Active Learning for Coreference Resolution

Florian Laws[1] **Florian Heimerl**[2] **Hinrich Schütze**[1]
[1] Institute for Natural Language Processing (IMS)
Universität Stuttgart
florian.laws@ims.uni-stuttgart.de
[2] Institute for Visualization and Interactive Systems
Universität Stuttgart
florian.heimerl@vis.uni-stuttgart.de

Abstract

We present an active learning method for coreference resolution that is novel in three respects. (i) It uses bootstrapped neighborhood pooling, which ensures a class-balanced pool even though gold labels are not available. (ii) It employs neighborhood selection, a selection strategy that ensures coverage of both positive and negative links for selected markables. (iii) It is based on a query-by-committee selection strategy in contrast to earlier uncertainty sampling work. Experiments show that this new method outperforms random sampling in terms of both annotation effort and peak performance.

1 Introduction

Coreference resolution (CR) – the task of determining if two expressions in natural language text refer to the same real-world entity – is an important NLP task. One popular approach to CR is supervised classification. This approach needs manually labeled training data that is expensive to create. Active learning (AL) is a technique that can reduce this cost by setting up an interactive training/annotation loop that selects and annotates training examples that are maximally useful for the classifier that is being trained. However, while AL has been proven successful for many other NLP tasks, such as part-of-speech tagging (Ringger et al., 2007), parsing (Osborne and Baldridge, 2004), text classification (Tong and Koller, 2002) and named entity recognition (Tomanek et al., 2007), AL has not been successfully applied to coreference resolution so far.

In this paper, we present a novel approach to AL for CR based on query-by-committee sampling and bootstrapping and show that it performs better than a number of baselines.

2 Related work

Coreference resolution. The perhaps most widely used supervised learning approach to CR is the mention-pair model (Soon et al., 2001). This model classifies links (pairs of two mentions) as coreferent or disreferent, followed by a clustering stage that partitions entities based on the link decisions. Our AL method is partially based on the class balancing strategy proposed by Soon et al. (2001).

While models other than mention-pair have been proposed (Culotta et al., 2007), none performs clearly better as evidenced by recent shared evaluations such as SemEval 2010 (Recasens et al., 2010) and CoNLL 2011 (Pradhan et al., 2011).

Active learning. The only existing publication on AL for CR that we are aware of is (Gasperin, 2009). She uses a mention-pair model on a biomedical corpus. The classifier is Naive Bayes and the AL method uncertainty sampling (Lewis and Gale, 1994). The results are negative: AL is not better than random sampling. In preliminary experiments, we replicated this result for our corpus and our system: Uncertainty sampling is not better than random sampling for CR. Uncertainty sampling can fail if uncertainty assessments are too unstable for successful example selection (cf. Dwyer and Holte (2007)). This seems to be the case for the decision trees we use. Naive Bayes is also known to give bad uncertainty assessments (Domingos and Pazzani,

2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 508–512,
Montréal, Canada, June 3-8, 2012.

1997). We therefore adopted a query-by-committee approach combined with a class-balancing strategy.

3 Active learning for CR

The classifier in the mention-pair model is faced with a severe class imbalance: there are many more disreferent than coreferent links. To address this imbalance, we use a *neighborhood pool* or N-pool as proposed by Soon et al. (2001).

Generation of the N-pool. The neighborhood of markable x used in N-pooling is defined as the set consisting of the link between x and its closest coreferent markable $y(x)$ to the left and all disreferent links in between. For a particular markable x, let $y(x)$ be the closest coreferent markable for x to the left of x. Between $y(x)$ and x, there are disreferent markables z_i, so we have a constellation like $y(x), z_1, \ldots, z_n, x$. The neighborhood of x is then the set of links

$$\{(y, x), (z_1, x) \ldots, (z_n, x)\}$$

This set is empty if x does not have a coreferent markable to the left.

We call the set of all such neighborhoods the N-pool. The N-pool is a subset of the entire pool of links.

Bootstrapping the neighborhood. Soon et al. (2001) introduce N-pooling for labeled data. In AL, no labeled data (or very little of it) is available. Instead, we employ the committee of classifiers that we use for AL example selection for bootstrapping the N-pool. We query the committee of classifiers from the last AL iteration and treat a link as coreferent if and only if the majority of the classifiers classifies it as coreferent. We then construct the N-pool using these bootstrapped labels to determine the coreferent markables $y(x)$ and then construct the neighborhoods as described above.

