Enforcing Transitivity in Coreference Resolution

Jenny Rose Finkel and Christopher D. Manning
Department of Computer Science
Stanford University
Stanford, CA 94305
{jrfinkel|manning}@cs.stanford.edu

Abstract

A desirable quality of a coreference resolution system is the ability to handle transitivity constraints, such that even if it places high likelihood on a particular mention being coreferent with each of two other mentions, it will also consider the likelihood of those two mentions being coreferent when making a final assignment. This is exactly the kind of constraint that integer linear programming (ILP) is ideal for, but, surprisingly, previous work applying ILP to coreference resolution has not encoded this type of constraint. We train a coreference classifier over pairs of mentions, and show how to encode this type of constraint on top of the probabilities output from our pairwise classifier to extract the most probable legal entity assignments. We present results on two commonly used datasets which show that enforcement of transitive closure consistently improves performance, including improvements of up to 3.6% using the b^3 scorer, and up to 16.5% using cluster f-measure.

1 Introduction

Much recent work on coreference resolution, which is the task of deciding which noun phrases, or *mentions*, in a document refer to the same real world entity, builds on Soon et al. (2001). They built a decision tree classifier to label pairs of mentions as coreferent or not. Using their classifier, they would build up *coreference chains*, where each mention was linked up with the most recent previous mention that the classifier labeled as coreferent, if such a mention existed. Transitive closure in this model was done implicitly. If *John Smith* was labeled coreferent with *Smith*, and *Smith* with *Jane Smith*, then *John Smith* and *Jane Smith* were also coreferent regardless of the classifier's evaluation of that pair. Much work that followed improved upon this strategy, by improving the features (Ng and Cardie, 2002b), the type of classifier (Denis and Baldridge, 2007), and changing mention links to be to the most likely antecedent rather than the most recent positively labeled antecedent (Ng and Cardie, 2002b). This line of work has largely ignored the implicit transitivity of the decisions made, and can result in unintuitive chains such as the *Smith* chain just described, where each pairwise decision is sensible, but the final result is not.

Ng and Cardie (2002a) and Ng (2004) highlight the problem of determining whether or not common noun phrases are anaphoric. They use two classifiers, an anaphoricity classifier, which decides if a mention should have an antecedent and a pairwise classifier similar those just discussed, which are combined in a cascaded manner. More recently, Denis and Baldridge (2007) utilized an integer linear programming (ILP) solver to better combine the decisions made by these two complementary classifiers, by finding the globally optimal solution according to both classifiers. However, when encoding constraints into their ILP solver, they did not enforce transitivity.

The goal of the present work is simply to show that transitivity constraints are a useful source of information, which can and should be incorporated into an ILP-based coreference system. For this goal, we put aside the anaphoricity classifier and focus on the pairwise classifier and transitivity constraints. We build a pairwise logistic classifier, trained on all pairs of mentions, and then at test time we use an ILP solver equipped with transitivity constraints to find the most likely legal assignment to the variables which represent the pairwise decisions.[1] Our results show a significant improvement compared to the naïve use of the pairwise classifier.

Other work on global models of coreference (as

[1] A *legal assignment* is one which respects transitive closure.

Proceedings of ACL-08: HLT, Short Papers (Companion Volume), pages 45–48,
Columbus, Ohio, USA, June 2008.

opposed to pairwise models) has included: Luo et al. (2004) who used a Bell tree whose leaves represent possible partitionings of the mentions into entities and then trained a model for searching the tree; McCallum and Wellner (2004) who defined several conditional random field-based models; Ng (2005) who took a reranking approach; and Culotta et al. (2006) who use a probabilistic first-order logic model.

2 Coreference Resolution

For this task we are given a document which is annotated with a set of mentions, and the goal is to cluster the mentions which refer to the same entity. When describing our model, we build upon the notation used by Denis and Baldridge (2007).

