Entity-Based Noun Phrase Coreference Resolution

Xiaofeng Yang, Jian Su, and Lingpeng Yang

Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore, 119613
{xiaofengy, sujian, lpyang}@i2r.a-star.edu.sg

Abstract. In this paper we propose an NP coreference resolution system which does resolution on the entity-level. The framework of the system is presented and different resolution strategies are investigated.

1 Introduction

Coreference resolution is the process of linking multiple expressions which refer to the same entity. Traditional supervised machine learning approaches (e.g. [1, 2, 3]) do resolution based on the mention-level. Specifically, a pairwise classifier is learned and used to determine whether or not two NPs in a document refer to the same entity in the world. However, as an individual mention usually lacks adequate information about its referred entity (e.g, we could not know the gender or the name of "the president"), it is often difficult to determine whether or not two NPs refer to the same entity simply from the pair itself. Recent research ([4, 5]) has revealed that entity information could help resolution. In our work we would like to further study how to effectively incorporate the entity information into coreference resolution. The framework of such a entity-based system is presented and different resolution strategies are investigated in this paper.

2 Baseline: A Mention-Based System

We built a Mention-Mention based system as the baseline, which adopts a learning framework similar to the paradigm proposed by Soon et al. [2].

Each instance takes the form of $\boldsymbol{i}\{\mathrm{NP}_i,\ \mathrm{NP}_j\}$, which is associated with a feature vector consisting of 12 features ($f_1 \sim f_{12}$) as described in Table 1. During training, for each anaphor NP_j in a given text, a positive instance is generated by pairing NP_j with its closest antecedent. A set of negative instances is also formed by NP_j and each NP occurring between NP_j and NP_i.

When the training instances are ready, a classifier is learned by C5.0 algorithm [6]. During resolution, each encountered noun phrase, NP_j, is paired in turn with each preceding noun phrase, NP_i. For each pair, a testing instance is created and then presented to the decision tree, which returns a confidence value (CF) indicating the likelihood that they co-refer. NP_j will be linked to the NP with the maximal CF (above 0.5).

A. Gelbukh (Ed.): CICLing 2005, LNCS 3406, pp. 218–221, 2005.

Table 1. The features used in the coreference resolution system

Features describing the relationships between NP_j and NP_i		
1.	Type_1	the type of NP_j (Indefinite NP, Definte NP, Pronoun, . . .)
2.	Type_1	the type of NP_j (Indefinite NP, Definte NP, Pronoun, . . .)
3.	NumAgree	NP_i and NP_j are compatible in number
4.	GenderAgree	NP_i and NP_j are compatible in gender
5.	Sdist	the distance between NPi and NPj in sentences
6.	Pdist	the distance between NPi and NPj in paragraphs
7.	Appositive	NP_i and NP_j are in an appositive structure
8.	NameAlias	NP_i and NP_j are in an alias of the other
9.	HeadStrMatch	NP_i and NP_j contain the same head string
10.	FullStrMatch	NP_i and NP_j contain the same string
11.	StrSim_1	The string similarity of NP_j against NP_i
12.	StrSim_2	The string similarity of NP_i against NP_j
Features describing the relationships between NP_j and ENT_i		
13.	C_NumAgree	NP_j is compatible in number with any mention of ENT_i
14.	C_GenAgree	NP_j is compatible in gender with any mention of ENT_i
15.	C_Appositive	NP_j is in an appositive structure with a mention of ENT_i
16.	C_NameAlias	NP_j is in an alias of a mention of ENT_i; else 0
17.	C_HeadStrMatch	NP_j contains the same head string as a mention of ENT_i
18.	C_FullStrMatch	NP_j contains the same string as a mention of ENT_i
19.	C_MaxStrSim	The maximal string similarity between NP_j and the mentions of ENT_i
20.	C_StrSim	The string similarity of NP_j against ENT_i

3 The Entity-Based System

3.1 Instance Representation

An instance in our approach has the form of $\boldsymbol{i}\{ENT_i, ENT_j\}$, where ENT_i and ENT_j are two partial entities under consideration.

In our system, each instance is represented as a set of 20 features as shown in Table 1. The features are supposed to capture the properties and relationships between two entities. Note that here NP_i is the last mention in ENT_i, while NP_j is the first mention in ENT_j.

An instance is labelled as positive if ENT_i and ENT_j are of the same entity, or negative if otherwise.

3.2 Training Procedure

Given an annotated training document, we process the noun phrases from beginning to end. For each anaphoric noun phrase NP_j, we represent it as a partial entity ENT_j. ENT_j will be paired with each preceding coreferential chain, ENT_i, to form a training instance. The process continues until the chain to which ENT_j belongs is found.

3.3 Resolution Procedure

The resolution could be thought of as a clustering problem. Initially, each NP in a given documents is represented as a single cluster, and then small clusters referring to the partial entities are merged together to form a complete entity.

