The Same-head Heuristic for Coreference

Micha Elsner and Eugene Charniak
Brown Laboratory for Linguistic Information Processing (BLLIP)
Brown University
Providence, RI 02912
{melsner,ec}@cs.brown.edu

Abstract

We investigate coreference relationships between NPs with the same head noun. It is relatively common in unsupervised work to assume that such pairs are coreferent– but this is not always true, especially if realistic mention detection is used. We describe the distribution of non-coreferent same-head pairs in news text, and present an unsupervised generative model which learns not to link some same-head NPs using syntactic features, improving precision.

1 Introduction

Full NP coreference, the task of discovering which non-pronominal NPs in a discourse refer to the same entity, is widely known to be challenging. In practice, however, most work focuses on the subtask of linking NPs with different head words. Decisions involving NPs with the same head word have not attracted nearly as much attention, and many systems, especially unsupervised ones, operate under the assumption that all same-head pairs corefer. This is by no means always the case– there are several systematic exceptions to the rule. In this paper, we show that these exceptions are fairly common, and describe an unsupervised system which learns to distinguish them from coreferent same-head pairs.

There are several reasons why relatively little attention has been paid to same-head pairs. Primarily, this is because they are a comparatively easy subtask in a notoriously difficult area; Stoyanov et al. (2009) shows that, among NPs headed by common nouns, those which have an exact match earlier in the document are the easiest to resolve (variant MUC score .82 on MUC-6) and while those with partial matches are quite a bit harder (.53), by far the worst performance is on those without any match at all (.27). This effect is magnified by most popular metrics for coreference, which reward finding links within large clusters more than they punish proposing spurious links, making it hard to improve performance by linking conservatively. Systems that use gold mention boundaries (the locations of NPs marked by annotators)[1] have even less need to worry about same-head relationships, since most NPs which disobey the conventional assumption are not marked as mentions.

In this paper, we count how often same-head pairs fail to corefer in the MUC-6 corpus, showing that gold mention detection hides most such pairs, but more realistic detection finds large numbers. We also present an unsupervised generative model which learns to make certain same-head pairs non-coreferent. The model is based on the idea that pronoun referents are likely to be salient noun phrases in the discourse, so we can learn about NP antecedents using pronominal antecedents as a starting point. Pronoun anaphora, in turn, is learnable from raw data (Cherry and Bergsma, 2005; Charniak and Elsner, 2009). Since our model links fewer NPs than the baseline, it improves precision but decreases recall. This tradeoff is favorable for CEAF, but not for b^3.

2 Related work

Unsupervised systems specify the assumption of same-head coreference in several ways: by as-

[1] Gold mention detection means something slightly different in the ACE corpus, where the system input contains every NP annotated with an entity type.

sumption (Haghighi and Klein, 2009), using a head-prediction clause (Poon and Domingos, 2008), and using a sparse Dirichlet prior on word emissions (Haghighi and Klein, 2007). (These three systems, perhaps not coincidentally, use gold mentions.) An exception is Ng (2008), who points out that head identity is not an entirely reliable cue and instead uses exact string match (minus determiners) for common NPs and an alias detection system for proper NPs. This work uses mentions extracted with an NP chunker. No specific results are reported for same-head NPs. However, while using exact string match raises precision, many non-matching phrases are still coreferent, so this approach cannot be considered a full solution to the problem.

Supervised systems do better on the task, but not perfectly. Recent work (Stoyanov et al., 2009) attempts to determine the contributions of various categories of NP to coreference scores, and shows (as stated above) that common NPs which partially match an earlier mention are not well resolved by the state-of-the-art RECONCILE system, which uses pairwise classification. They also show that using gold mention boundaries makes the coreference task substantially easier, and argue that this experimental setting is "rather unrealistic".