2.1 Pairwise Classification

Our baseline systems are based on a logistic classifier over pairs of mentions. The probability of a pair of mentions takes the standard logistic form:

$$P(x_{\langle i,j\rangle}|m_i, m_j; \theta) = \left(1 + e^{-f(m_i,m_j)\cdot\theta}\right)^{-1} \quad (1)$$

where m_i and m_j correspond to mentions i and j respectively; $f(m_i, m_j)$ is a feature function over a pair of mentions; θ are the feature weights we wish to learn; and $x_{\langle i,j\rangle}$ is a boolean variable which takes value 1 if m_i and m_j are coreferent, and 0 if they are not. The log likelihood of a document is the sum of the log likelihoods of all pairs of mentions:

$$\mathcal{L}(\mathbf{x}|\mathbf{m};\theta) = \sum_{m_i,m_j\in\mathbf{m}^2} \log P(x_{\langle i,j\rangle}|m_i, m_j; \theta) \quad (2)$$

where $\mathbf{m}$ is the set of mentions in the document, and $\mathbf{x}$ is the set of variables representing each pairwise coreference decision $x_{\langle i,j\rangle}$. Note that this model is degenerate, because it assigns probability mass to nonsensical clusterings. Specifically, it will allow $x_{\langle i,j\rangle} = x_{\langle j,k\rangle} = 1$ while $x_{\langle i,k\rangle} = 0$.

Prior work (Soon et al., 2001; Denis and Baldridge, 2007) has generated training data for pairwise classifiers in the following manner. For each mention, work backwards through the preceding mentions in the document until you come to a true coreferent mention. Create negative examples for all intermediate mentions, and a positive example for the mention and its correct antecedent. This approach made sense for Soon et al. (2001) because testing proceeded in a similar manner: for each mention, work backwards until you find a previous mention which the classifier thinks is coreferent, add a link, and terminate the search. The COREF-ILP model of Denis and Baldridge (2007) took a different approach at test time: for each mention they would work backwards and add a link for *all* previous mentions which the classifier deemed coreferent. This is equivalent to finding the most likely assignment to each $x_{\langle i,j\rangle}$ in Equation 2. As noted, these assignments may not be a legal clustering because there is no guarantee of transitivity. The transitive closure happens in an ad-hoc manner after this assignment is found: any two mentions linked through other mentions are determined to be coreferent. Our SOON-STYLE baseline used the same training and testing regimen as Soon et al. (2001). Our D&B-STYLE baseline used the same test time method as Denis and Baldridge (2007), however at training time we created data for all mention pairs.

2.2 Integer Linear Programming to Enforce Transitivity

Because of the ad-hoc manner in which transitivity is enforced in our baseline systems, we do not necessarily find the most probable legal clustering. This is exactly the kind of task at which integer linear programming excels. We need to first formulate the objective function which we wish the ILP solver to maximize at test time.[2] Let $p_{\langle i,j\rangle} = \log P(x_{\langle i,j\rangle}|m_i, m_j; \theta)$, which is the log probability that m_i and m_j are coreferent according to the pairwise logistic classifier discussed in the previous section, and let $\bar{p}_{\langle i,j\rangle} = \log(1 - p_{\langle i,j\rangle})$, be the log probability that they are not coreferent. Our objective function is then the log probability of a particular (possibly illegal) variable assignment:

$$\max \sum_{m_i,m_j\in\mathbf{m}^2} p_{\langle i,j\rangle} \cdot x_{\langle i,j\rangle} - \bar{p}_{\langle i,j\rangle} \cdot (1 - x_{\langle i,j\rangle}) \quad (3)$$

We add binary constraints on each of the variables: $x_{\langle i,j\rangle} \in \{0, 1\}$. We also add constraints, over each triple of mentions, to enforce transitivity:

$$(1 - x_{\langle i,j\rangle}) + (1 - x_{\langle j,k\rangle}) \geq (1 - x_{\langle i,k\rangle}) \quad (4)$$

[2] Note that there are no changes from the D&B-STYLE baseline system at training time.

