Narrative Schema as World Knowledge for Coreference Resolution

Joseph Irwin
Nara Institute of Science and Technology
Nara Prefecture, Japan
joseph-i@is.naist.jp

Mamoru Komachi
Nara Institute of Science and Technology
Nara Prefecture, Japan
komachi@is.naist.jp

Yuji Matsumoto
Nara Institute of Science and Technology
Nara Prefecture, Japan
matsu@is.naist.jp

Abstract

In this paper we describe the system with which we participated in the CoNLL-2011 Shared Task on modelling coreference. Our system is based on a cluster-ranking model proposed by Rahman and Ng (2009), with novel semantic features based on recent research on narrative event schema (Chambers and Jurafsky, 2009). We demonstrate some improvements over the baseline when using schema information, although the effect varied between the metrics used. We also explore the impact of various features on our system’s performance.

1 Introduction

Coreference resolution is a problem for automated document understanding. We say two segments of a natural-language document corefer when they refer to the same real-world entity. The segments of a document which refer to an entity are called mentions. In coreference resolution tasks, mentions are usually restricted to noun phrases.

The goal of the CoNLL-2011 Shared Task (Pradhan et al., 2011) is to model unrestricted coreference using the OntoNotes corpus. The OntoNotes corpus is annotated with several layers of syntactic and semantic information, making it a rich resource for investigating coreference resolution (Pradhan et al., 2007).

We participated in both the “open” and “closed” tracks. The “closed” track requires systems to only use the provided data, while the “open” track allows use of external data. We created a baseline system based on the cluster-ranking model proposed by Rahman and Ng (2009). We then experimented with adding novel semantic features derived from co-referring predicate-argument chains. These narrative schema were developed by Chambers and Jurafsky (2009). They are described in more detail in a later section.

2 Related Work

Supervised machine-learning approaches to coreference resolution have been researched for almost two decades. Recently, the state of the art seems to be moving away from the early mention-pair classification model toward entity-based models. Ng (2010) provides an excellent overview of the history and recent developments within the field.

Both entity-mention and mention-pair models are formulated as binary classification problems; however, ranking may be a more natural approach to coreference resolution (Ng, 2010; Rahman and Ng, 2009). Rahman and Ng (2009) in particular propose the cluster-ranking model which we used in our baseline. In another approach, Daumé and Marcu (2005) apply their Learning as Search Optimization framework to coreference resolution, and show good results.

Feature selection is important for good performance in coreference resolution. Ng (2010) discusses commonly used features, and analyses of the contribution of various features can be found in (Daumé and Marcu, 2005; Rahman and Ng, 2011; Ponzetto and Strube, 2006b). Surprisingly, Rahman and Ng (2011) demonstrated that a system using almost exclusively lexical features could outperform...
systems which used more traditional sets of features.

Although string features have a large effect on performance, it is recognized that the use of semantic information is important for further improvement (Ng, 2010; Ponzetto and Strube, 2006a; Ponzetto and Strube, 2006b; Haghighi and Klein, 2010). The use of predicate-argument structure has been explored by Ponzetto and Strube (2006b; 2006a).

3 Narrative Schema for Coreference

Narrative schema are extracted from large-scale corpora using coreference information to identify predicates whose arguments often corefer. Similarity measures are used to build up schema consisting of one or more event chains – chains of typically-coreferring predicate arguments (Chambers and Jurafsky, 2009). Each chain corresponds to a role in the schema.

A role defines a class of participants in the schema. Conceptually, if a schema is present in a document, then each role in the schema corresponds to an entity in the document. An example schema is shown with some typical participants in Figure 1. In this paper the temporal order of events in the schema is not considered.

![Diagram of a narrative schema with two roles](image)

Figure 1: An example narrative schema with two roles.

Narrative schema are similar to the script concept put forth by Schank and Abelson (1977). Like scripts, narrative schema can capture complex structured information about events described in natural language documents (Schank and Abelson, 1977; Abelson, 1981; Chambers and Jurafsky, 2009).