3 Descriptive study: MUC-6

We begin by examining how often non-same-head pairs appear in the MUC-6 coreference dataset. To do so, we compare two artificial coreference systems: the **link-all** strategy links all, and only, full (non-pronominal) NP pairs with the same head which occur within 10 sentences of one another. The **oracle** strategy links NP pairs with the same head which occur within 10 sentences, but only if they are actually coreferent (according to the gold annotation)[2] The link-all system, in other words, does what most existing unsupervised systems do on the same-head subset of NPs, while the oracle system performs perfectly.

We compare our results to the gold standard using two metrics. b^3(Bagga and Baldwin, 1998) is a standard metric which calculates a precision and recall for each mention. The mention CEAF (Luo, 2005) constructs a maximum-weight bipartite matching between gold and proposed clusters, then gives the percentage of entities whose gold label and proposed label match. b^3 gives more weight to errors involving larger clusters (since these lower scores for several mentions at once); for mention CEAF, all mentions are weighted equally.

We annotate the data with the self-trained Charniak parser (McClosky et al., 2006), then extract mentions using three different methods. The **gold mentions** method takes only mentions marked by annotators. The **nps** method takes all base noun phrases detected by the parser. Finally, the **nouns** method takes all nouns, even those that do not head NPs; this method maximizes recall, since it does not exclude prenominals in phrases like "a **Bush** spokesman". (High-precision models of the internal structure of flat Penn Treebank-style NPs were investigated by Vadas and Curran (2007).) For each experimental setting, we show the number of **mentions** detected, and how many of them are **linked** to some antecedent by the system.

The data is shown in Table 1. b^3 shows a large drop in precision when all same-head pairs are linked; in fact, in the *nps* and *nouns* settings, only about half the same-headed NPs are actually coreferent (864 real links, 1592 pairs for *nps*). This demonstrates that non-coreferent same-head pairs not only occur, but are actually rather common in the dataset. The drop in precision is much less obvious in the *gold mentions* setting, however; most unlinked same-head pairs are not annotated as mentions in the gold data, which is one reason why systems run in this experimental setting can afford to ignore them.

Improperly linking same-head pairs causes a loss in precision, but scores are dominated by recall[3]. Thus, reporting b^3 helps to mask the impact of these pairs when examining the final f-score.

We roughly characterize what sort of same-headed NPs are non-coreferent by hand-examining 100 randomly selected pairs. 39 pairs denoted different entities ("recent employees" vs "employees who have worked for longer") disambiguated by modifiers or sometimes by discourse position. The next largest group (24) consists of time and measure phrases like "ten miles". 12 pairs refer to parts or quantities

[2]The choice of 10 sentences as the window size captures most, but not all, of the available recall. Using *nouns* mention detection, it misses 117 possible same-head links, or about 10%. However, precision drops further as the window size increases.

[3]This bias is exaggerated for systems which only link same-head pairs, but continues to apply to real systems; for instance (Haghighi and Klein, 2009) has a b^3 precision of 84 and recall of 67.

	Mentions	Linked	b^3 pr	rec	F	mention CEAF
Gold mentions						
Oracle	1929	**1164**	100	32.3	**48.8**	**54.4**
Link all	1929	**1182**	80.6	31.7	**45.5**	**53.8**
Alignment	1929	495	93.7	22.1	35.8	40.5
NPs						
Oracle	3993	**864**	100	30.6	**46.9**	**73.4**
Link all	3993	**1592**	67.2	29.5	**41.0**	**62.2**
Alignment	3993	518	87.2	24.7	38.5	67.0
Nouns						
Oracle	5435	**1127**	100	41.5	**58.6**	**83.5**
Link all	5435	**2541**	56.6	40.9	**45.7**	**67.0**
Alignment	5435	935	83.0	32.8	47.1	74.4

Table 1: Oracle, system and baseline scores on MUC-6 test data. **Gold mentions leave little room for improvement** between baseline and oracle; **detecting more mentions widens the gap between them**. With realistic mention detection, precision and CEAF scores improve over baselines, while recall and f-scores drop.

("members of..."), and 12 contained a generic ("In a corporate campaign, a union tries..."). 9 contained an annotator error. The remaining 4 were mistakes involving proper noun phrases headed by *Inc.* and other abbreviations; this case is easy to handle, but apparently not the primary cause of errors.

