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

Proceedings of the 15th Conference on Computational Natural Language Learning: Shared Task, pages 86–92,
Portland, Oregon, 23-24 June 2011.

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, than 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.

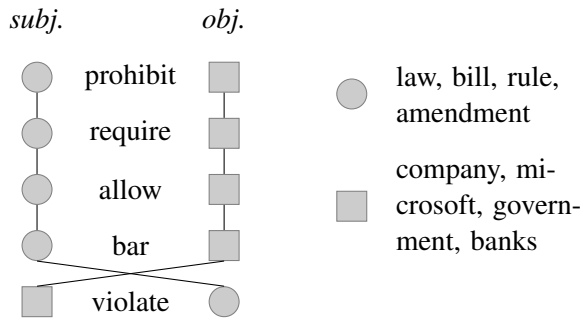

Figure 1: An example narrative schema with two roles.

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.

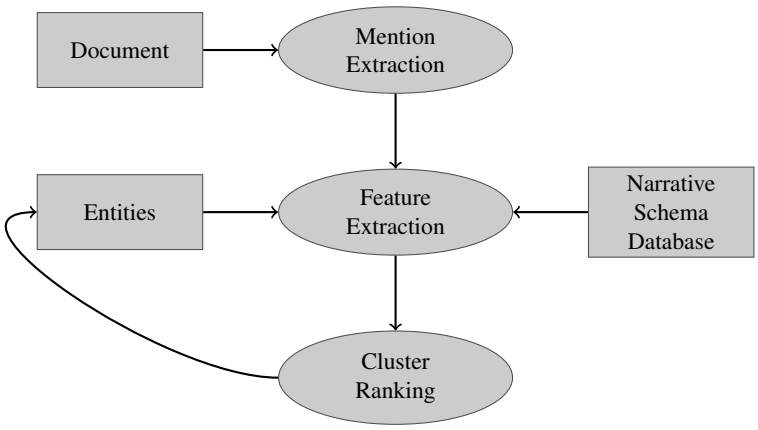

Figure 2: System execution flow

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. [1]

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

[1]The set of types is: PERSON, NORP, FACILITY, ORGANIZATION, GPE, LOCATION, PRODUCT, EVENT, WORK, LAW, LANGUAGE, DATE, TIME, PERCENT, MONEY, QUANTITY, ORDINAL, CARDINAL

Features involving m_j only	
SUBJECT	Y if m_j is the grammatical subject of a verb; N otherwise
*NE_TYPE1	the NE label for m_j if there is one else NONE
Features involving m_k only	
DEFINITE	Y if the first word of m_k is *the*; N otherwise
DEMONSTRATIVE	Y if the first word of m_k is one of *this*, *that*, *these*, or *those*; N otherwise
DEF_DEM_NA	Y if neither DEFINITE nor DEMONSTRATIVE is Y; N otherwise
PRONOUN2	Y if m_k is a personal pronoun; N otherwise
PROTYPE2	nominative case of m_k if m_k is a pronoun or NA if it is not (e.g., HE if m_k is *him*)
NE_TYPE2	the NE label for m_k if there is one
Features involving both m_j and m_k	
DISTANCE	how many sentences separate m_j and m_k; the values are A) same sentence, B) previous sentence, and C) two sentences ago or more
HEAD_MATCH	Y if the head words are the same; N otherwise
PRONOUN_MATCH	if either of m_j and m_k is not a pronoun, NA; if the nominative case of m_j and m_k is the same, C; I otherwise
*NE_TYPE′	the concatenation of the NE labels of m_j and m_k (if either or both are not labelled NEs, the feature is created using NONE as the corresponding label)
SCHEMA_PAIR_MATCH	Y if m_j and m_k appear in the same role in a schema, and N if they do not
Features involving c_j and m_k	
SCHEMA_CLUSTER_MATCH	a cluster-level feature between m_k and c_j (details in Section 4.7)

Table 1: Features implemented in our coreference resolver. Binary-valued features have values of **Y**ES or **N**O. Multi-valued features are converted into equivalent sets of binary-valued features before being used to create the cluster-level features used by the ranker.

predicates: NONE, MOST-FALSE, MOST-TRUE, and ALL.

To be precise, a feature F may be thought of as a function taking m_j as a parameter, e.g., $F(m_j)$. To simplify notation, features which apply to the *pair* m_j,m_k take m_k as an implicit parameter. The logical predicates then compare the two counts $n = |\{m_j \mid F(m_j) = true\}|$ and $C = |c_j|$. The resulting features are shown in Table 2.

NONE_F	TRUE iff $n = 0$
MOST-FALSE_F	TRUE iff $n < \frac{C}{2}$
MOST-TRUE_F	TRUE iff $\frac{C}{2} \leq n < C$
ALL_F	TRUE iff $n = C$

Table 2: Cluster-level features created from binary-valued feature F

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_CLUSTER_MATCH

The SCHEMA_CLUSTER_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_CLUSTER_MATCH$_n$ 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} schema 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^{rank}, an efficient implementation for training ranking SVMs (Joachims, 2006) [3].