We hypothesize that narrative schema can be a good source of information for making coreference decisions. One reason they could be useful is that they can directly capture the fact that arguments of certain predicates are relatively more likely to refer to the same entity. In fact, they can capture global information about verbs ranging over the entire document, which we expect may lead to greater accuracy when combined with the incremental clustering algorithm we employ.

Additionally, the information that two predicates often share arguments yields semantic information about the argument words themselves. For example, if the subjects of the verbs *eat* and *drink* often corefer, we may be able to infer that words which occur in the subject position of these verbs share some property (e.g., animacy). This last conjecture is somewhat validated by Ponzetto and Strube (2006b), who reported that including predicate-argument pairs as features improved the performance of a coreference resolver.

4 System Description

4.1 Overview

We built a coreference resolution system based on the cluster-ranking algorithm proposed by Rahman and Ng (2009). During document processing maintains a list of clusters of coreferring mentions which are created iteratively. Our system uses a deterministic mention-detection algorithm that extracts candidate NPs from a document. We process the mentions in order of appearance in the document. For each mention a ranking query is created, with features generated from the clusters created so far. In each query we include a null-cluster instance, to allow joint learning of discourse-new detection, following (Rahman and Ng, 2009).

For training, each mention is assigned to its correct cluster according to the coreference annotation. The resulting queries are used to train a classification-based ranker.

In testing, the ranking model thus learned is used to rank the clusters in each query as it is created; the active mention is assigned to the cluster with the highest rank.

A data-flow diagram for our system is shown in Figure 2.
4.2 Cluster-ranking Model

Our baseline system uses a cluster-ranking model proposed by Rahman and Ng (2009; 2011). In this model, clusters are iteratively constructed after considering each active mention in a document in order. During training, features are created between the active mention and each cluster created so far. A rank is assigned such that the cluster which is coreferent to the active mention has the highest value, and each non-coreferent cluster is assigned the same, lower rank (The exact values are irrelevant to learning a ranking; for the experiments in this paper we used the values 2 and 1). In this way it is possible to learn to preferentially rank correct clustering decisions higher.

For classification, instances are constructed exactly the same way as for training, except that for each active mention, a query must be constructed and ranked by the classifier in order to proceed with the clustering. After the query for each active mention has been ranked, the mention is assigned to the cluster with the highest ranking, and the algorithm proceeds to the next mention.

4.3 Notation

In the following sections, $m_k$ is the active mention currently being considered, $m_j$ is a candidate antecedent mention, and $c_j$ is the cluster to which it belongs. Most of the features used in our system actually apply to a pair of mentions (i.e., $m_k$ and $m_j$) or to a single mention (either $m_k$ or $m_j$). To create a training or test instance using $m_k$ and $c_j$, the features which apply to $m_j$ are converted to cluster-level features by a procedure described in 4.6.

4.4 Joint Anaphoric Mention Detection

We follow Rahman and Ng (2009) in jointly learning to detect anaphoric mentions along with resolving coreference relations. For each active mention $m_k$, an instance for a ‘null’ cluster is also created, with rank 2 if the mention is not coreferent with any preceding mention, or rank 1 if it has an antecedent. This allows the ranker the option of making $m_k$ discourse-new. To create this instance, only the features which involve just $m_k$ are used.

4.5 Features

The features used in our system are shown in Table 1. For the NE features we directly use the types from the OntoNotes annotation.

4.6 Making Cluster-Level Features

Each feature which applies to $m_j$ must be converted to a cluster-level feature. We follow the procedure described in (Rahman and Ng, 2009). This procedure uses binary features whose values correspond to being logically true or false. Multi-valued features are first converted into equivalent sets of binary-valued features. For each binary-valued feature, four corresponding cluster-level features are created, whose values are determined by four logical

---

1The set of types is: PERSON, NORP, FACILITY, ORGANIZATION, GPE, LOCATION, PRODUCT, EVENT, WORK, LAW, LANGUAGE, DATE, TIME, PERCENT, MONEY, QUANTITY, ORDINAL, CARDINAL.
The two features marked with * are treated differently. For each value of NE_TYPE1 and NE_TYPE', a new cluster-level feature is created whose value is the number of times that feature/value appeared in the cluster (i.e., if there were two PERSON NEs in a cluster then the feature NE_TYPE1_PERSON would have the value 2).