4 System

Our system is a version of the popular IBM model 2 for machine translation. To define our generative model, we assume that the parse trees for the entire document D are given, *except* for the subtrees with root nonterminal NP, denoted n_i, which our system will generate. These subtrees are related by a hidden set of alignments, a_i, which link each NP to another NP (which we call a *generator*) appearing somewhere before it in the document, or to a null antecedent. The set of potential generators G (which plays the same role as the source-language text in MT) is taken to be all the NPs occurring within 10 sentences of the target, plus a special null antecedent which plays the same role as the null word in machine translation– it serves as a dummy generator for NPs which are unrelated to any real NP in G.

The generative process fills in all the NP nodes in order, from left to right. This process ensures that, when generating node n_i, we have already filled in all the NPs in the set G (since these all precede n_i). When deciding on a generator for NP n_i, we can extract features characterizing its relationship to a potential generator g_j. These features, which we denote $f(n_i, g_j, D)$, may depend on their relative position in the document D, and on any features of g_j, since we have already generated its tree. However, we cannot extract features from the subtree under n_i, since we have yet to generate it!

As usual for IBM models, we learn using EM, and we need to start our alignment function off with a good initial set of parameters. Since antecedents of NPs and pronouns (both salient NPs) often occur in similar syntactic environments, we use an alignment function for pronoun coreference as a starting point. This alignment can be learned from raw data, making our approach unsupervised.

We take the pronoun model of Charniak and Elsner (2009)[4] as our starting point. We re-express it in the IBM framework, using a log-linear model for our alignment. Then our alignment (parameterized by feature weights w) is:

$$p(a_i = j|G, D) \propto exp(f(n_i, g_j, D) \bullet w)$$

The weights w are learned by gradient descent on the log-likelihood. To use this model within EM, we alternate an E-step where we calculate the expected alignments $E[a_i = j]$, then an M-step where we run gradient descent. (We have also had some success with stepwise EM as in (Liang and Klein, 2009), but this requires some tuning to work properly.)

[4]Downloaded from http://bllip.cs.brown.edu.

As features, we take the same features as Charniak and Elsner (2009): sentence and word-count distance between n_i and g_j, sentence position of each, syntactic role of each, and head type of g_j (proper, common or pronoun). We add binary features for the nonterminal directly over g_j (NP, VP, PP, any S type, or other), the type of phrases modifying g_j (proper nouns, phrasals (except QP and PP), QP, PP-of, PP-other, other modifiers, or nothing), and the type of determiner of g_j (possessive, definite, indefinite, deictic, other, or nothing). We designed this feature set to distinguish prominent NPs in the discourse, and also to be able to detect abstract or partitive phrases by examining modifiers and determiners.

To produce full NPs and learn same-head coreference, we focus on learning a good alignment using the pronoun model as a starting point. For translation, we use a trivial model, $p(n_i|g_{a_i}) = 1$ if the two have the same head, and 0 otherwise, except for the null antecedent, which draws heads from a multinomial distribution over words.

While we could learn an alignment and then treat all generators as antecedents, so that only NPs aligned to the null antecedent were not labeled coreferent, in practice this model would align nearly all the same-head pairs. This is true because many words are "bursty"; the probability of a second occurrence given the first is higher than the a priori probability of occurrence (Church, 2000). Therefore, our model is actually a mixture of two IBM models, p_C and p_N, where p_C produces NPs with antecedents and p_N produces pairs that share a head, but are not coreferent. To break the symmetry, we allow p_C to use any parameters w, while p_N uses a uniform alignment, $w \equiv \vec{0}$. We interpolate between these two models with a constant λ, the single manually set parameter of our system, which we fixed at .9.

