

Exploration of Coreference Resolution: The ACE Entity Detection and Recognition Task

Ying Chen¹ and Kadri Hacioglu²

¹ Center for Spoken Language Research, University of Colorado at Boulder, USA

² Center for Spoken Language Research, University of Colorado at Boulder, USA

Abstract. In this paper, we consider the coreference resolution problem in the context of information extraction as envisioned by the DARPA Automatic Content Extraction (ACE) program. Given a set of entity mentions referring to real world entities and a similarity matrix that characterizes how similar those mentions are, we seek a set of entities that are uniquely co-referred to by those entity mentions. The quality of the clustering of entity mentions into unique entities significantly depends on the quality of (1) the similarity matrix and (2) the clustering algorithm. We explore the coreference resolution problem along those two dimensions and clearly show the tradeoff among several ways of learning similarity matrix and using it while performing clustering.

1 Introduction

Availability of large annotated corpora and advances in machine learning techniques have enabled Information Extraction (IE) to become an active research area. The detection and recognition of entities as unique objects that belong to the physical world are the initial but crucial steps in many NLP applications including question answering, summarization etc. Here we present a detailed exploration of an entity detection and recognition (EDR) system that has been developed as a part of a system that participated in the ACE 2005 Evaluation.

The EDR system consists of two steps; first, it detects all mentions of entities occurring in raw text (mention detection) and, then, resolves all different mentions of an entity into an object that uniquely represents that entity (coreference resolution). The coreference resolution can be considered as a clustering problem; given a set of objects to be clustered and a similarity matrix among pairs of objects to guide the clustering algorithm. We mainly focus on how to determine the similarity matrix and how to use it. Assuming that the correct mentions are available with the ideal similarity matrix (it just tells the pair of objects are similar or not), the coreference resolution is trivial by taking connected sets of mentions as unique entities. However, the estimated similarity matrix are usually far from perfect and, to mitigate that, one needs to exploit relatively advanced clustering algorithms, for example, an agglomerative or correlation clustering algorithm. Intuitively, a better estimation of similarity matrix reduces the burden on clustering and an advanced clustering strategy reduces the burden on the accurate estimation of similarity matrix. In this paper, we

explore the tradeoff between those phenomena for entities occurring in English and Chinese languages.

Our framework for the exploration of the coreference resolution problem consists of two case studies; (1) advanced similarity matrix estimation (or learning) with simpler clustering, and (2) simpler similarity matrix estimation (or learning) with advanced clustering. Here, we consider the similarity matrix estimation as a link classification problem and the output of the classifier is used to estimate the similarity matrix. In our exploration, we try to improve the classification, and hence the similarity matrix estimation, by extending our classification models from mention-level to entity-level. Similarly, we vary the complexity of the clustering algorithm by using different linkage and clustering schemes. It is interesting to note that the linkage schemes in clustering can also be considered as the adaptation schemes for the similarity matrix as we proceed in clustering. We clearly demonstrate that exploitation of this coupling between classification and clustering is very important to overcome the problems of inaccurate similarity matrix estimation and improve the overall system performance.

2 Related Work

In this section we focus on and describe related work based on data-driven approaches, and exclude those that are rule-based. There are several supervised or unsupervised machine learning (ML) methods that can be applied to coreference resolution. Usually, coreference resolution is cast as a classification problem in the supervised ML framework (Soon et al., 2001; Ng and Cardie, 2002) and as a clustering problem in the unsupervised machine learning framework (Cardie and Wagstaff, 1999). Here, in contrast, we consider coreference resolution as a clustering problem with similarity matrix learning, either supervised or unsupervised. The outputs of the pairwise classifier, such as in (Soon et al., 2001), can be used to construct the similarity matrix. Alternatively, the ad-hoc distance metric defined over the feature representation of each reference (or mention) in an unsupervised ML framework can be used to construct a similarity matrix. Therefore, both approaches differ only in similarity matrix learning and allow a rich collection of clustering algorithms to be equivalently applied. Compared to previous related work, our view allows us to couple classification and clustering stages for the estimation and adaptation of the similarity matrix for better performance. That is, instead of focusing separately on classification and clustering we focus on the similarity matrix and use classification for its initial estimation and clustering for its incremental adaptation using several linkage schemes. Coupling of classification and clustering stages can also be seen in (Li and Roth, 2005; Luo et al., 2004; McCallum and Wellner, 2003) in different ways. All report performance improvements when compared to their baseline systems. However, no much work has been done to compare the contribution of the similarity metrics and clustering to coreference resolution in detail and explain why the performance will be improved with the more accurate similarity metrics or the complex clustering. This is the focus of our paper.