This constraint ensures that whenever $x_{\langle i,j \rangle} = x_{\langle j,k \rangle} = 1$ it must also be the case that $x_{\langle i,k \rangle} = 1$.

3 Experiments

We used *lp_solve*[3] to solve our ILP optimization problems. We ran experiments on two datasets. We used the MUC-6 formal training and test data, as well as the NWIRE and BNEWS portions of the ACE (Phase 2) corpus. This corpus had a third portion, NPAPER, but we found that several documents where too long for *lp_solve* to find a solution.[4]

We added named entity (NE) tags to the data using the tagger of Finkel et al. (2005). The ACE data is already annotated with NE tags, so when they conflicted they overrode the tags output by the tagger. We also added part of speech (POS) tags to the data using the tagger of Toutanova et al. (2003), and used the tags to decide if mentions were plural or singular. The ACE data is labeled with mention type (*pronominal*, *nominal*, and *name*), but the MUC-6 data is not, so the POS and NE tags were used to infer this information. Our feature set was simple, and included many features from (Soon et al., 2001), including the pronoun, string match, definite and demonstrative NP, number and gender agreement, proper name and appositive features. We had additional features for NE tags, head matching and head substring matching.

3.1 Evaluation Metrics

The MUC scorer (Vilain et al., 1995) is a popular coreference evaluation metric, but we found it to be fatally flawed. As observed by Luo et al. (2004), if all mentions in each document are placed into a single entity, the results on the MUC-6 formal test set are 100% recall, 78.9% precision, and 88.2% F1 score – significantly higher than any published system. The b^3 scorer (Amit and Baldwin, 1998) was proposed to overcome several shortcomings of the MUC scorer. However, coreference resolution is a clustering task, and many cluster scorers already exist. In addition to the MUC and b^3 scorers, we also evaluate using cluster f-measure (Ghosh, 2003), which is the standard f-measure computed over true/false coreference decisions for pairs of mentions; the Rand index (Rand, 1971), which is pairwise accuracy of the clustering; and variation of information (Meila, 2003), which utilizes the entropy of the clusterings and their mutual information (and for which lower values are better).

3.2 Results

Our results are summarized in Table 1. We show performance for both baseline classifiers, as well as our ILP-based classifier, which finds the most probable legal assignment to the variables representing coreference decisions over pairs of mentions. For comparison, we also give the results of the COREF-ILP system of Denis and Baldridge (2007), which was also based on a naïve pairwise classifier. They used an ILP solver to find an assignment for the variables, but as they note at the end of Section 5.1, it is equivalent to taking all links for which the classifier returns a probability ≥ 0.5, and so the ILP solver is not really necessary. We also include their JOINT-ILP numbers, however that system makes use of an additional anaphoricity classifier.

For all three corpora, the ILP model beat both baselines for the cluster f-score, Rand index, and variation of information metrics. Using the b^3 metric, the ILP system and the D&B-STYLE baseline performed about the same on the MUC-6 corpus, though for both ACE corpora, the ILP system was the clear winner. When using the MUC scorer, the ILP system always did worse than the D&B-STYLE baseline. However, this is precisely because the transitivity constraints tend to yield smaller clusters (which increase precision while decreasing recall). Remember that going in the opposite direction and simply putting *all* mentions in one cluster produces a MUC score which is higher than any in the table, even though this clustering is clearly not useful in applications. Hence, we are skeptical of this measure's utility and provide it primarily for comparison with previous work. The improvements from the ILP system are most clearly shown on the ACE NWIRE corpus, where the b^3 f-score improved 3.6%, and the cluster f-score improved 16.5%.

4 Conclusion

We showed how to use integer linear programming to encode transitivity constraints in a corefer-

[3] From `http://lpsolve.sourceforge.net/`

[4] Integer linear programming is, after all, NP-hard.