The full model, therefore, is:

$$\begin{aligned} p(n_i|G,D) =& \lambda p_T(n_i|G,D) \\ &+ (1-\lambda) p_N(n_i|G,D) \\ p_T(n_i|G,D) =& \frac{1}{Z} \sum_{j \in G} exp(f(n_i, g_j, D) \bullet w) \\ &\times \mathbb{I}\{head(n_i) = head(j)\} \\ p_T(n_i|G,D) =& \sum_{j \in G} \frac{1}{|G|} \mathbb{I}\{head(n_i) = head(g_j)\} \end{aligned}$$

NPs for which the maximum-likelihood generator (the largest term in either of the sums) is from p_T and is not the null antecedent are marked as coreferent to the generator. Other NPs are marked not coreferent.

5 Results

Our results on the MUC-6 formal test set are shown in Table 1. In all experimental settings, the model improves precision over the baseline while decreasing recall– that is, it misses some legitimate coreferent pairs while correctly excluding many of the spurious ones. Because of the precision-recall tradeoff at which the systems operate, this results in reduced b^3 and link F. However, for the *nps* and *nouns* settings, where the parser is responsible for finding mentions, the tradeoff is positive for the CEAF metrics. For instance, in the *nps* setting, it improves over baseline by 57%.

As expected, the model does poorly in the gold mentions setting, doing worse than baseline on both metrics. Although it is possible to get very high precision in this setting, the model is far too conservative, linking less than half of the available mentions to anything, when in fact about 60% of them are coreferent. As we explain above, this experimental setting makes it mostly unnecessary to worry about non-coreferent same-head pairs because the MUC-6 annotators don't often mark them.

6 Conclusions

While same-head pairs are easier to resolve than same-other pairs, they are still non-trivial and deserve further attention in coreference research. To effectively measure their effect on performance, researchers should report multiple metrics, since under b^3 the link-all heuristic is extremely difficult to beat. It is also important to report results using a realistic mention detector as well as gold mentions.

Acknowledgements

We thank Jean Carletta for the SWITCHBOARD annotations, and Dan Jurafsky and eight anonymous reviewers for their comments and suggestions. This work was funded by a Google graduate fellowship.

References

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *LREC Workshop on Linguistics Coreference*, pages 563–566.

Eugene Charniak and Micha Elsner. 2009. EM works for pronoun anaphora resolution. In *Proceedings of EACL*, Athens, Greece.

Colin Cherry and Shane Bergsma. 2005. An Expectation Maximization approach to pronoun resolution. In *Proceedings of CoNLL*, pages 88–95, Ann Arbor, Michigan.

Kenneth W. Church. 2000. Empirical estimates of adaptation: the chance of two Noriegas is closer to p/2 than p^2. In *Proceedings of ACL*, pages 180–186.

Aria Haghighi and Dan Klein. 2007. Unsupervised coreference resolution in a nonparametric Bayesian model. In *Proceedings of ACL*, pages 848–855.

Aria Haghighi and Dan Klein. 2009. Simple coreference resolution with rich syntactic and semantic features. In *Proceedings of EMNLP*, pages 1152–1161.

Percy Liang and Dan Klein. 2009. Online EM for unsupervised models. In *HLT-NAACL*.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of HLT-EMNLP*, pages 25–32, Morristown, NJ, USA. Association for Computational Linguistics.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Proceedings of HLT-NAACL*, pages 152–159.

Vincent Ng. 2008. Unsupervised models for coreference resolution. In *Proceedings of EMNLP*, pages 640–649, Honolulu, Hawaii. Association for Computational Linguistics.

Hoifung Poon and Pedro Domingos. 2008. Joint unsupervised coreference resolution with Markov Logic. In *Proceedings of EMNLP*, pages 650–659, Honolulu, Hawaii, October. Association for Computational Linguistics.

Veselin Stoyanov, Nathan Gilbert, Claire Cardie, and Ellen Riloff. 2009. Conundrums in noun phrase coreference resolution: Making sense of the state-of-the-art. In *Proceedings of ACL-IJCNLP*, pages 656–664, Suntec, Singapore, August. Association for Computational Linguistics.

David Vadas and James Curran. 2007. Adding noun phrase structure to the penn treebank. In *Proceedings of ACL*, pages 240–247, Prague, Czech Republic, June. Association for Computational Linguistics.