3 Methodology

As mentioned earlier, the performance of coreference resolution depends on the coupling between classification and clustering. We will describe our coreference system that improves the similarity matrix through this coupling.

3.1 Classification

3.1.1 Entity-level model In coreference resolution, a similarity metric for a pair of mentions is required to cluster. Here, we consider the learning of similarity metrics (elements of the similarity matrix) as link/no link classification problem of a mention pair, and the confidence of a linking decision from the classifier can be used as the similarity metric of the two mentions under consideration. However, in most of the previous work, the features for classification are extracted by considering the mention-pairs in focus and their local context. Intuitively, there might be several cases that the mention-level features might not supply enough information to make better linking decisions. For example, consider a document section that includes the fragment

“Clinton also touched on the matter of American Edmond Bob who is being tried in a closed court in Russia on charges of spying. The United States believe he is innocent of these charges and are demanding his release on humanitarian grounds. The official said Putin understands our concern.”

Assume that “Edmond Bob” and “he” are correctly linked. Using only mention pairs and their local context for making decisions might result in an incorrect and conflicting link of “Putin” and “he”. However, this incorrect decision could have been overcome by using the information that “he” has been already linked to another mention “Edmond Bob” to establish a partial entity. Therefore, it becomes less likely to link “he” and “Putin” since the fact that “Edmond Bob” and “Putin” are two different personal names that can be captured through entity-level features. This clearly indicates the necessity of using information beyond mention pairs and their local context.

One way to go beyond the local context of the mention pairs is to take the advantage of the previous linking decisions. Using those decisions one can construct the corresponding partial entity for each mention in the mention-pair in focus. Considering the previous example, there is a link between “he” and “Edmond Bob”, which means that there is a partial entity including “he” and “Edmond Bob”. Now, when we try to decide on linking “Putin” to “he”, we can trace back the previous link decision between “he” and “Edmond Bob”. This allows us to compare “Putin” to the another proper name “Edmond Bob”. Since the partial entities might contain one or more mentions, it is not computationally feasible to pair each mention to other mentions inside a partial entity to extract features for classification. Therefore, we need to define a single mention, called the canonical mention (Luo et al., 2004), that best represents the partial entity. For each partial entity, we select a mention as the canonical mention according to the following ordered preference list: longest NAME mention, longest NOMINAL mention and longest PRONOUN mention. In doing so, one can easily derive

entity-level features from the pair of canonical mentions of the corresponding partial entities.

3.1.2 Feature extraction In the preceding section, we briefly talked about the mention-level features. Table 1 shows the set of features extracted in this study. It is an extension of the basic features in (Luo et al., 2004). Note that, due to the availability of linguistic resources and differences in English and Chinese, the corresponding feature sets are different in size and types. We organized source of features into three broad groups: (i) `mention_string`: the information containing within the mention string, (ii) `mention_context`: the information within the mention context, (iii) `mention_pair`: the information related with the mention pair.

The feature set in Table 1 serves as the baseline. Recall that one can also extract entity-level features in accordance with Table 1 using corresponding canonical mention pairs to create the entity-level feature set. Now we have two classification models: (i) Mention-level model (baseline model): Mention-level features in Table 1; (ii) Entity-level model: Baseline features plus entity-level features between the pair of canonical mentions as described in Table 1 (for the features in mention-pair category, we only implement (13) and (14)).

3.2 Clustering

3.2.1 Linkage schemes The similarity matrix, defined as a collection of similarity metrics among pairs of all mentions, provides guidance when we try to decide if two clusters, each with a single mention, can be merged. As we progress in clustering we start to have clusters with more than one mention. So, we need to develop a linkage scheme to compute the similarity between two clusters to be used for merging. In this paper, in addition to commonly used linkage schemes, such as maximum, minimum and average linkage schemes, we have developed another linkage scheme, which has been customized for the coreference resolution problem in consideration. Our linkage scheme can exploit entity-level information. We describe it after a brief description of the standard linkage schemes:

Maximum Linkage: The distance of a cluster to another cluster is the maximum of the distances between items of each cluster.

Minimum Linkage: The distance of a cluster to another cluster is the minimum of the distances between items of each cluster.

Average Linkage: The distance of a cluster to another cluster is the average of the distances between items of each cluster.

Longest Canonical Maximum Linkage: Here we combine the canonical mention selection strategy in (Luo et al., 2004) with the linkage scheme proposed in (Daume and Marcu, 2005). It consists of the following steps: (1) Choose the canonical mention for each cluster as in (Luo et al., 2004), which is briefly described in Section 3.1, (2) For each partial entity, compute the similarity metric of its canonical mention to the other partial entity cluster as described in (Daume and Marcu, 2005), (3) Choose the maximum among the two linkage metrics.