### 4.7 SCHEMA_CLUSTERS MATCH

The SCHEMA_CLUSTERS MATCH feature is actually three features, which are calculated over an entire candidate antecedent cluster $c_j$. First a list is created of all of the schema roles which the mentions in $c_j$ participate in, and sorted in decreasing order according to how many mentions in $c_j$ participate in each. Then, the value of the feature SCHEMA_CLUSTERS MATCH is $Y$ if mention $m_k$ also participates in the $n_{th}$ schema role in the list, for $n = 1, 2, 3$. If it does not, or if the corresponding $n_{th}$ role has fewer than two participants in $c_j$, the value of this feature is $N$.

### 4.8 Implementation Details

Our system was implemented in Python, in order to make use of the NLTK library\(^2\). For the ranker we used SVM\(^{\text{rank}}\), an efficient implementation for training ranking SVMs (Joachims, 2006) \(^3\).

\(^2\)http://www.nltk.org/

\(^3\)http://svmlight.joachims.org/
5 Experiments and Results

5.1 CoNLL System Submission

We submitted two results to the CoNLL-2011 Shared Task. In the “closed” track we submitted the results of our baseline system without the schema features, trained on all documents in both the training and development portions of the OntoNotes corpus.

We also submitted a result in the “open” track: a version of our system with the schema features added. Due to issues with the implementation of this second version, however, we were only able to submit results from a model trained on just the WSJ portion of the training dataset. For the schema features, we used a database of narrative schema released by Chambers and Jurafsky (2010) – specifically the list of schemas of size 12. 4

The official system scores for our system are listed in Table 3. We can attribute some of the low performance of our system to features which are too noisy, and to having not enough features compared to the large size of the dataset. It is likely that these two factors adversely impact the ability of the SVM to learn effectively. In fact, the features which we introduced partially to provide more features to learn with, the NE features, had the worst impact on performance according to later analysis. Because of a problem with our implementation, we were unable to get an accurate idea of our system’s performance until after the submission deadline.

4Available at http://cs.stanford.edu/people/nc/schemas/

Table 3: Official system results

<table>
<thead>
<tr>
<th></th>
<th>R</th>
<th>P</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>MUC</td>
<td>12.45%</td>
<td>50.60%</td>
<td>19.98</td>
</tr>
<tr>
<td>CLOSED</td>
<td>B³</td>
<td>35.07%</td>
<td>89.90%</td>
</tr>
<tr>
<td>CEAF</td>
<td>45.84%</td>
<td>17.38%</td>
<td>25.21</td>
</tr>
<tr>
<td>Overall score:</td>
<td>31.88</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MUC</td>
<td>18.56%</td>
<td>51.01%</td>
<td>27.21</td>
</tr>
<tr>
<td>OPEN</td>
<td>B³</td>
<td>38.97%</td>
<td>85.57%</td>
</tr>
<tr>
<td>CEAF</td>
<td>43.33%</td>
<td>19.36%</td>
<td>26.76</td>
</tr>
<tr>
<td>Overall score:</td>
<td>35.84</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Schema features evaluated on the development set. Training used the entire training dataset.