If this procedure yields no coreferent links in an iteration, we sample links left of randomly selected markables instead of N-pooling.

Example selection granularity. We use a query-by-committee approach to AL. The committee consists of 10 instances of the link classifier of the CR system, each trained on a randomly chosen subset of the links that have been manually labeled so far.

In each iteration, the N-pool is recomputed and a small subset of the N-pool is selected for labeling. We experiment with two selection granularities. In *neighborhood selection*, entire neighborhoods are selected and labeled in each iteration. We define the utility of a neighborhood as the average of the vote entropies (Argamon-Engelson and Dagan, 1999) of its links.

In *link selection*, individual links with the highest utility are selected – in most cases these will be from different neighborhoods. Utility is again defined as vote entropy.

Our hypothesis is that, compared to selection of individual links, neighborhood selection yields a more balanced sample that covers both positive and negative links for a markable. At the same time, neighborhood selection retains the benefits of AL sampling: difficult (or highly informative) links are selected.

4 Experiments

We use the mention-pair CR system SUCRE (Kobdani et al., 2011). The link classifier is a decision tree and the clustering algorithm a variant of best-first clustering (Ng and Cardie, 2002). SUCRE results were competitive in SEMEVAL 2010 (Recasens et al., 2010). We implemented N-pool bootstrapping and selection methods on top of the AL framework of Tomanek et al. (2007).

We use the English part of the SemEval-2010 CR task data set, a subset of OntoNotes 2.0 (Hovy et al., 2006). Training and test set sizes are about 96,000 and 24,000 words. Since we focus on the coreference resolution subtask, we use the true mention boundaries for the markables.

The pool for example selection is created by pairing every markable with every preceding markable within a window of 100 markables. This yields a pool of 1.7 million links, of which only 1.5% are labeled as coreferent. This drastic class imbalance necessitates our bootstrapped class-balancing.

We run two baseline experiments for comparison: (i) random selection on the entire pool, without any class balancing, and (ii) random selection from a gold-label-based N-pool. We chose to use gold neighborhood information for the baseline to remove the influence of badly predicted neighbor-

			20,000 links				50,000 links			
			MUC	B3	CEAF	mean	MUC	B3	CEAF	mean
(1)	random	entire pool	49.68	86.07	82.34	72.70	48.81	86.00	82.24	72.34
(2)		N-pooling	61.60	85.00	82.85	76.48	62.60	85.99	83.44	77.33
(3)	AL	link selection	55.65	86.91†	83.67†	75.41	55.84	86.94†	83.70	75.49
(4)		neighborhood sel.	63.07†	86.94†	84.42†	78.14†	63.81†	87.11†	84.33†	78.42†

Table 1: Performance of different methods. All measures are F_1 measures.

hoods and focus on the performance of random sampling. Hence, this is a very strong random baseline. The performance with bootstrapped neighborhoods would likely be lower.

We run 10 runs of each experiment, starting from 10 different seed sets. These seed sets contained 200 links, drawn randomly from the entire pool, for random sampling; and 20 neighborhoods for neighborhood selection, with a comparable number of links. We verified that each seed set contained instances of both classes.

5 Results

We determine the performance of CR depending on the number of links used for training. The results of the experiments are shown in Table 1 and Figures 1a to 1d. We show results for four coreference measures: MUC, B3, entity-based CEAF (henceforth: CEAF), and the arithmetic mean of MUC, B3 and CEAF (as suggested by the CoNLL-2011 shared evaluation).

In all four figures, the AL curves have reached a plateau at 20,000 links. At this point, neighborhood selection AL (line 4 in Table 1) outperforms random sampling from the N-pool (line 2) for all coreference measures, with gains from 1.47 points for MUC to 1.94 points for B3.

At 20,000 links, the N-pooling random baseline (line 2) has not yet reached maximum performance, but even at 50,000 links, neighborhood selection AL still outperforms the baselines. (AL and baseline performance will eventually converge when most links from the pool are sampled, but this will happen much later, since the pool has 1.7 million links in total).

Link selection AL (line 3) outperforms the baselines for B3 and CEAF, but is performing markedly worse than the N-pooling random baseline (line 2) for MUC (due to low recall for MUC) and mean F_1. Link selection yields a CR system that proposes a lot of singleton entities that are not coreferent with any other entity. The MUC scoring scheme does not give credit to singletons at all, thus the lower recall.