Category	Features	Remark	Language
Mention_head	(1) Spell, Count, POS	Same as in (Luo, 2004)	English & Chinese
	(2) Mention type	The ACE mention type	
	(3) Entity type	The ACE entity type	
	(4) Gazetteer info	Information from gazetteer	
	(5) Low-case spell	Low-case of the head word	English
	(6) Capitalized word	# of capitalized word	
	(7) Definite word	Whether the head word is a definite word	
	(8) Gender, number, possessive, reflexive	Same as in (Luo, 2004)	
Mention_context	(9) Dependent info	The dependent word, pos, and relation	English & Chinese
	(10) Possessive info	Whether there is a possessive indicator	
	(11) Verb info	The nearest verb string	
	(12) Preposition word	Whether there is a preposition word around	
Mention_pair	(13) String match	Same as in (Luo, 2004)	English & Chinese
	(14) Token, sentence, mention distance	Same as in (Luo, 2004)	
	(15) Acronym	Same as in (Luo, 2004)	
	(16) Apposition	Same as in (Luo, 2004)	

Table 1. The mention-level feature set for classification; the same table is used for both basic and canonical mention pairs.

3.2.2 Clustering schemes Clustering schemes we describe here are about the order of clustering process given the clusters, independent of linking scheme used. Clustering can be viewed from several different perspectives. The traditional way processes the mentions in the order of left to right, like left2right first-link and left2right best-link methods. The other way is linking the best similar pairs in a bottom-up manner as in agglomerative clustering. Yet another way treats clustering as a graph problem and tries to solve clustering using the graph theory, such as correlation clustering.

Left2right best-link clustering: process the mentions in the order of left to right. For each mention, link it to the most similar previous cluster with the confidence greater than the fixed threshold (0.5 in our algorithm).

Agglomerative bottom-up clustering: initialize each mention as a cluster. Iteratively merge the most similar two clusters until the distance of any two clusters is below a fixed threshold (0.5 in our algorithm).

Unweighted correlation clustering: the graph outputted from the classifier is an unweighted complete graph with "Link" or "Nolink" tags. Unweighted correlation clustering algorithm tries to find a partition which agrees with the original graph as much as possible on the edge tags. For details, the reader is referred to (Bansal et.al., 2002).

4 Experiments

We report experimental results based on ACE 2005 English and Chinese training data, given the golden mentions. For each language, 20% of data is reserved for development and 80% for training. All the following performances are reported for the development data.

4.1 Classification Evaluation

We select Yamcha toolkit³ for implementing SVM based classifiers. To avoid the impact of clustering on performance, we used the naïve-clustering algorithm: the mentions are partitioned into disjoint connected groups according the linking decisions. The results of the two models are shown in Table 2. Here “AF” is the ACE F score and is calculated using the ACE scoring script⁴. As can be easily seen from Table 2, the coreference resolution performance consistently increases with the complexity of the feature set in both languages. It is interesting to note that the improvement of coreference resolution with the entity-level model is almost due to significant increase in recall with marginal drop in precision. With the entity-level features, the entity-level model can capture information more than the mention-level model, effectively increasing the recall without sacrificing much the precision. Also, the relatively larger increase in the recall for English when compared to Chinese indicates that the English coreference resolution benefits from entity-level information more than the Chinese one. Comparing the overall absolute performances, it seems that the coreference resolution is easier for Chinese than for English. This is probably due to two reasons; (i) diverse nature of English data, and (ii) the difficulty of pronoun resolution in English.

4.2 Clustering Evaluation

4.2.1 Linkage schemes evaluation To fairly compare the different linkage schemes described in 3.2.1, we run the agglomerative (bottom-up) clustering scheme with the similarity metrics derived from the baseline classifier and show the performances in Table 3 with the same format in Table 2. From Table 3, except the maximum linkage scheme, we found that the same phenomenon shows up as in entity-level classification: the improvement of coreference resolution with the entity-level model is almost due to significant increase in recall with marginal drop in precision. This indicates that the linkage scheme can work as the entity-level classifier: capture the entity-level information to improve the linkage similarity that can be viewed as an extension of similarity matrix in clustering. At the same time, the last three linkage schemes seem to achieve the similar performances for English and Chinese. With Student’s t test, we found that, for English, “longest canonical maximum linkage” outperform “average linkage” and “minimum linkage” with 95% confidence, and for Chinese, the three linkages really perform similarly.

³ <http://chasen.org/~taku/software/yamcha/>

⁴ <http://www.nist.gov/speech/tests/ace/ace05/index.htm>

	Mention-level AF (precision/recall)	Entity-level AF (precision/recall)
English	71.6 (91.4/58.9)	82.9 (90.8/76.3)
Chinese	82.9 (91.9/75.6)	90.8 (93.3/88.4)

Table 2. Performances of mention-level and entity-level classification models.