[2] http://www.nltk.org/

[3] http://svmlight.joachims.org/

		R	P	F_1
CLOSED	MUC	12.45%	50.60%	19.98
	B^3	35.07%	89.90%	50.46
	CEAF	45.84%	17.38%	25.21
			Overall score:	31.88
OPEN	MUC	18.56%	51.01%	27.21
	B^3	38.97%	85.57%	53.55
	CEAF	43.33%	19.36%	26.76
			Overall score:	35.84

Table 3: Official system results

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.

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

		R	P	F_1
Baseline	MUC	12.77%	57.66%	20.91
	B^3	35.1%	91.05%	50.67
	CEAF	47.80%	17.29%	25.40
+SCHEMA	MUC	12.78%	54.84%	20.73
	B^3	35.75%	90.39%	51.24
	CEAF	46.62%	17.43%	25.38

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

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 B^3 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 B^3 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.

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 PRONOUN2, PROTYPE2, and PRONOUN_MATCH; -DEF_DEM removes DEFINITE, DEMONSTRATIVE, and DEF_DEM_NA; and -NE removes the named entity features.

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

Baseline	MUC	12.77%	57.66%	20.91	
	B^3	35.1%	91.05%	50.67	
	CEAF	47.80%	17.29%	25.40	
		R	P	F_1	ΔF_1
-HEAD	MUC	0.00%	33.33%	0.01	-20.90
	B^3	26.27%	99.98%	41.61	-9.06
	CEAF	52.88%	13.89%	22.00	-3.40
-DIST	MUC	0.39%	60.86%	0.79	-20.12
	B^3	26.59%	99.72%	41.99	-8.68
	CEAF	52.76%	13.99%	22.11	-3.29
-SUBJ	MUC	12.47%	47.69%	19.78	-1.13
	B^3	36.54%	87.80%	51.61	0.94
	CEAF	43.75%	17.22%	24.72	-0.68
-PRO	MUC	18.36%	55.98%	27.65	6.74
	B^3	37.45%	85.78%	52.14	1.47
	CEAF	47.86%	19.19%	27.40	2.00
-DEF_DEM	MUC	18.90%	51.72%	27.68	6.77
	B^3	41.65%	86.11%	56.14	5.47
	CEAF	46.39%	21.61%	29.48	4.08
-NE	MUC	22.76%	49.5%	31.18	10.27
	B^3	46.78%	84.92%	60.33	9.66
	CEAF	45.65%	25.19%	32.46	7.06

Table 5: Effect of each feature on performance.

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 aquiring 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 aquisition 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

Robert P. Abelson. 1981. Psychological status of the script concept. *American Psychologist*, 36(7):715–729.

Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised Learning of Narrative Schemas and their Participants. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 602–610, Suntec, Singapore.

Nathanael Chambers and Dan Jurafsky. 2010. A database of narrative schemas. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC 2010)*, Malta.

Hal Daumé and Daniel Marcu. 2005. A large-scale exploration of effective global features for a joint entity detection and tracking model. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing - HLT '05*, pages 97–104, Morristown, NJ, USA.

Aria Haghighi and Dan Klein. 2010. Coreference resolution in a modular, entity-centered model. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 385–393.

Thorsten Joachims. 2006. Training linear SVMs in linear time. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD 06*, pages 217–226.

Vincent Ng. 2010. Supervised noun phrase coreference research: the first fifteen years. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1396–1411.

Simone Paolo Ponzetto and Michael Strube. 2006a. Exploiting semantic role labeling, WordNet and Wikipedia for coreference resolution. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 192–199.

Simone Paolo Ponzetto and Michael Strube. 2006b. Semantic role labeling for coreference resolution. In *Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational*

Linguistics - EACL '06, pages 143–146, Morristown, NJ, USA.

Sameer Pradhan, Lance Ramshaw, Ralph Weischedel, Jessica MacBride, and Linnea Micciulla. 2007. Unrestricted Coreference: Identifying Entities and Events in OntoNotes. In *International Conference on Semantic Computing (ICSC 2007)*, pages 446–453.

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 Fifteenth Conference on Computational Natural Language Learning (CoNLL 2011)*, Portland, Oregon.

Altaf Rahman and Vincent Ng. 2009. Supervised Models for Coreference Resolution. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 968–977, Singapore.

Altaf Rahman and Vincent Ng. 2011. Narrowing the Modeling Gap: A Cluster-Ranking Approach to Coreference Resolution. *Journal of Artificial Intelligence Research*, 40:469–521.

Roger C. Schank and Robert P. Abelson. 1977. *Scripts, plans, goals and understanding: An inquiry into human knowledge structures.* Lawrence Erlbaum, Oxford, England.