions or the sequential ordering of mentions in a document. We implemented distance as weights on hyperedges which resulted in decent performance. However, this is limited to pairwise relations and thus does not exploit the power of the high degree relations available in COPA. We expect further improvements, once we manage to include positional information directly.

**Acknowledgements.** This work has been funded by the Klaus Tschira Foundation, Heidelberg, Germany. The first author has been supported by a HITS PhD. scholarship. We would like to thank Byoung-Tak Zhang for bringing hypergraphs to our attention and Èva Mújdricza-Maydt for implementing the mention tagger. Finally we would like to thank our colleagues in the HITS NLP group for providing us with useful comments.

## References

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

Bagga, Amit & Breck Baldwin (1998). Algorithms for scoring coreference chains. In *Proceedings of the 1st International Conference on Language Resources and Evaluation,* Granada, Spain, 28–30 May 1998, pp. 563–566.

Basu, Sugato, Ian Davidson & Kiri L. Wagstaff (Eds.) (2009). *Constrained Clustering: Advances in Algorithms, Theory, and Applications.* Boca Raton, Flo.: CRC Press.

Bengtson, Eric & Dan Roth (2008). 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.

Bhattacharya, Indrajit & Lise Getoor (2009). Collective relational clustering. In S. Basu, I. Davidson & K. Wagstaff (Eds.), *Constrained Clustering: Advances in Algorithms, Theory, and Applications*, pp. 221–244. Boca Raton, Flo.: CRC Press.

Cai, Jie & Michael Strube (2010). Evaluation metrics for end-to-end coreference resolution systems. In *Proceedings of the SIGdial 2010 Conference: The 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue,* Tokyo, Japan, 24–25 September 2010. To appear.

Chinchor, Nancy (2001). *Message Understanding Conference (MUC) 7.* LDC2001T02, Philadelphia, Penn: Linguistic Data Consortium.

Chinchor, Nancy & Beth Sundheim (2003). *Message Understanding Conference (MUC) 6.* LDC2003T13, Philadelphia, Penn: Linguistic Data Consortium.

Culotta, Aron, Michael Wick & Andrew McCallum (2007). 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, N.Y., 22–27 April 2007, pp. 81–88.

Daumé III, Hal & Daniel Marcu (2005). A large-scale exploration of effective global features for a joint entity detection and tracking model. 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. 97–104.

Denis, Pascal & Jason Baldridge (2009). Global joint models for coreference resolution and named entity classification. *Procesamiento del Lenguaje Natural*, 42:87–96.

Klenner, Manfred (2007). Enforcing consistency on coreference sets. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing,* Borovets, Bulgaria, 27–29 September 2007, pp. 323–328.

Luo, Xiaoqiang (2005). On coreference resolution performance metrics. 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. 25–32.

Luo, Xiaoqiang, Abe Ittycheriah, Hongyan Jing, Nanda Kambhatla & Salim Roukos (2004). A mention-synchronous coreference resolution algorithm based on the Bell Tree. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics,* Barcelona, Spain, 21–26 July 2004, pp. 136–143.

Mitchell, Alexis, Stephanie Strassel, Shudong Huang & Ramez Zakhary (2004). *ACE 2004 Multilingual Training Corpus*. LDC2005T09, Philadelphia, Penn.: Linguistic Data Consortium.

Mitchell, Alexis, Stephanie Strassel, Mark Przybocki, JK Davis, George Doddington, Ralph Grishman, Adam Meyers, Ada Brunstain, Lisa Ferro & Beth Sundheim (2003). *TIDES Extraction (ACE) 2003 Multilingual Training Data*. LDC2004T09, Philadelphia, Penn.: Linguistic Data Consortium.

Ng, Vincent (2008). Unsupervised models for coreference resolution. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing,* Waikiki, Honolulu, Hawaii, 25-27 October 2008, pp. 640–649.

Ng, Vincent & Claire Cardie (2002). Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics,* Philadelphia, Penn., 7–12 July 2002, pp. 104–111.

Nicolae, Cristina & Gabriel Nicolae (2006). BestCut: A graph algorithm for coreference resolution. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing,* Sydney, Australia, 22–23 July 2006, pp. 275–283.

Rahman, Altaf & Vincent Ng (2009). Supervised models for coreference resolution. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing,* Singapore, 6-7 August 2009, pp. 968–977.

Shi, Jianbo & Jitendra Malik (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905.

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

Stoyanov, Veselin, Nathan Gilbert, Claire Cardie & 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, pp. 656–664.

Versley, Yannick, Simone Paolo Ponzetto, Massimo Poesio, Vladimir Eidelman, Alan Jern, Jason Smith, Xiaofeng Yang & 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, pp. 9–12.

Vilain, Marc, John Burger, John Aberdeen, Dennis Connolly & Lynette Hirschman (1995). A model-theoretic coreference scoring scheme. In *Proceedings of the 6th Message Understanding Conference (MUC-6),* pp. 45–52. San Mateo, Cal.: Morgan Kaufmann.

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

Yang, Xiaofeng, Jian Su & Chew Lim Tan (2008). A twin-candidate model for learning-based anaphora resolution. *Computational Linguistics*, 34(3):327–356.