	Max	Min	Average	Longest Canonical Maximum
English	72.4(92.4/59.5)	86.1(86.8/85.5)	87.4(88.9/86.0)	88.6(90.3/86.9)
Chinese	83.3(92.4/75.9)	90.7(91.8/89.6)	91.2(92.5/89.8)	91.4(93.3/89.5)

Table 3. The comparison of different linkage schemes under bottom-up clustering on the baseline classifier.

	Left2right best-link clustering	Bottom-up clustering	Correlation clustering
English	88.6	88.6	83.5
Chinese	91.9	91.4	89.1

Table 4. The comparison of clustering schemes on the baseline classifier and longest canonical maximum linkage (ACE F score).

	Entity-level classifier + customized clustering	Mention-level classifier + customized clustering
English	87.3	88.6
Chinese	91.8	91.4

Table 5. The comparison of the coupling of classification and clustering.

4.2.2 Clustering schemes evaluation To avoid impact on performance from classification and linkage, we compare the performance variation with clustering schemes using the same classifier (baseline classifier) and the same linkage scheme (Longest Canonical Maximum Linkage). The performances of the three clustering algorithms are shown in Table 4. For both English and Chinese languages, the bottom-up and left2right best-link clustering have shown similar performance, while outperforming the correlation clustering with 95% confidence. Although correlation clustering is a global optimization resolution for clustering, it is unweighted version that we have implemented and it makes use of the hard decisions from the classifier, so there is no chance to improve the imperfect similarity matrix through the linkage schemes and therefore its performance is worst comparing other clustering schemes that can incorporate the linkage schemes.

4.3 Coupling of Classification and Clustering

From the experiments in 4.1 and 4.2, we found that the best coupling of classification and clustering is the combination of the entity-level classification and the

customized clustering strategy: longest canonical maximum linkage and agglomerative bottom-up clustering. The performance of the best coupling is shown in Table 5. It is a little surprise that the best coupling does not outperform the combination of the mention-level classification and the customized clustering strategy. With Student's t test, the two coupling perform similar for both English and Chinese with 95% confidence. We are currently exploring the reason for the unexpected performances.

5 Conclusion

We have considered the coreference resolution problem as a clustering problem. We have argued that there are two ways to improve the performance: improved similarity matrix estimation via classification, or incremental adaptation of the similarity matrix in clustering through customized linkage schemes. In this paper, we have done extensive exploration of those phenomena within the context of content extraction as envisioned by the ACE program. We have provided several experimental results that we believe that they will provide useful guidance for coreference resolution.

References

1. Nikhil Bansal, Avrim Blum, and Shuchi Chawla: Correlation clustering. The 43rd Annual Symposium on Foundations of Computer Science (FOCS). (2002)
2. C. Cardie and K. Wagstaff: Noun phrase coreference as clustering. In Proc. of Empirical Methods in Natural Language Processing. (1999)
3. Hal Daume III, Daniel Marcu: A Large-Scale Exploration of Effective Global Features for a Joint Entity Detection and Tracking Model. In Proc. of Human Language Technology Conference. (2005)
4. X. Li and D. Roth, Discriminative Training of Clustering Functions: Theory and Experiments with Entity Identification. In Proc. of the Annual Conference on Computational Natural Language Learning (CoNLL). (2005)
5. Xiaoqiang Luo, Abe Ittycheriah, Hongyan Jing, Nanda Kambhatla and Salim Roukos: A Mention-Synchronous Coreference Resolution Algorithm Based on the Bell Tree. In Proc. Of the Association for Computational Linguistics. (2004)
6. Andrew McCallum and Ben Wellner: Toward Conditional Models of Identity Uncertainty with Application to Proper Noun Coreference. In IJCAI Workshop on Information Integration on the Web. (2003)
7. Joseph F. McCarthy and Wendy G. Lehner: Using Decision Trees for Coreference Resolution. In Proc. Of the International Conference on Artificial Intelligence. (1995)
8. Thomas S. Morton: Coreference for NLP Applications. In Proc. Of the Association for Computational Linguistics. (2000)
9. Vincent Ng and Claire Cardie: Improving Machine Learning Approaches to Coreference Resolution. In Proc. Of the Association for Computational Linguistics. (2002)
10. Wee M. Soon, Hwee T. Ng, and Chung Y. Lim: A Machine Learning Approach to Coreference Resolution of Noun Phrases. In Computational Linguistics. (2001)

This article was processed using the L^AT_EX macro package with LLNCS style