Neighborhood selection AL initially has low MUC, but starts to outperform the baseline at 15,000 links (Figure 1a). For B3 and CEAF, neighborhood selection AL outperforms the baselines much earlier, at a few 1000 links (Figures 1b and 1c). It thus shows more robust performance for all evaluation metrics.

Neighborhood selection AL also performs at least as well as (for B3) or better than (MUC and CEAF) link selection AL. Learning curves of neighborhood selection AL are consistently above the link selection curves. We therefore consider neighborhood selection AL to be the preferred AL setup for CR.

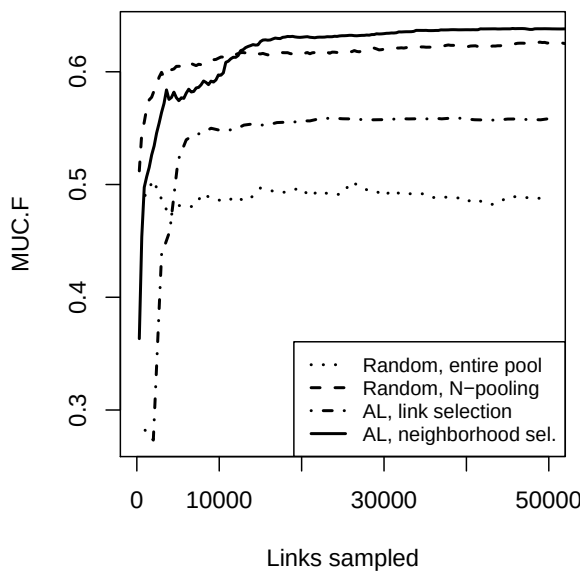
6 Conclusion

We have presented a new AL method for coreference resolution. The proposed method is novel in three respects. (i) It uses bootstrapped N-pooling, which ensures a class-balanced pool even though gold labels are not available. (ii) It further improves class balancing by neighborhood selection, a selection strategy that ensures coverage of positive and negative links per markable while still focusing on selecting difficult links. (iii) It is based on a query-by-committee selection strategy in contrast to earlier uncertainty sampling work. Experiments show that this new method outperforms random sampling in terms of both annotation effort and peak performance.

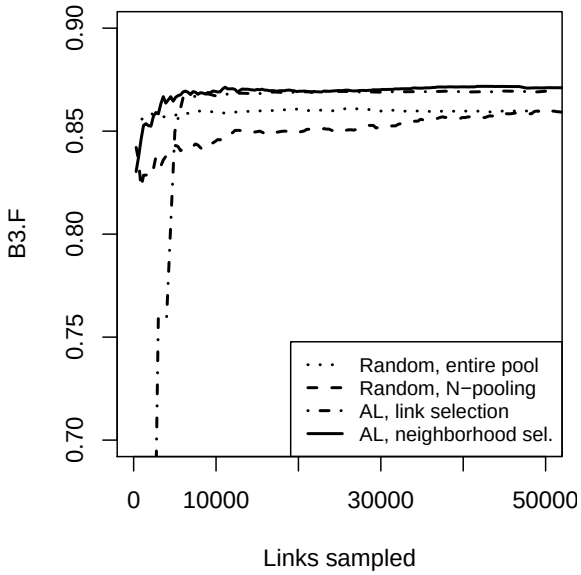
†Statistically significant at $p < .05$ compared to baseline 2 using the sign test ($N = 10$, $k \geq 9$ successes).