We use two similarity metrics to evaluate the likelihood that two partial entity, ENT_i and ENT_j, are co-referring:

– **Single Similarity**: it simply uses the confidence returned by the classifier. Suppose function CF is the confidence value of an instance

$$Similarity(ENT_i, ENT_j) = CF_{i\{ENT_i, ENT_j\}} \quad (1)$$

– **Maximal Similarity**: ENT_i is divided into several sub_clusters. The similarity is the maximal confidence between ENT_j and the sub_clusters. Specifically, Suppose ENT_i contains k mentions, $M_{i1}, M_{i2}, \ldots, M_{ik}$. Let $SubSet_i = \{ENT_{id} | ENT_{id} = \{M_{i1}, \ldots, M_{id}\}, 1 \leq d \leq k\}$, then

$$Similarity(ENT_i, ENT_j) = \max_{ENT_{id} \in SubSet_i} CF_{i\{ENT_{id}, ENT_j\}} \quad (2)$$

And three clustering strategies are considered to group the partial entities:

– **Simple Clustering**: Each cluster is simply merged to the best preceding cluster with the highest similarity (above 0.5), if any.

– **Incremental Clustering**: Clusters are processed from left to right. A cluster is merged into the best preceding cluster, if any, before proceeding to subsequent ones.

– **Greedy Clustering**: Clustering is done iteratively. In each iteration, every two clusters are tested and the pair with the highest similarity is merged together. The iteration continues until no remaining clusters could be merged.

4 Evaluation and Discussion

In our study we used the standard MUC-6 and MUC-7 coreference corpora. In each data set, around 30 “dry-run” documents were annotated for training as well as 20-30 documents for testing.

In the experiments we evaluated our system under the two similarity metrics and the three clustering strategies. The performance is listed in Table 2. The Recall and Precision were calculated based on the standard MUC coreference resolution scoring scheme [7].

The baseline system produces the F-measure of 60.7% (MUC-6) and 63.7% (MUC-7). The score is similar to that of Soon et al.’s system (62.6% and 60.4%).

Compared with the baseline, our entity-based system obtains large gain (7.1% for MUC-6 and 2.5% for MUC-7) in Precision, with slight loss (less than 1%)

Table 2. Experimental Results

	MUC-6						MUC-7					
	Single			Maximal			Single			Maximal		
	R	P	F	R	P	F	R	P	F	R	P	F
Baseline	**65.2**	56.8	60.7				**68.4**	59.5	63.6			
Simple	64.7	60.2	62.3	64.7	60.2	62.3	67.8	60.4	63.9	67.8	60.4	63.9
Incremental	63.7	63.5	63.6	64.8	59.8	62.3	66.2	61.8	64.2	67.8	**62.4**	**65.0**
Greedy	63.4	**63.9**	**63.7**	64.7	59.8	62.2	66.5	62.0	64.2	67.8	**62.4**	**65.0**

in Recall. Overall, the system achieves F-measure up to about 3% higher than the baseline. This result indicates that our entity-based system is effective for coreference resolution.

From the table, the performance difference under the two similarity metrics is obscure. For MUC-6, *single-similarity* is slightly better than *maximal-similarity*, while the latter seems to be superior for MUC-7.

In comparing the three clustering strategies, we observe no apparent performance difference between *incremental-clustering* and *greedy-clustering*. By contrast, in most cases these two clustering methods outperform *simple-clustering*, especially in Precision. It should be due to the fact that simple clustering does not use the entity information during resolution. The results further prove that entity information will help not only training, but also resolution.

References

1. McCarthy, J., Lehnert, Q.: Using decision trees for coreference resolution. In: Proceedings of the 14th International Conference on Artificial Intelligences. (1995) 1050–1055
2. Soon, W., Ng, H., Lim, D.: A machine learning approach to coreference resolution of noun phrases. Computational Linguistics **27** (2001) 521–544
3. Ng, V., Cardie, C.: Improving machine learning approaches to coreference resolution. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia (2002) 104–111
4. Yang, X., Su, J., Zhou, G., Tan, C.: An np-cluster approach to coreference resolution. In: Proceedings of the 20th International Conference on Computational Linguistics, Geneva (2004)
5. Luo, X., Ittycheriah, A., Jing, H., Kambhatla, N., Roukos, S.: A mention-synchronous coreference resolution algorithm based on the bell tree. In: Proceedings of 42th Annual Meeting of the Association for Computational Linguistics. (2004) 135–142
6. Quinlan, J.R.: C4.5: Programs for machine learning. Morgan Kaufmann Publishers, San Francisco, CA (1993)
7. Vilain, M., Burger, J., Aberdeen, J., Connolly, D., Hirschman, L.: A model-theoretic coreference scoring scheme. In: Proceedings of the Sixth Message understanding Conference (MUC-6), San Francisco, CA, Morgan Kaufmann Publishers (1995) 45–52