	MUC Scorer			b^3 Scorer			Cluster				
Model	P	R	F1	P	R	F1	P	R	F1	RAND	VOI
MUC-6											
D&B-style baseline	84.8	59.4	69.9	79.7	54.4	64.6	43.8	44.4	44.1	89.9	1.78
Soon-style baseline	91.5	51.5	65.9	94.4	46.7	62.5	88.2	31.9	46.9	93.5	1.65
ILP	89.7	55.1	68.3	90.9	49.7	64.3	74.1	37.1	49.5	93.2	1.65
ACE − NWIRE											
D&B coref-ILP	74.8	60.1	66.8		–			–		–	–
D&B joint-ILP	75.8	60.8	67.5		–			–		–	–
D&B-style baseline	73.3	67.6	70.4	70.1	71.4	70.8	31.1	54.0	39.4	91.7	1.42
Soon-style baseline	85.3	37.8	52.4	94.1	56.9	70.9	67.7	19.8	30.6	95.5	1.38
ILP	78.7	58.5	67.1	86.8	65.2	74.5	76.1	44.2	55.9	96.5	1.09
ACE − BNEWS											
D&B coref-ILP	75.5	62.2	68.2		–			–		–	–
D&B joint-ILP	78.0	62.1	69.2		–			–		–	–
D&B-style baseline	77.9	51.1	61.7	80.3	64.2	71.4	35.5	33.8	34.6	0.89	1.32
Soon-style baseline	90.0	43.2	58.3	95.6	58.4	72.5	83.3	21.5	34.1	0.93	1.09
ILP	87.8	46.8	61.1	93.5	59.9	73.1	77.5	26.1	39.1	0.93	1.06

Table 1: Results on all three datasets with all five scoring metrics. For VOI a lower number is better.

ence classifier which models pairwise decisions over mentions. We also demonstrated that enforcing such constraints at test time can significantly improve performance, using a variety of evaluation metrics.

Acknowledgments

Thanks to the following members of the Stanford NLP reading group for helpful discussion: Sharon Goldwater, Michel Galley, Anna Rafferty.

This paper is based on work funded by the Disruptive Technology Office (DTO) Phase III Program for Advanced Question Answering for Intelligence (AQUAINT).

References

B. Amit and B. Baldwin. 1998. Algorithms for scoring coreference chains. In *MUC7*.

A. Culotta, M. Wick, and A. McCallum. 2006. First-order probabilistic models for coreference resolution. In *NAACL*.

P. Denis and J. Baldridge. 2007. Joint determination of anaphoricity and coreference resolution using integer programming. In *HLT-NAACL*, Rochester, New York.

J. Finkel, T. Grenager, and C. Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *ACL*.

J. Ghosh. 2003. Scalable clustering methods for data mining. In N. Ye, editor, *Handbook of Data Mining*, chapter 10, pages 247–277. Lawrence Erlbaum Assoc.

X. Luo, A. Ittycheriah, H. Jing, N. Kambhatla, and S. Roukos. 2004. A mention-synchronous coreference resolution algorithm based on the Bell tree. In *ACL*.

A. McCallum and B. Wellner. 2004. Conditional models of identity uncertainty with application to noun coreference. In *NIPS*.

M. Meila. 2003. Comparing clusterings by the variation of information. In *COLT*.

V. Ng and C. Cardie. 2002a. Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution. In *COLING*.

V. Ng and C. Cardie. 2002b. Improving machine learning approaches to coreference resolution. In *ACL*.

V. Ng. 2004. Learning noun phrase anaphoricity to improve coreference resolution: issues in representation and optimization. In *ACL*.

V. Ng. 2005. Machine learning for coreference resolution: From local classification to global ranking. In *ACL*.

W. M. Rand. 1971. Objective criteria for the evaluation of clustering methods. In *Journal of the American Statistical Association, 66*, pages 846–850.

W. Soon, H. Ng, and D. Lim. 2001. A machine learning approach to coreference resolution of noun phrases. In *Computational Linguistics, 27(4)*.

K. Toutanova, D. Klein, and C. Manning. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *HLT-NAACL 2003*.

M. Vilain, J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman. 1995. A model-theoretic coreference scoring scheme. In *MUC6*.