Acknowledgments

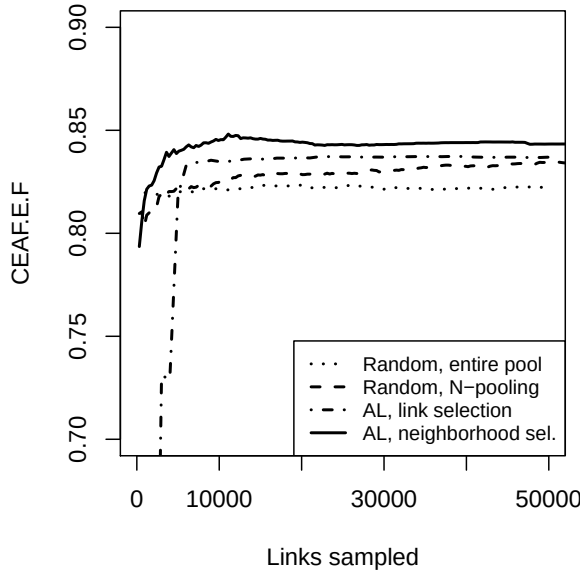
Florian Laws is a recipient of the Google Europe Fellowship in Natural Language Processing, and this research is supported in part by his fellowship. Florian Heimerl was supported by the Deutsche Forschungsgemeinschaft as part of the priority program 1335 'Scalable Visual Analytics'.



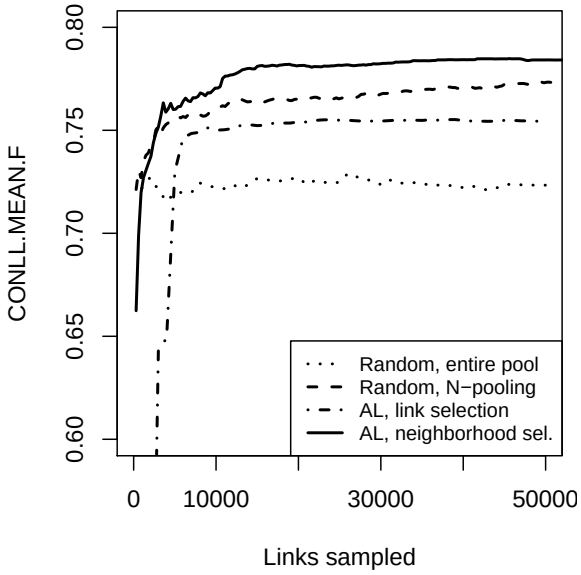
(a) Learning curve for MUC



(b) Learning curve for B3



(c) Learning curve for CEAF



(d) Learning curve for the mean of the CR measures.

Figure 1: Learning curves for AL and baseline experiments. All measures are F_1 measures.

References

S. Argamon-Engelson and I. Dagan. 1999. Committee-based sample selection for probabilistic classifiers. *JAIR*, 11:335–360.

A. Culotta, M. Wick, R. Hall, and A. McCallum. 2007. First-order probabilistic models for coreference resolution. In *HLT-NAACL 2007*.

Pedro Domingos and Michael J. Pazzani. 1997. On the optimality of the simple bayesian classifier under zero-one loss. *Mach. Learn.*, 29(2-3):103–130.

K. Dwyer and R. Holte. 2007. Decision tree instability and active learning. In *ECML*.

C. Gasperin. 2009. Active learning for anaphora resolution. In *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*.

E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel. 2006. Ontonotes: The 90% solution. In *HLT-NAACL*.

H. Kobdani, H. Schütze, M. Schiehlen, and H. Kamp. 2011. Bootstrapping coreference resolution using word associations. In *ACL*.

D. Lewis and W. Gale. 1994. A sequential algorithm for training text classifiers. In *SIGIR*.

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

M. Osborne and J. Baldridge. 2004. Ensemble-based active learning for parse selection. In *HLT-NAACL*.

S. Pradhan, L. Ramshaw, M. Marcus, M. Palmer, R. Weischedel, and N. Xue. 2011. Conll-2011 shared task: Modeling unrestricted coreference in ontonotes. In *CoNLL*.

M. Recasens, L. Màrquez, E. Sapena, M. A. Martí, M. Taulé, V. Hoste, M. Poesio, and Y. Versley. 2010. Semeval-2010 task 1: Coreference resolution in multiple languages. In *Proceedings of the 5th International Workshop on Semantic Evaluation*.

E. Ringger, P. McClanahan, R. Haertel, G. Busby, M. Carmen, J. Carroll, K. Seppi, and D. Lonsdale. 2007. Active learning for part-of-speech tagging: Accelerating corpus annotation. In *Linguistic Annotation Workshop at ACL-2007*.

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

K. Tomanek, J. Wermter, and U. Hahn. 2007. An approach to text corpus construction which cuts annotation costs and maintains reusability of annotated data. In *EMNLP-CoNLL*.

S. Tong and D. Koller. 2002. Support vector machine active learning with applications to text classification. *JMLR*, 2:45–66.