<table>
<thead>
<tr>
<th></th>
<th>R</th>
<th>P</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>MUC</td>
<td>12.77%</td>
<td>57.66%</td>
<td>20.91</td>
</tr>
<tr>
<td>Baseline</td>
<td>B³</td>
<td>35.1%</td>
<td>91.05%</td>
</tr>
<tr>
<td>CEAF</td>
<td>47.80%</td>
<td>17.29%</td>
<td>25.40</td>
</tr>
<tr>
<td>+SCHEMA</td>
<td>B³</td>
<td>35.75%</td>
<td>90.39%</td>
</tr>
<tr>
<td>CEAF</td>
<td>46.62%</td>
<td>17.43%</td>
<td>25.38</td>
</tr>
</tbody>
</table>

5.2 Using Narrative Schema as World Knowledge for Coreference Resolution

We conducted an evaluation of the baseline without schema features against a model with both schema features added. The results are shown in Table 4. The results were mixed, with B³ going up and MUC and CEAF falling slightly. Cross-validation using just the development set showed a more positive picture, however, with both MUC and B³ scores increasing more than 1 point (p = 0.06 and p < 0.01, respectively), and CEAF increasing about 0.5 points as well (although this was not significant at p > 0.1). 5

One problem with the schema features that we had anticipated was that they may have a problem with sparseness. We had originally intended to extract schema using the coreference annotation in OntoNotes, predicting that this would help alleviate the problem; however, due to time constraints we were unable to complete this effort.

5All significance tests were performed with a two-tailed t-test.

5.3 Feature Analysis

We conducted a feature ablation analysis on our baseline system to better understand the contribution of each feature to overall performance. The results are shown in Table 5. We removed features in blocks of related features; -HEAD removes HEAD_MATCH; -DIST removes the DISTANCE feature; -SUBJ is the baseline system without SUBJECT; -PRO is the baseline system without PRO- NOUN2, PROTYPE2, and PRONOUN_MATCH; -DEF_DEM removes DEFINITE, DEMONSTRATIVE, and DEF_DEM NA; and -NE removes the named entity features.
Table 5: Effect of each feature on performance.

<table>
<thead>
<tr>
<th>Feature</th>
<th>MUC 12.77%</th>
<th>57.66%</th>
<th>20.91</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline B 3</td>
<td>35.1%</td>
<td>91.05%</td>
<td>50.67</td>
</tr>
<tr>
<td>CEAF 47.80%</td>
<td>17.29%</td>
<td>25.40</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>R</th>
<th>P</th>
<th>F1</th>
<th>ΔF1</th>
</tr>
</thead>
<tbody>
<tr>
<td>MUC</td>
<td>0.00%</td>
<td>33.33%</td>
<td>0.01</td>
</tr>
<tr>
<td>B³</td>
<td>26.27%</td>
<td>99.98%</td>
<td>41.61</td>
</tr>
<tr>
<td>CEAF</td>
<td>52.88%</td>
<td>13.89%</td>
<td>22.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>R</th>
<th>P</th>
<th>F1</th>
<th>ΔF1</th>
</tr>
</thead>
<tbody>
<tr>
<td>MUC</td>
<td>0.39%</td>
<td>60.86%</td>
<td>0.79</td>
</tr>
<tr>
<td>B³</td>
<td>26.59%</td>
<td>99.72%</td>
<td>41.99</td>
</tr>
<tr>
<td>CEAF</td>
<td>52.76%</td>
<td>13.99%</td>
<td>22.11</td>
</tr>
</tbody>
</table>

The fact that for three of the features, removing the feature actually improved performance is troubling. Possibly these features were too noisy; we need to improve the baseline features for future experiments.

6 Conclusions

Semantic information is necessary for many tasks in natural language processing. Most often this information is used in the form of relationships between words – for example, how semantically similar two words are, or which nouns are the objects of a verb. However, it is likely that humans make use of much higher-level information than the similarity between two concepts when processing language (Abelson, 1981). We attempted to take advantage of recent developments in automatically acquiring just this sort of information, and demonstrated the possibility of making use of it in NLP tasks such as coreference. However, we need to improve both the implementation and data for this approach to be practical.

For future work, we intend to investigate avenues for improving the acquisition and use of the narrative schema information, and also compare narrative schema with other types of semantic information in coreference resolution. Because coreference information is central to the extraction of narrative schema, the joint learning of coreference resolution and narrative schema is another area we would like to explore.

References


