Coreference Resolution with Markov Logic

Fei Chen
Department of Computer Sciences
University of Wisconsin Madison
Madison, WI 53706

Abstract

Joint unsupervised coreference resolution using Markov Logic is a relatively new method and shows a good performance. However, it needs some extra human efforts and can't distinguish mentions with the same head which are different entities. In our project, we implement MLN to coreference resolution in a simple and efficient way with little human effort and distinguish some mentions with the same head based on their neighbors. On our toy data, our improvements outperform the original MLN rules, and achieve high accuracies.

Introduction

Coreference resolution is the process in which we identify the noun phrases (i.e., mentions) that refer to a same real-world entity(Ng 2008). In Natural Language Processing, coreference resolution has become one of core research topics(Clark and Gonzalez 2008). Many machine learning approaches to coreference resolution are supervised, assume that data are independent and do pairwise classification(McCallum and Wellner 2002)(Denis and Baldridge 2007). Markov Logic Networks (MLN) is a new machine learning algorithm, which considers relations among items(Richardson and Domingos 2006). MLN has been applied to joint unsupervised coreference resolution, however it needs extra human efforts to do necessary tagging tasks and can't distinguish mentions with the same head which are different entities indeed(Poon and Domingos 2008). In this project, we improve their method to reduce human effort and raise accuracies. On our toy data, our improvements outperform the original method, and achieve high accuracies.

Background

Poon and Domingos (2008) presented a joint unsupervised rule-based coreference resolution method using MLN. MLN is a collection of first-order knowledge base with a weight attached to each formula. Taken as a Markov network, the verticies of the network graph are atomic formulas, and the edges are the logical connectives used to construct the formula. They cluster non-pronouns by their heads and attaches a pronoun to the cluster which has no conflicting type, number, or gender, and contains the closest antecedent for the pronoun. Their system has four rules coded as a MLN:

$$InClust(m,c) \Rightarrow (Type(m,e) \Leftrightarrow Type(c,e)) \quad (1)$$

$$InClust(m,c) \Rightarrow (Number(m,e) \Leftrightarrow Number(c,e)) \quad (2)$$

$$InClust(m,c) \Rightarrow (Gender(m,e) \Leftrightarrow Gender(c,e)) \quad (3)$$

$$\neg IsPrn(m1) \wedge \neg IsPrn(m2) \wedge Head(m1,h1) \wedge Head(m2,h2) \wedge InClust(m1,c1) \wedge InClust(m2,c2) \Rightarrow (c1 = c2 \Leftrightarrow h1 = h2) \quad (4)$$

m is a mention, c is a cluster, e is a kind of type, number or gender, and h is a head. Weights of rule 1, 2 and 3 are infinite and must be satisfied, and weight of rule 4 is 100 in their system. They use MLN to do the unsupervised inference on generating clusters for the same entities, i.e. $InClust(m,c)$.

Our Improvement

To generate these predicates on entity type, number and gender, they use simple linguistic cues to infer and we believe they do this by hand. We only find Named Entity Recognizer (Stanford NER)(Finkel and Manning 2005) to generate type automatically. The head is generated by Stanford Parser(Klein and Manning 2003) and we will do the same way. They also mentioned that they use apposition and predicate nominatives to help coreference resolution. Again they detect them using simple heuristics based on parses by hand.

Location information of mentions may include some useful information and here we investigate neighbors. For example in "Merrill CEO John Thain" and "John A. Thain, its chief executive,", there are coreferent pairs of "Merrill CEO" and "John Thain", "John A. Thain" and "its chief executive". But sometimes neighbors separated with a comma don't mean the same entity, such as "I want an apple, a banana and a pear.". So we only consider neighbors just separated by one space. We think that neighbors with no conflict in types are the same entity with a large probability. Using Stanford NER, only special mentions can be identified with types and many others such as "its chief executive" will not be tagged. So if the type of one of neighbors is not tagged, there is no known type confliction. Whether two mentions are neighbors (separated by only one space) is easily obtained automatically.

Mentions with the same head may not mean the same thing, such as "Yahoo CEO" and "Merrill CEO John Thain". Our neighbor determination has a much higher accuracy than head information. So we try to use neighbor information to correct errors generated by the same head. If there is also "Yahoo CEO CarolBartz", we could use the relationship of "CarolBartz" and "John Thain" to examine the relationship between "Yahoo CEO" and "Merrill CEO". In our method, we can take a look at their coreference cluster probability value generated by MLN and do some correlations then.

We introduce some new rules to improve the performance:

(a) For neighbors (two mentions separated with only one space), if both two are the same type or one of them is untagged "Other", they are the same entity with a large probability. We also don't need to examine that mentions in one cluster must have the same head.

Now our improvement is shown below:

$$InClust(m,c) \wedge !(Type(m,Other)) \Rightarrow (Type(m,e) \Leftrightarrow Type(c,e)) \quad (5)$$

$$Neighbor(m1,m2) \wedge Type(m1,Other) \Rightarrow (InClust(m2,c) \Leftrightarrow InClust(m2,c)) \quad (6)$$

$$Neighbor(m1,m2) \wedge Type(m2,Other) \Rightarrow (InClust(m1,c) \Leftrightarrow InClust(m2,c)) \quad (7)$$

$$Neighbor(m1,m2) \wedge Type(m1,e1) \wedge Type(m2,e2) \wedge (e1 = e2) \Rightarrow (InClust(m1,c) \Leftrightarrow InClust(m2,c)) \quad (8)$$

$$\neg IsPrn(m1) \wedge \neg IsPrn(m2) \wedge Head(m1,h1) \wedge Head(m2,h2) \wedge InClust(m1,c1) \wedge InClust(m2,c2) \Rightarrow (h1 = h2 \Rightarrow c1 = c2) \quad (9)$$

Weights of rule 5, 6, 7 and 8 are infinite and must be satisfied, and weight of rule 9 is set as 100 in our system.

(b) Sometimes mentions with the same head are different entities. We find that neighbors we defined have a large probability to be a same entity than mentions with the same head. So we use this useful neighbor information to eliminate errors of the same head. Also mentions with tagged types (i.e. people) has a higher probability then mentions with un-tagged type in head determination. If two mentions are clustered by the same head, and one of them is typed "Other" and has a neighbor, we can use information whether its neighbor is in this cluster (its neighbor has the same head as or neighbors to at least one another mention in this cluster) to adjust our predication on whether they really clustered.

If $InClust(m1,c1) \wedge InClust(m2,c1) \wedge !(m1 = m2) \wedge Head(m1,h1) \wedge Head(m2,h2) \wedge (h1 = h2) \wedge Type(m1,Other) \wedge neighbor(m1,m3) \wedge !(m2 = m3)$, and for the coreference relationship is transitive, $m2$ and $m3$ must refer to the same entity. So $m3$ should also in the Cluster $c1$. In our method,the probability of a mention in a cluster, $P(m,c)$, can be output by the MLN inference. So $P(m3,c1)$ can be used to adjust our predication of $P(m1,c1)$). We will update $P(m1,c1)$:

$$P(m1,c1) = \frac{P(m1,c1) + P(m3,c1)}{2} \quad (10)$$

So if $m3$ has a low probability belonging to $c1$, $P(m1,c1)$ will be reduced by the probability of $P(m3,c1)$.

Experimental Analysis

Our dataset is a piece of news from New York Times, it includes 372 words. Stanford Parser gets distinct 60 mentions (the toppest level noun phrases of a sentence). In this text, there is a complex coreference example "John A. Thain, its chief executive and an alumnus of Goldman Sachs and the New York Stock Exchange,". Our ground truth was created by human experts using semantic knowledge and analysis to generate the real coreference. There are many coreferent mentions that can't be found by simple rules. We implemented our method on the Alchemy system(Kok and Lowd 2007). We evaluated it using MUC scoring program(Vilain and Hirschman 1995) with metrics: precision, recall and F1, and also false positive rate and false negative rate.

We use Stanford Parser to identify: (1) mentions (noun phrases); (2) whether a mention is a non-pronounce or pronounce; (3) neighbors (separated with only one space); (4) the head of a mention. We use Stanford NER to extract (some) named entities present in a noun phrase and classifies them into three different types: person, organization, location. Remaining un-tagged mentions are set as "Other".

We compared three algorithms: the original one, improvement (a), improvement (a)+(b). The result is shown below:

Method	Prec	Rec	F1	FP	FN
MLN	23.8%	25%	24.4	0.90%	79.2%
Improve(a)	36%	37.5%	36.7	0.90%	66.7%
Improve(a)+(b)	37.5%	37.5%	37.5	0.85%	66.7%

Table 1: Comparison of coreference results in MCU scores and false positive rate(FP) and false negative rate(FN)

From the result, we can see that: improvement (a) using the neighbor information can get more real coreferent mentions and improvement (b) eliminates some same-head errors increasing false positive rate without hurting the predication accuracy. The result of us and (Poon and Domingos 2008) are different because their ground truth are assumed that non-pronouns are clustered by heads while our ground truth are real coreference clusters including many conreferent mentions which can't be discovered by simple rules.

Conclusion

This paper introduces an efficient coreference resolution method using MLN. Experiments show that our implementation needs little human effort and new rules achieve a high coreference resolution accuracy.

References

Clark, J., and Gonzalez, J. 2008. Coreference resolution: Current trends and future directions. Technical report, The Language Techonologies Institute, CMU.

Denis, P., and Baldridge, J. 2007. Joint determination of anaphoricity and coreference resolution using integer programming. In *Proceedings of Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*.

Finkel, J. R.; Grenager, T., and Manning, C. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics*.

Klein, D., and Manning, C. D. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*.

Kok, S.; Singla, P. R. M. D. P. S. M. P. H., and Lowd, D. 2007. The alchemy system for statistical relational ai. Technical report, http:alchemy.cs.washington.edu.

McCallum, A., and Wellner, B. 2002. Conditional models of identity uncertainty with application to noun coreference. In *Proceedings of the Conference on Neural Information Processing Systems*.

Ng, V. 2008. Unsupervised models for coreference resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Poon, H., and Domingos, P. 2008. Joint unsupervised coreference resolution with markov logic. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Richardson, M., and Domingos, P. 2006. Markov logic networks. *Machine Learning*.

Vilain, M.; Burger, J. A. J. C. D., and Hirschman, L. 1995. A model-theoretic coreference scoring schema. In *Proceedings of the 6th Message Understanding Coference*.